



TESI DOCTORAL

Títol Facing-up Challenges of Multiobjective Clustering Based on Evolutionary Algorithms: Representations, Scalability and Retrieval Solutions

Realitzada per Álvaro García Piquer

en el Centre Escola Tècnica Superior d'Enginyeria Electrònica i Informàtica La Salle

i en el Departament Informàtica

**Dirigida per Dr. Albert Fornells Herrera
Dra. Elisabet Golobardes i Ribé**

PhD Thesis

Facing-up Challenges of Multiobjective Clustering Based on Evolutionary Algorithms: Representations, Scalability and Retrieval Solutions

Álvaro Garcia-Piquer

Grup de Recerca en Sistemes Intel·ligents
Computer Science Department
ETSEEI La Salle - Universitat Ramon Llull

March 2012

Advisors: Dr. Albert Fornells Herrera and Dr. Elisabet Golobardes i Ribé

Summary

The era in which we live can be considered the Information Age because it is characterized by a technological revolution centered on digital technologies of information and communication. Large amount of information is collected every day, being the cornerstone of modern society. However, information is not useful if it is not properly managed to be transformed into wisdom through the extraction of understandable knowledge. Data Mining is the process of automatically extracting and discovering new, useful and understandable knowledge from huge volumes of data. It allows experts to accost the problems better in a specific domain and to obtain wisdom, such as the melanoma detection and managing the demand of energy more efficiently.

Data Mining involves four kind of techniques, and one of them is the clustering approach. It is based on grouping data according to a set of criteria, summarized in a single objective, obtaining groups where the elements are similar among them and different from the elements of the other clusters. These groupings provide a possible classification or categorization of the elements. Experts can obtain wisdom if they properly understand this categorization, for this reason it is necessary to obtain understandable patterns. Thus, experts may need to define several criteria to be optimized in the clustering process that cannot be summarized in a single objective due to their characteristics. Nevertheless, conventional clustering algorithms are not useful when more than one objective has to be optimized and it is necessary to apply other kind of methods.

This thesis is focused on multiobjective clustering algorithms, which are based on optimizing several objectives simultaneously obtaining a collection of potential solutions with different trade-offs among objectives. Specifically, the goal of the thesis is to design and implement a new multiobjective clustering technique based on evolutionary algorithms for facing up three current challenges related to this kind of techniques. The first challenge is focused on successfully defining the area of possible solutions that is explored in order to find the best solution, and this depends on the knowledge representation. The second challenge tries to scale-up the system splitting the original data set into several data subsets in order to work with less data in the clustering process. The third challenge is addressed to the retrieval of the most suitable solution according to the quality and shape of the clusters from the most interesting region of the collection of solutions returned by the multiobjective clustering algorithm. All the contributions related to these challenges are integrated in a framework called CAOS and successfully tested in a wide range of artificial and real-world data sets.

Resum

L'època a la que vivim pot ser considerada com l'Era de la Informació perquè es caracteritza per una revolució tecnològica centrada en les tecnologies de la informació i la comunicació. Cada dia es recullen grans quantitats d'informació, sent aquesta la pedra angular de la societat moderna. Però la informació no és útil si no es gestiona adequadament per transformar-la en saviesa mitjançant l'extracció de coneixement comprensible. La mineria de dades (*Data Mining*) és un procés que consisteix en extreure i descobrir automàticament coneixement nou, útil i comprensible a partir de grans volums de dades. Això permet als experts afrontar de manera més adequada els problemes en un domini específic i obtenir saviesa, com la detecció de melanomes o la gestió eficient de la demanda energètica, entre d'altres.

La mineria de dades comprèn quatre tipus de tècniques, entre elles el *clustering*. Aquesta tècnica es basa en agrupar dades segons un conjunt de criteris sintetitzats en un únic objectiu, obtenint grups de *clusters* on els elements són similars entre ells i diferents als elements dels altres *clusters*. Aquests grups ofereixen una possible classificació o categorització dels elements. Els experts poden ser capaços d'obtenir saviesa si entenen adequadament aquesta categorització, per aquesta raó és necessari obtenir patrons fàcilment comprensibles. Per tant, els experts poden necessitar definir varis criteris a optimitzar en el procés de *clustering* que no poden ser sintetitzats en un únic objectiu degut a les seves característiques. No obstant això, els algorismes de *clustering* convencionals no són útils quan s'ha d'optimitzar més d'un objectiu i, en aquests casos, és necessari aplicar altre tipus de mètodes.

Aquesta tesi es centra en algorismes de *clustering* multiobjectiu, que estan basats en optimitzar varis objectius simultàniament obtenint una col·lecció de solucions potencials amb diferents compromisos entre els objectius. Concretament, el propòsit d'aquesta tesi consisteix en dissenyar i implementar un nou algorisme de *clustering* multiobjectiu basat en algorismes evolutius per afrontar tres reptes actuals relacionats amb aquest tipus de tècniques. El primer repte es centra en definir adequadament l'àrea, on resideixen les possibles solucions, que s'explora per obtenir la millor solució i que depèn de la representació del coneixement. El segon repte consisteix en escalar el sistema dividint el conjunt de dades original en varis subconjunts per treballar amb menys dades en el procés de *clustering*. El tercer repte es basa en recuperar la solució més adequada tenint en compte la qualitat i la forma dels *clusters* a partir de la regió més interessant de la col·lecció de solucions ofertes per l'algorisme de *clustering* multiobjectiu. Totes les contribucions relacionades amb aquests reptes s'han integrat dins d'un marc anomenat CAOS i comprovades en un ampli rang de conjunts de dades artificials i del món real.

Resumen

La época en la que vivimos puede considerarse la Era de la Información porque se caracteriza por una revolución tecnológica centrada en las tecnologías de la información y la comunicación. Cada día se recogen grandes cantidades de información, siendo ésta la piedra angular de la sociedad moderna. Sin embargo, la información no es útil si no se gestiona adecuadamente para transformarla en sabiduría mediante la extracción de conocimiento comprensible. La minería de datos (*Data Mining*) es un proceso que consiste en extraer y descubrir automáticamente conocimiento nuevo, útil y comprensible a partir de grandes volúmenes de datos. Esto permite a los expertos afrontar de manera más adecuada los problemas en un dominio específico y obtener sabiduría, como la detección de melanomas o la gestión eficiente de la demanda energética, entre otros.

La minería de datos comprende cuatro tipos de técnicas, entre ellas el *clustering*. Esta técnica se basa en agrupar datos según un conjunto de criterios sintetizados en un único objetivo, obteniendo grupos (*clusters*) donde los elementos son similares entre ellos y diferentes a los elementos de los otros *clusters*. Estos grupos ofrecen una posible clasificación o categorización de los elementos. Los expertos pueden ser capaces de obtener sabiduría si entienden adecuadamente esta categorización, por esta razón es necesario obtener patrones fácilmente comprensibles. Por tanto, los expertos pueden necesitar definir varios criterios a optimizar en el proceso de *clustering* que no pueden ser sintetizados en un único objetivo debido a sus características. No obstante, los algoritmos de *clustering* convencionales no son útiles cuando se ha de optimizar más de un objetivo y, en estos casos, es necesario aplicar otro tipo de métodos.

Esta tesis se centra en los algoritmos de *clustering* multiobjetivo, que están basados en optimizar varios objetivos simultáneamente obteniendo una colección de soluciones potenciales con diferentes compromisos entre los objetivos. Concretamente, el propósito de esta tesis consiste en diseñar e implementar un nuevo algoritmo de *clustering* multiobjetivo basado en algoritmos evolutivos para afrontar tres retos actuales relacionados con este tipo de técnicas. El primer reto se centra en definir adecuadamente el área, donde residen las posibles soluciones, que se explora para obtener la mejor solución y que depende de la representación del conocimiento. El segundo reto consiste en escalar el sistema dividiendo el conjunto de datos original en varios subconjuntos para trabajar con menos datos en el proceso de *clustering*. El tercer reto se basa en recuperar la solución más adecuada teniendo en cuenta la calidad y la forma de los *clusters* a partir de la región más interesante de la colección de soluciones ofrecidas por el algoritmo de *clustering* multiobjetivo. Todas las contribuciones relacionadas con estos retos se han integrado en un marco llamado CAOS y comprobadas en un amplio rango de conjuntos de datos artificiales y del mundo real.

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this PhD thesis.

First and foremost, I owe my deepest gratitude to my supervisors Dr. Albert Fornells and Dr. Elisabet Golobardes for their guidance, patience and support. I am grateful to the *Grup de Recerca en Sistemes Intel·ligents* (GRSI) for giving me the opportunity to start my research career with them, and I would extend this acknowledgement to all members of the group who, directly or indirectly, trained me as a researcher.

I would like to show my gratitude to the *Automated Scheduling, Optimisation and Planning* (ASAP) research group for allowing me to visit them, and specially to Dr. Jaume Bacardit for receiving me so well and for all the great support and guidance provided. I also thank all the great people I met in the ASAP and the efforts they made to help me to improve my English skills.

The present work is the result of the collaboration with several researchers. I would like to thank my coauthors Jaume Bacardit, Guiomar Corral, Albert Fornells, Elisabet Golobardes, Albert Orriols and Andreu Sancho. I also want to extend this gratitude to Francesc Teixidó for the technical support he has always given so selflessly.

Last but not least, I would like to thank the unconditional support that all my family and friends have given me over these long four years. Specially, I want to thank my parents for having instilled in me the desire to learn and to improve in everything I do.

This thesis would not have been possible unless the financial support of the *Departament d'Universitats, Recerca i Societat de la Informació* (DURSI) and the *European Social Fund* (ESF) under a scholarship in the FI research program (2011FI_B1 00022 and 2010FI_B 01084) and under the BE travel grant (2010BE 01026) for my visit to the ASAP. Finally, I would like to acknowledge the *Ministerio de Educación y Ciencia* for its support under the MID-CBR (TIN2006-15140-C03-03) and KEEL-III (TIN2008-06681-C06-05) projects, to the *Ministerio de Industria, Turismo y Comercio* for its support under the GAD project (CEN200710126), to the *Agència per a la Qualitat del Sistema Universitari de Catalunya* (AQU) for supporting the *Guidelines for Competence Assessment in Engineering and Architecture* (IUE/3013/2007), and to the *Generalitat de Catalunya* for its support under the grant 2009-SGR-183.

Contents

Abstract	3
Acknowledgements	9
Contents	11
List of Figures	15
List of Tables	21
List of Algorithms	27
1 Introduction	29
1.1 Framework	29
1.2 Dissertation Scope	30
1.3 Objectives and Contributions	34
1.4 Roadmap	35
1.5 Summary	36
I Theoretical Background	37
2 Clustering	39
2.1 Taxonomy	39
2.2 Algorithms	43
2.3 When One Criterion Is Not Enough	45
2.4 When Many Criteria is Too Much	47
2.5 Validation Techniques	49
2.5.1 Statistical Testing in Clustering Validation	49
2.5.2 External Criteria	50
2.5.3 Internal Criteria	53
2.5.4 Relative Criteria	54
2.6 Summary	56

3	Evolutionary Algorithms	57
3.1	Natural Principles	57
3.2	Taxonomy	58
3.3	Optimizing Several Objectives	61
3.4	Scalability	64
3.4.1	Parallelism	64
3.4.2	Data Subsets Sampling	66
3.5	Summary	68
II	CAOS: Designing a New Multiobjective Clustering Algorithm	69
4	Foundations	71
4.1	Motivation	71
4.2	Related Work	72
4.3	Process and Design	72
4.4	Objective Functions	74
4.5	Selection of the Best Solution	75
4.6	Additional Features	76
4.6.1	Bloat Control	76
4.6.2	Cluster Merging	76
4.7	Summary and Conclusions	78
5	Definition and Exploration of the Search Space	81
5.1	Motivation	81
5.2	Related Work	82
5.3	Representations	83
5.3.1	Prototype-Based Representation	83
5.3.2	Label-Based Representation	85
5.3.3	Graph-Based Representation	86
5.3.4	Search Space and Computational Performance	86
5.4	Experiments, Results and Discussion	87
5.4.1	Experimental Methodology	87
5.4.2	Comparison of Individual Representation	89
5.4.3	Performance of CAOS Regarding Single-Objective Clustering Methods	97
5.4.4	Discussion	101
5.5	Summary and Conclusions	102
6	Large Data Management	105
6.1	Motivation	105
6.2	Related Work	106

6.3	Data Subset Strategies	107
6.3.1	Creation of Strata	107
6.3.2	Evolution Based on Strata	108
6.3.3	Computational Performance Models	108
6.4	Experiments, Results and Discussion	110
6.4.1	Experimental Methodology	111
6.4.2	Comparison of Results	112
6.4.3	Discussion	116
6.5	Summary and Conclusions	116
7	Selection of the Most Suitable Solution	119
7.1	Motivation	119
7.2	Related Work	120
7.3	Sweet Spot Selection Technique	121
7.3.1	The Sweet Spot Combined with Clustering Validation Techniques	123
7.3.2	Identification of the Sweet Spot in CAOS	123
7.4	Experiments, Results and Discussion	124
7.4.1	Experimental Methodology	124
7.4.2	Comparison of Results	126
7.4.3	Discussion	129
7.5	Summary and Conclusions	129
III	Practical Application of CAOS in Real-World Problems	131
8	Decision Support System for the Analysis of Vulnerabilities in Telematic Networks	133
8.1	Motivation	133
8.2	Related Work	134
8.3	Definition of Specific Optimization Objectives According to the Domain	135
8.4	Consensus and Analia to Analyze Security Tests	137
8.4.1	Description	137
8.4.2	Single-Objective Clustering	138
8.4.3	Multiobjective Clustering	139
8.5	Experiments, Results and Discussion	140
8.5.1	Experimental Methodology	141
8.5.2	Comparison of Optimization Objectives	142
8.5.3	Performance of CAOS with Specific Objectives Regarding Single-Objective Clustering Methods	142
8.5.4	Moving on to the Expert Side	145
8.5.5	Discussion	146
8.6	Summary and Conclusions	146

9	Validation of the Acquisition of Competences in University Degrees	149
9.1	Motivation	149
9.2	Related Work	150
9.3	Definition of Specific Optimization Objectives According to the Domain	150
9.4	Experiments, Results and Discussion	151
9.4.1	Experimental Methodology	151
9.4.2	Comparison of Results	152
9.4.3	Discussion	154
9.5	Summary and Conclusions	155
10	Multiobjective Knowledge Organization in CBR Systems	157
10.1	Motivation	157
10.2	Related Work	159
10.3	CAOSCBR: Organizing Case Memories Using CAOS	159
10.4	Experiments, Results and Discussion	159
10.4.1	Experimental Methodology	160
10.4.2	Comparison of Results	162
10.4.3	Analysis of Results on the Complexity Space	164
10.4.4	Discussion	167
10.5	Summary and Conclusions	167
IV	Conclusions and Further Work	169
11	Work Done, Lessons Learned and Future Work Lines	171
11.1	Recapitulation and Key Conclusions	171
11.2	Forthcoming Research	176
11.3	Summary	178
V	Appendix	179
A	Description of Single-Objective Clustering Algorithms	181
B	Full Results of the Experimentation of Chapter 5	183
C	Full Results of the Experimentation of Chapter 6	189
D	Full Results of the Experimentation of Chapter 7	217
E	Full Results of the Experimentation of Chapter 10	221
	References	231

List of Figures

1.1	Pyramid of the DIKW hierarchy proposed by Russell Ackoff.	31
1.2	Example of a Pareto front of solutions. The green solutions are non-dominated and the red one is dominated.	33
1.3	Overview of CAOS process. The input data is given to the CAOS system that faces up three MC challenges: (1) to manage large data sets dividing them in several data subsets; (2) to define the search space through the selection of the most suitable objectives, individual representation, genetic operators and MOEA parameters; and (3) to obtain the output of the system by selecting the final clustering result from the Pareto set of solutions.	35
2.1	Example of partitional and hierarchical clustering results from the same elements. Each color corresponds to a different level of hierarchy.	41
2.2	Example of a dendrogram to represent hierarchical clustering and its interpretation. Each color corresponds to a different level of hierarchy.	41
2.3	Example of hard and fuzzy clustering results from the same elements.	42
2.4	Taxonomy proposed for classifying clustering algorithms.	43
2.5	Example 1 of clustering problem. (a) Elements to be clustered. (b) Elements clustered optimizing one criterion: the same shape of the elements in each cluster. (c) Elements clustered optimizing one criterion: to minimize the number of clusters. (d) Elements clustered optimizing two criteria: the same shape of the elements in each cluster and to minimize the number of clusters.	45
2.6	Example 2 of clustering problem. (a) Elements to be clustered. (b) Elements clustered optimizing the intra-cluster variance. (c) Elements clustered optimizing the inter-cluster variance.	46
2.7	Ensemble Clustering and Multiobjective Clustering process for the clustering problem described in Example 2. (a) Process to obtain an Ensemble Clustering solution. (b) Multiobjective Clustering Pareto set of solutions.	47

2.8	Schematic view of Pareto-dominance (Farina and Amato, 2002) based on partial order in problems with (a) 2 objectives and (b) 3 objectives when a candidate solution (\bullet) is considered. Considering that the objectives have to be minimized, the green areas dominate the candidate solution, the red areas are dominated by the candidate solution, and the white areas contain equivalent solutions to the candidate one.	48
3.1	Example of an individual representation in a EA. The genotype of N genes represents the genetic information of a fly.	58
3.2	Cycle of the steps of an Evolutionary Algorithm.	59
3.3	Example of crossover and mutation operators used to generate a new individual from two existing individuals. The new individual is generated crossing individual 1 and individual 2 using the crossover operator, which mixes the first half of the genotype of individual 1 with the second half of the genotype of individual 2. Finally, the generated individual is mutated with the mutation operator, which changes the allele of one gene, obtaining the final genetic information of the new individual.	60
3.4	Cycle of the steps of a Multiobjective Evolutionary Algorithm.	63
3.5	Parallel EA taxonomy.	64
4.1	Overview of the CAOS process with the steps that will be modified to face up the challenges. The steps related to the challenge that deals with the definition and exploration of the search space are colored in green, the step affected by facing up the scalability with large data sets is colored in blue, and the step related to the challenge that faces up the selection of the final clustering result is colored in orange.	77
5.1	Individual representation for a clustering solution using a data set of 7 instances with 2 attributes. The example shows (a) the example data set, (b) the phenotypic interpretation of the clustering, (c,d,e) the individual representation of each approach, and (f) the concept resulting from the graph-based representation.	83
5.2	Accuracy rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank is the best one. E , B , M and BM represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. In addition to these abbreviations, each retrieved solution is represented by rd , dv , dn , sl corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. CD indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.	91


- 5.3 Number of individuals rank with Nemenyi test of each CAOS configuration of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank indicates that the configuration deals with less number of individuals in the genetic cycle. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method. 91
- 5.4 Number of clusters rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank indicates that the solution has less number of clusters. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method. 92
- 5.5 Accuracy rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations obtained with heuristical and random initialization. The lower rank is the best one. *H* represents the heuristical initialization and *R* represents the random initialization. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method. 94
- 5.6 Accuracy rank with Nemenyi test of the solutions of the best configuration of each CAOS representation for (a) all the data sets, (b) the artificial data sets, (c) the handmade artificial data sets, and (d) real data sets. The lower rank is the best one. *P*, *L* and *G* represent the prototype-based, the label-based and the graph-based representations respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method. 95
- 5.7 Graphical representation of the original classes of the (a) *spiral*, and (b) *square3* data sets. 96

5.8	Graphical representation of the clusters found in the <i>spiral</i> data set by (a) CAOS prototype-based, (b) CAOS label-based, (c) CAOS graph-based, and (d) conventional clustering algorithms. The solutions presented at (a), (b) and (c) were retrieved with Adjusted Rand, Davies-Bouldin, Dunn and Silhouette indexes, from left to right. The solutions presented at (d) are <i>k</i> -means, <i>x</i> -means, EM and SOM from left to right.	98
5.9	Graphical representation of the clusters found in the <i>square3</i> data set by (a) CAOS prototype-based, (b) CAOS label-based, (c) CAOS graph-based, and (d) conventional clustering algorithms. The solutions presented at (a), (b) and (c) were retrieved with Adjusted Rand, Davies-Bouldin, Dunn and Silhouette indexes, from left to right. The solutions presented at (d) are <i>k</i> -means, <i>x</i> -means, EM and SOM, from left to right.	99
5.10	Accuracy rank with Nemenyi test of the solutions of the best configuration of each CAOS representation and the solutions of the single objective algorithms for (a) all the data sets, (b) the artificial data sets and (c) real data sets. The lower rank is the best one. <i>P</i> , <i>L</i> and <i>G</i> represent the prototype-based, the label-based and the graph-based representations respectively. In addition to these abbreviations, each retrieved solution is represented by <i>rd</i> , <i>dv</i> , <i>dn</i> , <i>sl</i> corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. <i>km</i> , <i>xm</i> , <i>EM</i> and <i>SOM</i> represent the results of the <i>k</i> -means, <i>x</i> -means, EM and SOM algorithms respectively. <i>CD</i> indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.	100
6.1	Accuracy difference of the three CAOS _{DS} strategies regarding CAOS _{CD} . (a) Artificial data sets and (b) real-world data sets.	114
6.2	Speedup of the three CAOS _{DS} strategies regarding CAOS _{CD} in artificial (a,c,e) and real-world data sets (b,d,f). The first two figures (a,b) are referred to the speedup of the precalculation time. To build the data subsets and to precalculate the distance and nearest neighbors structures. The two following figures (c,d) show the speedup of the clustering time to do the evolutionary process that obtains the Pareto set of solutions. The last two figures (e,f) are related to the speedup of the overall time taking into account both times.	114
7.1	Validation indexes results from two non-dominated clustering solutions. The red color indicates the solutions selected by an index but not selected by experts.	122
7.2	Validation indexes results from two non-dominated clustering solutions. The green color indicates the solutions selected by each index.	122
7.3	Graphical representation of the sweet spot identification. α_1 and α_2 are the angles that determine the size of the sweet spot.	123

7.4	Accuracy rank with Nemenyi test of the most suitable CAOS solution obtained with the indexes (a) Davies, (b) Dunn, (c) Silhouette, (d) Calinski-Harabasz, and (e) the adjacent angles strategy with all the Pareto set (A) and with using different angles to the define the sweet spot size: 1 (F1), 2 (F2), 5 (F5), 10 (F10), 15 (F15), 20 (F25), 30 (F30), 35 (F35), 40 (F40), 43 (F43) and 44 (F44) degrees. CD indicates the value of the critical distance, representing with a line the area that is not significantly different with respect to the best ranked method.	127
7.5	Accuracy rank with Nemenyi test of the most suitable CAOS solution obtained with the indexes Davies (Dv), Dunn (Dn), Silhouette (Sl), Calinski-Harabasz (CH), and the adjacent angles strategy (Ag) with all the Pareto set (A) and with the best sweet spot size for each strategy (F). Also, the best solution retrieved according the Adjusted Rand index is shown (Rd). The results are obtained using (a) artificial data sets and (b) real-world data sets. CD indicates the value of the critical distance, representing with a line the area that is not significantly different with respect to the best ranked method.	128
8.1	Architecture of <i>Consensus</i> system and <i>Analia</i> data analysis module.	138
9.1	Clusters of some example subjects from the Computer Engineering degree according to the evaluation methods that they use. Only the most relevant evaluation methods are shown.	154
9.2	Clusters of some example subjects from the Computer Engineering degree according to the competences that they provide. Only the most relevant competences are shown.	154
10.1	Case retrieval using the 3 most similar elements ($K=3$) from the most similar cluster ($C=1$), v_i is the centroid of each cluster. In the example, the similarity is based on the normalized Euclidean distance using an input case composed by two attributes (0.5,0.7).	160
10.2	Comparison of the average test performance of each configuration against each other with the Nemenyi test for all the data sets. Grey area is the optimal one. . . .	162
10.3	Complexity map of the analyzed data sets. Complexity grows from A to C in radial shape.	165
10.4	Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type A. Grey area is the optimal one.	165
10.5	Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type B. Grey area is the optimal one.	166
10.6	Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type C. Grey area is the optimal one.	166

- 11.1 This thesis has faced up three challenges –the definition and exploration of the search space, the scalability with large data sets and the retrieval of solutions– in the context of MC based on MOEAs to develop the CAOS algorithm, which has been successfully tested in different real-world projects. Moreover, CAOS has been used in combination with other techniques such as Case-Based Reasoning and complexity measures. Starting from the bottom we can see the contexts of the thesis, the challenges faced, the system obtained and the domains of application in real-world projects. 174

List of Tables

5.1	Cost of each representation broken down in clustering cost and merge clusters cost. Where g is the number of generations, $ IP $ is the internal population size, m and t are the number of instances and attributes of the data set respectively, \bar{n} is the average of the number of clusters of the individuals, and n is the number of clusters of the retrieved individual.	87
5.2	Summary of the characteristics of the 35 artificial data sets (left block) and real-world data sets (right block) used. The symbol  indicates the handmade data sets. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).	88
6.1	Computational cost of CAOS applied to the complete data set and to data subsets (CAOS – CD and CAOS – DS respectively) broken down in initialization cost and clustering cost. Where g is the number of generations, $ IP $ is the internal population size, m and t are the number of instances and attributes of the data set respectively, \bar{n}_{cd} is the average of the number of clusters of the individuals (the minimum number of clusters is 1 and the maximum m), \bar{n}_{ds} is the average of the number of clusters of the individuals (the minimum number of clusters is 1 and the maximum $\frac{p}{numStrata}$), ℓ is the percentage of the nearest elements taken into account, and $numStrata$ is the number of strata generated.	110
6.2	Memory usage of CAOS applied to the complete data set and to data subsets (CAOS – CD and CAOS – DS respectively) to store the nearest neighbors. Where m is the number of instances of the data set, ℓ is the percentage of instances considered neighbors, $numStrata$ is the number of strata generated and $sizeof(data\ type)$ is the size in bytes of the data type.	110
6.3	Summary of the characteristics of the 25 real-world data sets used. The columns are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).	111

- 6.4 Comparison of the algorithms in the artificial data sets using Holm's procedure with $\alpha = 0.05$. The algorithms compared are CAOS using the CAOS_{CD} and the three CAOS_{DS} strategies to generate data subsets: based on classes, random and based on clusters; represented by CAOS – DS – Classes, CAOS – DS – Random and CAOS – DS – Clusters respectively. The results are shown for 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of information used from the complete data set. The symbols \oplus and \ominus show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column. Similarly, the symbols + and – denote a non-significant higher/lower results. . . . 113
- 6.5 Comparison of the algorithms in the real-world data sets using Holm's procedure with $\alpha = 0.05$. The algorithms compared are CAOS using the CAOS_{CD} and the three CAOS_{DS} strategies to generate data subsets: based on classes, random and based on clusters; represented by CAOS – DS – Classes, CAOS – DS – Random and CAOS – DS – Clusters respectively. The results are shown for 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of information used from the complete data set. The symbols \oplus and \ominus show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column. Similarly, the symbols + and – denote a non-significant higher/lower results. . . . 113
- 7.1 Summary of the characteristics of the 35 artificial data sets (left block) and real-world data sets (right block) used. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC). 125
- 8.1 Pairwise comparisons of the *Intra-Inter* value of the clustering solutions returned by CAOS-DC (CDC) and CAOS-II (CII) by means of a Holm's procedure. We have run CAOS-DC and CAOS-II considering all the indexes to recover the best solution, i.e., *Intracohesion* (Ita), *Intercohesion* (Ite), Davies-Bouldin (DB), Dunn (Dn), Silhouette (Sil), *Intra-Inter* (II). The symbol \oplus shows that the method in the row obtained results that were significantly higher than those obtained with the method in the column. Similarly, the symbol + denote a non-significant higher results. 142
- 8.2 Pairwise comparisons of the *Davies-Bouldin* value of the clustering solutions returned by CAOS-II (CII) with the different validation indexes to recover the best solution and SOM with 3×3 (SOM3), SOM with 4×4 (SOM4), SOM with 5×5 (SOM5), SOM with 6×6 (SOM6), *k*-means with *k* ranging from 3 (KM3) to 10 (KM10), and *x*-means with $min_k = 3$ and max_k ranging from 4 (XM4) to 10 (XM10). The symbol \oplus shows that the method in the row obtained results that were significantly higher than those obtained with the method in the column. Similarly, the symbol + denote a non-significant higher results. 144

8.3	Pairwise comparisons of the Intra-Inter value of the clustering solutions returned by CAOS-II (CII) with the different validation indexes to recover the best solution and SOM with 3×3 (SOM3), SOM with 4×4 (SOM4), SOM with 5×5 (SOM5), SOM with 6×6 (SOM6), k -means with k ranging from 3 (KM3) to 10 (KM10), and x -means with $min_k = 3$ and max_k ranging from 4 (XM4) to 10 (XM10). The symbol \ominus shows that the method in the row obtained results that were significantly lower than those obtained with the method in the column. Similarly, the symbol $-$ denote a non-significant lower results.	144
8.4	Pairwise comparison of the Intra-Inter value of the clustering solutions returned by CAOS-II with recuperation of the best solution based on Intra-Inter, SOM 6×6 , k -means with $k = 9$ (KM), and x -means with $max_k = 10$ (XM). The symbol \ominus shows that the method in the row obtained results that were significantly lower than those obtained with the method in the column. Similarly, the symbol $-$ denote a non-significant lower results.	145
9.1	Competences defined in La Salle–Ramon Llull University degrees. For each competence it is indicated the cluster assigned in the clustering result.	153
9.2	Most representative evaluation methods from each one of the competence clusters found. The first column indicates the competence cluster and the second one indicates the name of the evaluation methods.	153
10.1	Summary of the characteristics of the 56 used data sets. The columns are referred to the number of cases (nCs), to the number of attributes (nAt), to the number of classes (nCl) and to the data set complexity (Comp) defined (A, B, C).	161
B.1	Accuracy and standard deviation of conventional algorithms and CAOS solutions with the artificial data sets. The accuracy was calculated with the Adjusted Rand index.	184
B.2	Accuracy and standard deviation of conventional algorithms and CAOS solutions with the real-world data sets. The accuracy was calculated with the Adjusted Rand index.	185
B.3	The number of clusters and its standard deviation of conventional algorithms and CAOS solutions with the real-world data sets.	186
B.4	The number of clusters and its standard deviation of conventional algorithms and CAOS solutions with the real-world data sets.	187
C.1	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the artificial data sets.	190
C.2	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the artificial data sets.	191

C.3	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the artificial data sets.	192
C.4	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the artificial data sets.	193
C.5	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the artificial data sets.	194
C.6	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the artificial data sets.	195
C.7	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the artificial data sets.	196
C.8	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the artificial data sets.	197
C.9	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the real-world data sets.	198
C.10	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the real-world data sets.	198
C.11	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the real-world data sets.	198
C.12	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the real-world data sets.	199
C.13	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the real-world data sets.	199
C.14	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the real-world data sets.	199
C.15	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the real-world data sets.	200
C.16	Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the real-world data sets.	200
C.17	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the artificial data sets. . .	201
C.18	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the artificial data sets. .	202
C.19	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the artificial data sets. .	203
C.20	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the artificial data sets. .	204
C.21	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the artificial data sets. .	205

C.22	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the artificial data sets.	206
C.23	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the artificial data sets.	207
C.24	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the artificial data sets.	208
C.25	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the real-world data sets.	209
C.26	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the real-world data sets.	210
C.27	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the real-world data sets.	211
C.28	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the real-world data sets.	212
C.29	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the real-world data sets.	213
C.30	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the real-world data sets.	214
C.31	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the real-world data sets.	215
C.32	Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the real-world data sets.	216
D.1	Accuracy results of CAOS with the artificial data sets retrieving the most suitable solution from the overall Pareto set and from the sweet spot. The solutions retrieved by each clustering validation index and by the strategy based on adjacent angles are shown.	218
D.2	Accuracy results of CAOS with the real-world data sets retrieving the most suitable solution from the overall Pareto set and from the sweet spot. The solutions retrieved by each clustering validation index and by the strategy based on adjacent angles are shown.	219
E.1	Summary of the average error achieved by the CBR configurations when the most similar cluster is selected ($C=1$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.	222
E.2	Summary of the average error achieved by the CBR configurations when the two most similar clusters are selected ($C=2$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.	223

-
- E.3 Summary of the average error achieved by the CBR configurations when the three most similar clusters are selected ($C=3$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 224
- E.4 Summary of the average error achieved by the CBR configurations when the five most similar clusters are selected ($C=5$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 225
- E.5 Summary of the average number of cases used by the CBR configurations when the most similar cluster is selected ($C=1$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 226
- E.6 Summary of the average number of cases used by the CBR configurations when the two most similar clusters are selected ($C=2$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 227
- E.7 Summary of the average number of cases used by the CBR configurations when the three most similar clusters are selected ($C=3$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 228
- E.8 Summary of the average number of cases used by the CBR configurations when the five most similar clusters are selected ($C=5$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets. 229

List of Algorithms

- 2.1 Statistical test based on Monte Carlo techniques to test the null hypothesis that clustering C is randomly structured according to a prespecified partition P 52
- 2.2 Statistical test based on Monte Carlo techniques to test the null hypothesis that the prespecified partition P is randomly structured according to the similarity matrix S . . . 53
- 2.3 Statistical test based on Monte Carlo techniques to test the null hypothesis that clustering C is randomly structured according to the similarity matrix S 53

- 4.1 Life cycle of CAOS. 73

- 6.1 Strata generation based on classes. 108
- 6.2 Strata generation based on random instances selection. 109
- 6.3 Strata generation based on approximative clusters. 109

Chapter 1

Introduction

Data Mining is the process of automatically extracting and discovering new, useful and understandable knowledge from huge volumes of data. It involves four kind of techniques, and one of them is the clustering approach. This technique is based on grouping data according to a set of criteria, summarized in a single objective, obtaining groups where the elements are similar among them and different from the elements of the other clusters. These groupings can help to the experts to acquire wisdom. Nevertheless, conventional clustering algorithms may not be useful when the criteria to be optimized cannot be summarized in a single objective, and therefore it is necessary to optimize independently each one. This thesis is focused on multiobjective clustering algorithms, which are based on optimizing several objectives simultaneously. In this chapter we present the framework, the dissertation scope, the objectives and contributions, and the roadmap of the thesis in order to set the reader in the context of it.

1.1 Framework

The research done in this PhD thesis is framed on the graduate program in Information Technology and Management from La Salle at Ramon Llull University. It has been developed in the Research Group in Intelligent Systems¹ (GRSI), which is a research group created in 1994 and recognized as a consolidated group by the Government of Catalonia since 2002 (2002-SGR-00155, 2005-SGR-00302 and 2009-SGR-183). The research field of the group is focused on Artificial Intelligence, specifically, on Data Mining and Machine Learning under the paradigms of Case Based Reasoning, Evolutionary Algorithms and Soft-Computing with the aim of solving classification, diagnosis and prediction problems in different fields, like health, networks security, energy and education domains. During the development of my PhD thesis I have had the opportunity of relating my research to three projects founded by the Spanish Government: (1) MID-CBR (TIN2006-15140-C03-03) was based on developing a unified framework for the development of Case-Based Reasoning systems, in it my research was focused on improving the steps of the Case-based Reasoning using Soft Computing techniques; (2) GAD (CEN200710126) was focused on the active manage-

¹<http://www.salleurl.edu/GRSI>

ment of the electrical demand and my task was to conduct research on the module related to the efficient management of the power demand through the identification of consumer behavior using Clustering techniques; and (3) KEEL-III (TIN2008-06681-C06-05) is focused on the current trends and new challenges in Knowledge Discovery based on Evolutionary Learning and my task there was related to the knowledge extraction from data using Evolutionary Algorithms. Also I have taken part in a project founded by the *Generalitat de Catalunya* called *Guidelines for Competence Assessment in Engineering and Architecture* (IUE/3013/2007), which consisted in the evaluation of competences in engineering and architecture to help the experts to adapt the university graduates to the European Higher Education Area. My PhD research has been cosupervised by Dr. Albert Fornells Herrera and Dr. Elisabet Golobardes i Ribé, and it has been supported by the *Generalitat de Catalunya*, the Commission for Universities and Research of the DIUE and European Social Fund (2010FLB01084, 2011FLB100022, 2010BE_01026).

1.2 Dissertation Scope

Nowadays, the era in which we live is considered the Information Age, a term coined by the sociologist Manuel Castells for post-1990 era (Castells and Martínez, 2001). Castells describes the Information Age as a historical period characterized by a technological revolution centered on digital technologies of information and communication. Information is the cornerstone of modern society because it is considered that it leads to power and success, so large amount of information is collected each day. But information alone does not mean anything, it is necessary to properly manage it to turn it into wisdom. The American organizational theorist Russell Ackoff describes the content of the human mind in five categories: (1) Data, (2) Information, (3) Knowledge, (4) Understanding and (5) Wisdom (Ackoff, 1989). The relationship between these categories is described through a hierarchy called DIKW (see Figure 1.1), and it can be summarized as: (1) data are symbols that do not have meaning by themselves, they are usually the results of measurements or observations; (2) information is data that are preprocessed, it can be useful or not, and it provides “who”, “what”, “where”, and “when” questions; (3) knowledge is the appropriate collection of information, such that its intent is to be useful, it answers “how” questions; (4) understanding is the process by which knowledge can be taken and subsequently synthesized in new knowledge from the previously held knowledge, and it answers “why” question; and (5) wisdom is the evaluated understanding, unlike the previous four levels, it asks questions to which do not have an easy answer, and in some cases, to which there can be no humanly-known answer period. According to this hierarchy, information only provides wisdom when it contains previously unknown, useful and understandable knowledge. As Clifford Stoll said, “Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom”. The aim of data management in Information Technology is to obtain wisdom from data. Concretely, according to the DIKW hierarchy, the objective is to obtain new, useful and understandable knowledge from data that can become in wisdom. In the last decades, the amount and heterogeneity of data have

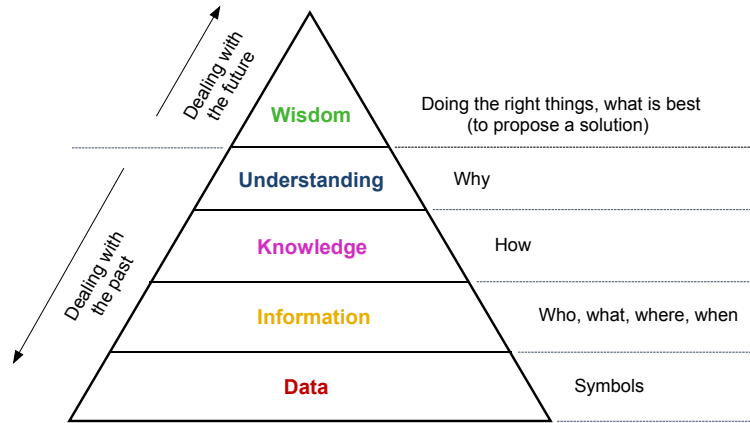


Figure 1.1: Pyramid of the DIKW hierarchy proposed by Russell Ackoff.

shown the limits of the conventional data management methods opening the door to a new kind of techniques focused on obtaining patterns and relationships between data in order to obtain new knowledge. Moreover, the spread of these techniques has been possible thanks to the technology improvements of the computers in terms of processing speed and storage capacity.

Data Mining and Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996) are the processes of automatically extracting and discovering new, useful and understandable knowledge from huge volumes of data. That is, automatically or semi-automatically obtaining groups or relationships between the objects of a collection of data according to their features, with the aim of allowing experts to confront better the problems of the domain (Hernández et al., 2004) or, in other words, to obtain wisdom. The reality is plenty of examples such as the detection of melanoma cancer in a patient using the knowledge obtained from the analysis and diagnosis of previous patients, and managing more efficiently the demand of energy by the identification of groups of consumers with similar behavior using recorded readings of electrical consumption. Even if the Data Mining term as it is known nowadays is quite recent and it dates from the nineties, the idea is older. In the seventies, the statisticians managed terms like data fishing, data mining or data archaeology with the idea of finding correlations without any previous hypothesis in noisy data bases to obtain knowledge that helps to better understand a concrete problem. In the early eighties, Rakesh Agrawal, Gio Wiederhold, Robert Blum and Gregory Piatetsky-Shapiro, among others, began to consolidate the Data Mining and Knowledge Discovery in Databases (KDD) terms (Fayyad et al., 1996), involving four typologies of techniques: (1) association rule learning, (2) classification, (3) regression and (4) clustering. The first one searches for relationships between variables. The second one can obtain a model generalizing from a known structure to apply to new data. The third one tries to find a function which models the data with the least error. Finally, in the last one is where this thesis is framed. Clustering approach (Duda et al., 2000) is based on grouping data according to a set of criteria, being the result a set of groups (also called clusters) where each one contains a set of similar elements. Thus, the elements in a cluster are similar among them and different from the elements of the other clusters and it provides to the experts a possible classification

or categorization of the elements. This issue is particularly useful in unsupervised domains, where the categorization of the elements is unknown, and these techniques turn into mechanisms to identify patterns and discover relationships. Nowadays, there are numerous clustering algorithms that are applied in a wide range of domains such as the analysis of gene expressions to identify living beings from data that contains DNA analysis (Eisen et al., 1998; Yeung et al., 2003); the identification of groups of patients to apply common medical treatments (Hodges and Wotring, 2000); the definition of market segments related to a product of service in order to recommend them to the suitable consumers (Hofstede et al., 1999); or the image segmentations based on the identification of objects and edges in images according to the features of each pixel (position, color, neighbors...) (Comaniciu and Meer, 2002).

Clustering algorithms can be classified according to many different points of view (Duda et al., 2000; Witten and E. Frank, 2011) such as: the relation between the clusters (partitional or hierarchical), how they are structured (for instance, center-based, search-based and model-based), the degree of membership to the cluster (for example, hard clustering and fuzzy clustering), or the criteria used to build the clusters (conventional clustering, Ensemble Clustering, or Multiobjective Clustering). This last point is key because criteria determine the final shape and size of groups. Conventional clustering algorithms (Gan et al., 2000) are based on grouping elements using only one objective based on one or more criteria expressed in a single function. Nevertheless, this kind of clustering may fail if criteria cannot be combined in only one objective and, consequently, several objectives have to be defined to tackle the problem. On the other hand, Ensemble Clustering (EC) (Strehl and Ghosh, 2002) and Multiobjective Clustering (MC) (Ferligoj and Batagelj, 1992) optimize several objectives at the same time. The main difference between both approaches lies in the strategy to optimize the criteria. EC divides the procedure into two phases: (1) application of many clustering algorithms following different single objectives; and (2) combination of the last results to create the final clustering. This last step is quite complex to achieve and it is usually inefficient because the objectives can be partially contradictory, thus it is not trivial to define how to combine or weight the relevance of objectives. In contrast, MC creates solutions based on multiple criteria taking into account all the objectives at the same time. Concretely, each clustering solution is evaluated for each objective, and the final result is a collection of clustering solutions with different trade-offs. Therefore, EC does not fully exploit the potential of using several objectives due to the fact that they are limited to the combination of the solutions returned by the conventional clustering algorithms, and they cannot explore trade-off solutions during the clustering process (Handl and Knowles, 2007). In contrast, MC are more flexible to tackle the clustering problem as a truly multiobjective optimization that has been previously pointed out (Dale and Dale, 1992; Ferligoj and Batagelj, 1992).

There are different techniques for multiobjective optimization such as Simulated Annealing (Saha and Bandyopadhyay, 2010) and Ant Colony Optimization (Iredi et al., 2000), but Multiobjective Evolutionary Algorithms (MOEAs) (Coello, 1999) have become one of the most capable techniques to solve these kinds of problems (Fonseca and Fleming, 1995; Zitzler et al., 2000) since

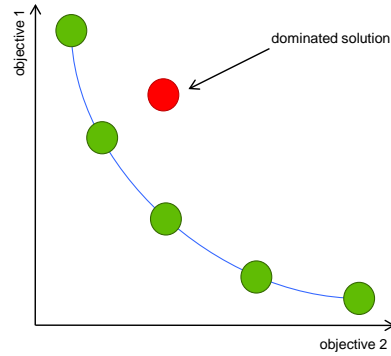


Figure 1.2: Example of a Pareto front of solutions. The green solutions are non-dominated and the red one is dominated.

they (1) work with a collection of solutions with different trade-offs among objectives, which are improving until obtaining a collection with optimal trade-offs, (2) can be easily adapted to the type of data of our domain, due to the flexible knowledge representation used, and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions. MOEAs are a subfield of Evolutionary Algorithms (EAs) (Holland, 1992), which are a paradigm that simulates the way nature acts (Cordón et al., 2001; Freitas, 2002): reproduction, recombination of the best individuals, mutation and selection. To solve a problem it is necessary to choose the best solution from the space of all feasible solutions, which is called search space. A search space is a huge area with a big amount of potential solutions, where there are regions with solutions of low quality and other regions with high quality solutions. EAs make it possible the exploration of the regions of the search space where the best solutions are placed. This kind of algorithms begin with a set of initial solutions called individuals that are improved through an iterative cycle based on the recombination and mutation of the individuals using special operators called crossover and mutation. In the case of clustering, each individual is considered a possible grouping of data. Concretely, MOEAs return a Pareto set (Corne et al., 2001) of solutions according to all the evaluated objectives and experts have to select the best one for their purpose. This Pareto set is a collection of non-dominated solutions with different trade-offs between objectives (see Figure 1.2). A solution S is non-dominated when there is no solution better than S for all the objectives, otherwise, the solution is dominated. EAs are one of the most powerful techniques for finding good solutions in a huge solution space where other approaches fail. However, the performance of these algorithms can be compromised in large databases due to their high computational and memory usage requirements.

This thesis is focused on proposing a new MC based on a MOEA that faces up three challenges identified from the literature related to MOEAs. These challenges are related to the definition and exploration of the search space where the EA is going to search the solutions, to the scalability of the algorithm with large data, and to the obtaining of the best clustering solution from the collection of potential solutions returned in the Pareto set. These issues applied to MC have not been studied in detail in the literature, in spite of some studies which tackle independently some of these challenges (Bacardit, 2004; Handl and Knowles, 2007; Hruschka et al., 2009). The next section presents the goals and contributions of this thesis.

1.3 Objectives and Contributions

The goal of this thesis is the definition and implementation of a new MC algorithm based on a MOEA that (1) optimizes the exploration of the search space, (2) allows to work with large data sets with reasonable computational time and memory usage, and (3) improves the selection of the best solution from the Pareto set of potential solutions. The proposed algorithm is called Clustering Algorithm based on multiObjective Strategies (**CAOS**) and it has been designed to successfully tackle the three aforementioned challenges. **CAOS** is graphically summarized in Figure 1.3. Next, the three MOEA challenges are explained more in detail:

Definition and exploration of the search space. The search space is an area of possible solutions that is explored in order to find the best solution. The definition of the search space is an important key to explore regions with solutions with high quality, that is regions where the best solutions are placed. To successfully define the search space where a MOEA is going to search the solutions, it is important to choose a suitable individual representation and the genetic operators related to it, according to the domain characteristics. In the related contribution, this stage consists in analyzing the best individual representation according to the specific problem to solve and the features of the data set that is going to be clustered.

Scalability with large data sets. MOEAs are computationally expensive and they may not obtain the results in a reasonable time and memory usage when they are applied to large data sets. This contribution proposes to scale-up the system splitting the original data set into several data subsets that are alternatively used in the system, in order to work with less data in each step of the evolutionary cycle while the performance is maintained.

Selection of the final clustering result. When the evolutionary cycle of a MOEA ends, it returns a collection of clustering solutions with different trade-offs among objectives. Thus, it is necessary to choose the most suitable solution because there is no solution better than the other ones according to the optimized objectives. In the related contribution, this stage is focused on selecting a final clustering result from the most interesting region of the collection of solutions according to some clustering validation techniques, which evaluate the clustering solutions according to features like the compactness, separation or shape of the clusters.

Finally, **CAOS** will be tested using artificial and real-world data sets. The first kind of data sets are used to test the behavior of **CAOS** in problems of different complexities. We understand as complex problems the domains where the suitable clusters meet some of these conditions: (1) they are not well separated, because they are overlapped or there is a small boundary between them, (2) they have arbitrary shapes, and (3) if an individual attribute has a low power to discriminate between them. On the other hand, the second kind of data sets are used to analyze the performance of **CAOS** in real-world problems of different domains, such as medicine, biology or image segmentation, among others. The real-world problems are extracted from the UCI repository ([Asuncion and Newman, 2010](#)) and from local repositories with data from network security and education.

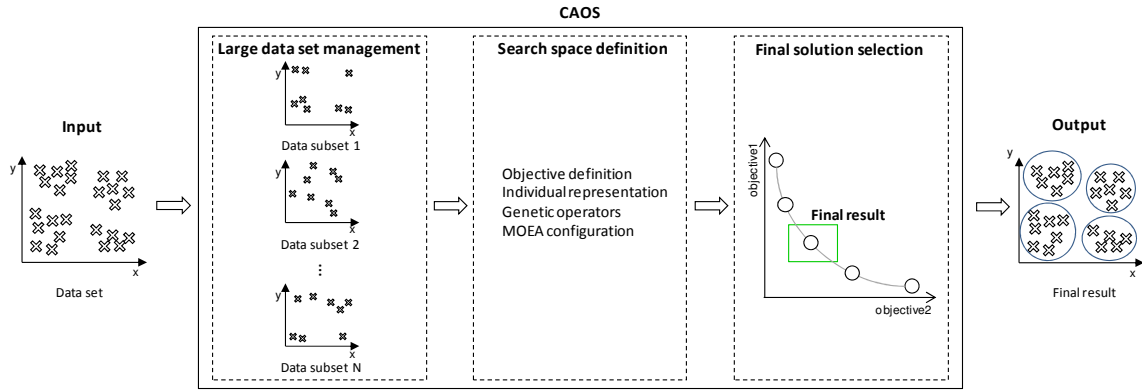


Figure 1.3: Overview of CAOS process. The input data is given to the CAOS system that faces up three MC challenges: (1) to manage large data sets dividing them in several data subsets; (2) to define the search space through the selection of the most suitable objectives, individual representation, genetic operators and MOEA parameters; and (3) to obtain the output of the system by selecting the final clustering result from the Pareto set of solutions.

All the results are analyzed using the recommendations pointed out by Demšar (Demšar, 2006) to perform the statistical analysis of the accuracy results, which is based on the use of nonparametric tests. The next section describes the roadmap followed along this dissertation.

1.4 Roadmap

This thesis has been organized in four parts: the first part contains an overview of the theoretical background needed to place the context of the dissertation, the second part describes the new multiobjective clustering algorithm proposed, the third part presents a practical application of the proposed algorithm in real-world problems, and the last part contains the conclusions and further work. Moreover, an appendix is annexed at the end of the dissertation.

The first part is divided into two chapters. Chapter 2 is focused on clustering techniques, presenting some clustering algorithms and the methods to validate them. Chapter 3 is centered on evolutionary algorithms, introducing this kind of techniques to optimize one or several objectives.

The second part is structured in four chapters. Chapter 4 is focused on the experimental framework of the thesis presenting the design of CAOS. The next three chapters describe the three contributions made in this thesis to face up the three aforementioned challenges. Chapter 5 analyzes three of the most used individual representations in evolutionary clustering applying them to CAOS in order to identify the situations where a representation is more suitable than the other ones. Chapter 6 is focused on scaling-up CAOS using techniques based on data subsets, presenting several methods to split up the complete data set and analyzing their performance in terms of accuracy and computational time. Chapter 7 presents an analysis of the strategies based on selecting the best solution from the collection of solutions returned by CAOS.

The third part is structured in three chapters. Chapter 8 presents the application of CAOS in the analysis of vulnerability assessments and compares it with previous strategies applied to the same problem. Chapter 9 is focused on the application of CAOS in the analysis of university degrees competences in order to help the educational experts to identify some improvements that should be made in the university degrees. Chapter 10 is centered on the knowledge organization in case-based

reasoning systems for organizing the case memory of these systems to improve their performance.

The last part contains one chapter that recapitulates the contributions of this thesis by summarizing, providing key conclusions, reviewing the main lessons extracted from this research, and presenting a proposal of future work lines.

The material presented in the eleven chapters is complemented with five appendices. Appendix A describes the conventional clustering algorithms used in some of the experimentations. Appendix B to Appendix E show the detailed results of the experiments done with large collection of data sets in Chapter 5, Chapter 6, Chapter 7 and Chapter 10, respectively.

Finally, it is important to highlight that the real-world applications of CAOS have been carried out in parallel to the development of the challenges that face up CAOS. Thus, the contributions of Chapter 8 to Chapter 10 do not include all the improvements presented in the contributions of Chapter 5 to Chapter 7.

1.5 Summary

This PhD thesis is framed on the graduate program in Information Technology and Management from La Salle at Ramon Llull University. It has been developed in the Research Group in Intelligent Systems and it has been supported by the *Generalitat de Catalunya*, the Commission for Universities and Research of the DIUE and European Social Fund (2010FI.B01084, 2011FI.B100022, 2010BE.01026).

Data Mining is the process of automatically extracting and discovering new, useful and understandable knowledge from huge volumes of data, with the aim of allowing experts to confront better the problems of the domain and to obtain wisdom. Clustering techniques are a way of extracting knowledge and they are based on grouping data according to a set of criteria, being the result a set of groups where each one contains a set of similar elements. Conventional clustering algorithms are based on grouping elements using only one objective, but they are not useful when several objectives have to be optimized. On the other hand, multiobjective clustering algorithms are based on grouping data according to a set of criteria evaluated simultaneously. This kind of algorithms are usually based in evolutionary algorithms due to their capabilities to solve this kind of problems.

This thesis is focused on the definition of a new multiobjective evolutionary clustering algorithm in order to face up three of the most important challenges related to this kind of algorithms: (1) definition and exploration of the search space, (2) scalability with large data sets, and (3) selection of the final clustering result. The performance of the proposed algorithm is tested in real and artificial problems with different clusters complexities and with arbitrary shapes in different domains, such as medicine, biology or image segmentation. All the results are analyzed using the recommendations pointed out by Demšar to perform the statistical analysis of the accuracy results, which is based on the use of nonparametric tests.

Part I

Theoretical Background

Chapter 2

Clustering

Clustering algorithms (Herrera et al., 2010) are based on grouping data according to a set of criteria, obtaining a set of groups (clusters) where each one contains similar elements. Thus, the elements in a cluster are similar among them and different from the elements of the other clusters and it provides to the experts a possible classification or categorization of the elements. These techniques are mechanisms to identify patterns and discover relationships to help experts to understand a specific domain. Clustering algorithms can be classified into different ways according to many points of view such as: the relation between the clusters (partitional or hierarchical), how they are structured (center-based, search-based and graph-based, among others), the relation of the class with the cluster (hard clustering and fuzzy clustering, among others), or the criteria used to build the clusters (conventional clustering, ensemble clustering, or multiobjective clustering). If we address the last criteria, conventional clustering is based on optimizing an objective function for assessing the quality of groups of elements. On the other hand, ensemble clustering and multiobjective clustering use a set of objectives to promote the definition of clusters. The main difference between both approaches is the procedure used to build the clusters. Ensemble clustering divides the procedure into two phases: (1) application of many clustering algorithms following different single objectives; and (2) combination of the last results to create the final clustering. This last step is quite complex to achieve and it is usually inefficient because the objectives can be partially contradicting. In contrast, multiobjective clustering techniques create solutions based on multiple criteria simultaneously. Concretely, each clustering solution is evaluated for each objective, and the final result is a collection of these solutions with different trade-offs among objectives represented in a Pareto set.

2.1 Taxonomy

Clustering algorithms are able to group data from different points of view, and their suitability mainly depends on the application domain. For this reason, it is important to select the algorithm according to the data typology and the features of the application domain. There is not a single criterion to classify the clustering algorithms, so they can be classified according to many criteria (Gan et al., 2000; Witten and E. Frank, 2011; Duda et al., 2000): (1) the search strategy to find the clusters, (2) the relationships between the clusters, (3) the instances distribution into the clusters, and (4) the optimization of the clusters. Next, each one of these points of view is described:

- The search strategy is related to how the clusters are found. The most used strategies are:
 - **Center-based.** These algorithms define each cluster using a prototype, which is an instance of the data or an artificial instance created using the data. They assign the instances of the data set to the closest prototype, trying to improve the clustering solution in each iteration until a convergence is achieved. These algorithms had some limitations like they are often sensitive to the initialization, they may fall in a local optimal solution, and they can only find clusters with convex shapes, thus they cannot find clusters with arbitrary shapes.
 - **Graph-based.** This kind of algorithms are based on graph theory. For example, a graph or hypergraph is built and then some heuristics are applied to partition it. These algorithms are iterative and try to improve the clustering solution in each iteration until convergence is achieved. The number of clusters is not required and they can have arbitrary shapes. These algorithms are sensitive to the initialization and they may fall in a local optimal solution.
 - **Model-based.** These algorithms assume that the data are generated by a mixture of probability distributions in which each one represents a different cluster. The distributions are estimated from the data and each data instance is assigned to each one. They usually are iterative and try to improve the clustering solution in each iteration until a convergence is achieved. This approach is sensitive to the initialization and they may fall in a local optimal solution. For example, if normal distributions are assumed, the instances are assigned according to the mean and deviation of each distribution.
 - **Search-based.** This technique is a complement of the previous strategies. The previous strategies may not be able to find the globally optimal clustering that fits the data set because they are Greedy algorithms (Cormen et al., 2001), thus they choose the best optimal in each iteration being possible to find a local optimum at the end of the process. This strategy tries to search in the overall solution space and find a globally optimal clustering that fits the data set, using global optimization techniques like genetic algorithms, ant colony optimization or simulated annealing. The main drawback of these techniques is that are very time consuming.
 - **Density-based.** These algorithms define clusters as dense regions separated by low-density regions. They are not iterative algorithms, thus they need only to manage the data once, and they are able to handle noise. The number of clusters is not required and they can find arbitrarily shaped clusters.
 - **Subspace clustering.** This strategy consists in finding clusters in each dimension identifying dense units. The final clusters are found overlapping the clusters found in each dimension.
- The relationship that exists between the clusters (see Figure 2.1) is classified as:

- **Partitional.** It divides the data splitting the space in independent clusters.
- **Hierarchical.** It creates a hierarchical relationship between clusters by obtaining clusters that contain other clusters inside. A hierarchical clustering is often represented as a dendrogram (Manning and Schuetze, 2000) (see Figure 2.2). There are two types of hierarchical clustering:
 - * **Divisive.** This technique considers that all the instances are in one cluster, and tries to divide it until each instance is in a single cluster.
 - * **Agglomerative.** This one considers that each instance is a different cluster and tries to merge the clusters until all the instances are in a cluster.
- There are two types of instance distribution into the clusters (see Figure 2.3):
 - **Hard.** Each instance of the data set belongs to only one cluster.
 - **Fuzzy.** The instances can belong to more than one cluster. In this case, the instances have assigned a membership grade which indicates the degree to which the objects belong to each cluster.

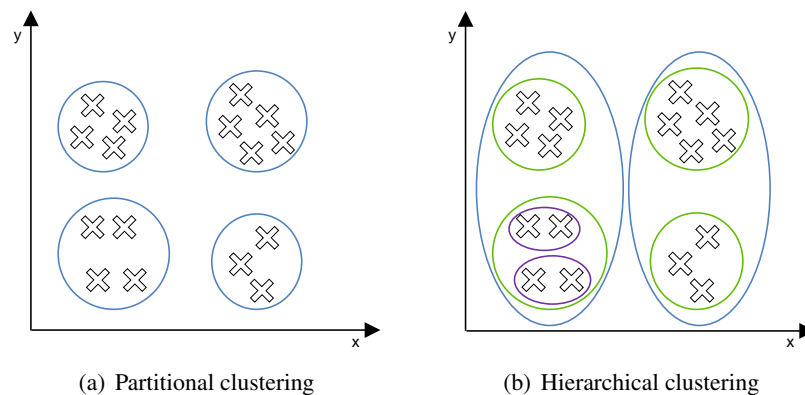


Figure 2.1: Example of partitional and hierarchical clustering results from the same elements. Each color corresponds to a different level of hierarchy.

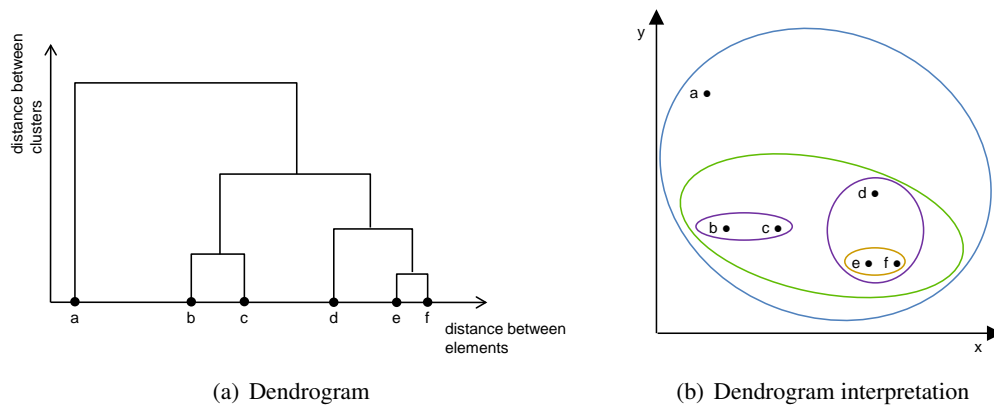


Figure 2.2: Example of a dendrogram to represent hierarchical clustering and its interpretation. Each color corresponds to a different level of hierarchy.

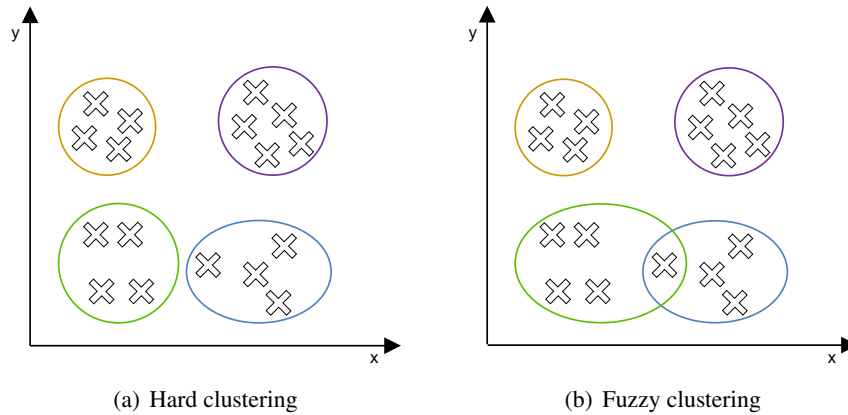


Figure 2.3: Example of hard and fuzzy clustering results from the same elements.

- The last feature takes into account the criteria optimization of the clusters. Some of the search strategies explained, need to evaluate the convergence of the algorithms at the end of each iteration according to:
 - **Single objective.** It optimizes only one objective represented as a single criterion or a single equation of several criteria.
 - **Several objectives.** It consists in optimizing several criteria expressed in different objectives. There are two kind of optimization approaches:
 - * **Ensemble clustering.** Each objective is optimized using a single-objective technique that creates a clustering solution for each one. Then, all the clustering solutions obtained are combined in a single solution that tries to achieve a trade-off among all the solutions. However, a good trade-off among objectives may not be achieved combining the solution obtained for each objective (Strehl and Ghosh, 2002; Law et al., 2004).
 - * **Multiobjective clustering.** It evaluates each objective simultaneously for each clustering solution (Ferligoj and Batagelj, 1992; Handl and Knowles, 2007). The clustering solutions are stored in a collection of solutions where each one has a different trade-off among objectives. Thus, it obtains several clustering solutions with different adjustments of the objectives.

These approaches are important in order to select the most suitable clustering algorithm according to the expected results. Figure 2.4 describes a possible taxonomy based on the previous division, where a set of the most representative algorithms from each search strategy are mapped and described in the next section.

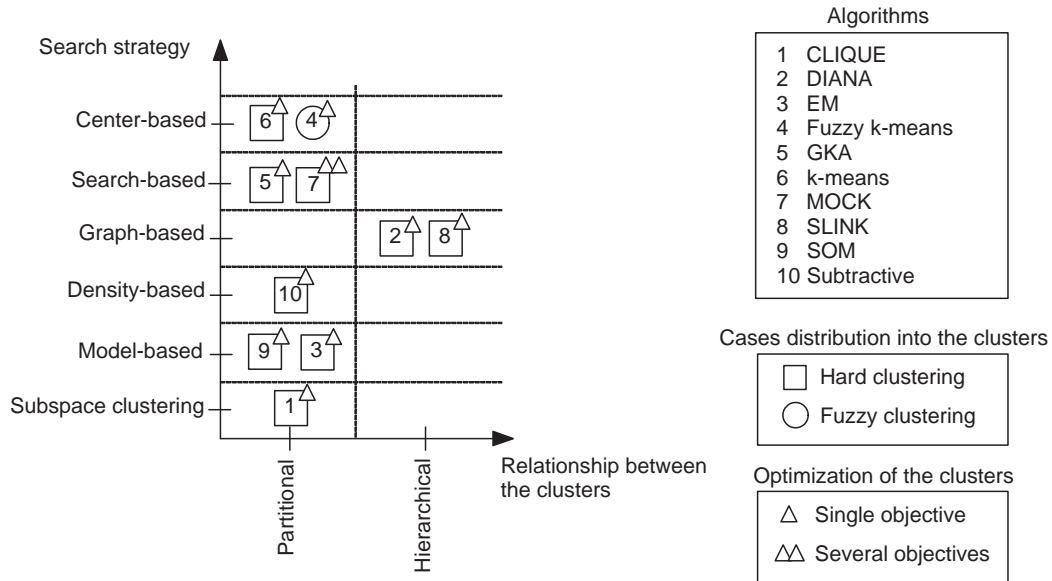


Figure 2.4: Taxonomy proposed for classifying clustering algorithms.

2.2 Algorithms

There are many different clustering algorithms with different features, and the suitable algorithm to solve a problem depends on the characteristics of the domain and the kind of expected clustering solution. Next, some of the most well-known and used algorithms are briefly explained to have a wide view about how clustering algorithms can work taking into account their different properties:

- ***k*-means** (MacQueen, 1967). It consists in grouping the instances into k circular clusters according to the distance between them and the center of the cluster represented by an instance (centroid). There are many variants like x -means (Pelleg and Moore, 2000) which sets automatically the number of clusters, or k -modes (Huang, 1997) which works with categorical attributes.
- **Fuzzy *k*-means** (Bezdek, 1974). It works like k -means but each instance has a cluster membership degree assigned. Thus, each instance can belong to several clusters. Some variants of this algorithm are c -means (Bezdek, 1981), which sets automatically the number of clusters; or Fuzzy k -modes (Huang and Ng, 1999) that can work with categorical attributes.
- **Genetic *k*-means (GKA)** (Krishna and Narasimha, 1999). It follows the philosophy of k -means but doing a global search, being able to avoid local minimum solutions obtaining better quality solutions. The global search is done with a genetic algorithm. Some variants of this algorithms are Genetic k -modes and Genetic Fuzzy k -means, which follow the philosophy of the k -modes and Fuzzy k -means algorithms.
- **Expectation-Maximization (EM)** (Dempster et al., 1977). It assumes that the data can be modeled by a mixture of probability distributions, being each cluster a different distribution.

First, the algorithm creates an initial model θ_0 estimating it from the instances of the data set. Second, it computes the membership probability of the instances to each cluster according to the model θ_t through the step called Expectation step (E-step). After this, it uses the membership probability to obtain another model $\theta_t + 1$ called Maximization step (M-step). The two last steps are repeated until reaching a local maximum of the log-likelihood of the data.

- **SLINK** (Sibson, 1973). It is based on the single-link concept (Johnson, 1967). Initially, each instance is considered an independent cluster and the process consists in merging all the instances until they are assigned to a single cluster. The merge step combines the two clusters whose two closest members have the smallest distance. The algorithm is based on graph theory due to the fact that the instances can be represented as nodes and the distances between the instances can be the links of the nodes, applying a strategy similar to the Minimum Spanning Tree (MST) algorithm (Prim, 1957) to obtain the clustering solution. Other similar approaches are Complete-link (Johnson, 1967), based on merge the two clusters whose two farthest members have the smallest distance; and the Average-link (Johnson, 1967), which merge the two clusters whose centers have the smallest distance.
- **Subtractive Clustering** (Chiu, 1994). It is an efficient method for estimating cluster centers in a greedy fashion. The idea is to identify the instance from the data that can be considered as a potential center of a new cluster considering that an instance with many neighboring data points will have a high potential value. After this, the nearest instances are assigned to the new cluster. These two steps are repeated until each instance is assigned to one cluster.
- **Divisive Analysis (DIANA)** (Kaufman and Rousseeuw, 1990). It is a hierarchical divisive clustering algorithm. Initially, all the instances are assigned to the same cluster. Next, each cluster is split until it has only one instance, creating a top-down relation between them. The split step consists in dividing the cluster with the largest diameter, which is defined to be the largest distance between two instances in it.
- **Self Organizing Map (SOM)** (Kohonen, 1990). It is inspired on neuronal networks. It projects the original N-dimensional space to another more reduced to identify hidden relationships among data. A variant of this algorithm is the Growing Self-Organizing Map (GSOM) (Alahakoon et al., 1998) which identifies the most suitable map size.
- **CLIQUE** (Agrawal et al., 1998). It was the first subspace clustering algorithm. It is able to identify dense clusters in subspaces of maximum dimensionality using two parameters. The first one specifies the number of intervals in each dimension and the second one specifies the density threshold. The clusters are represented by a minimal description in the form of a disjunct normal form (DNF) expression. One disadvantage of this algorithm is that it can only find clusters embedded in the same subspace.

- **Multiobjective Clustering with Automatic k -Determination (MOCK)** (Handl and Knowles, 2004b). It can be considered the first multiobjective clustering algorithm using a multiobjective evolutionary algorithm to obtain a trade-off between all the objectives evaluated. The results is a collection of solutions with different trade-offs between two objectives, where there is no solution better than the others for all the objectives.

Even the majority of the clustering algorithms are focused on optimizing only an objective, the capability of using more than one criterion to identify groups of element offers more flexibility to successfully tackle domains with arbitrary geometric shapes. The next section introduces the techniques that optimize several objectives to improve the results of the conventional clustering techniques in complex domains.

2.3 When One Criterion Is Not Enough

Figure 2.5(a) shows some elements characterized by two attributes (x and y) that have to be clustered. There is more than one solution because it is possible to group them using different points of view such as the shape of elements or minimizing the number of clusters, among others. However, they could be summarized in one single objective due to the fact that they do not affect to each other. This objective can be expressed in a conjunction of the following criteria: (1) to obtain clusters where each cluster contains elements of the same shape and (2) the number of clusters has to be minimized. In this scenario, Figure 2.5(d) shows one possible solution optimizing this composed objective using a conventional clustering algorithm.

Now let's imagine a situation where the desired optimization is based on the shape and size of the clusters. A clustering solution is considered good when the instances of each cluster are very similar among them (low intra-cluster variance) and very different to the elements of the other clusters (high inter-cluster variance). Figure 2.6(a) shows another scenario where elements have to be clustered according x and y using the two aforementioned criteria. Each criterion can be

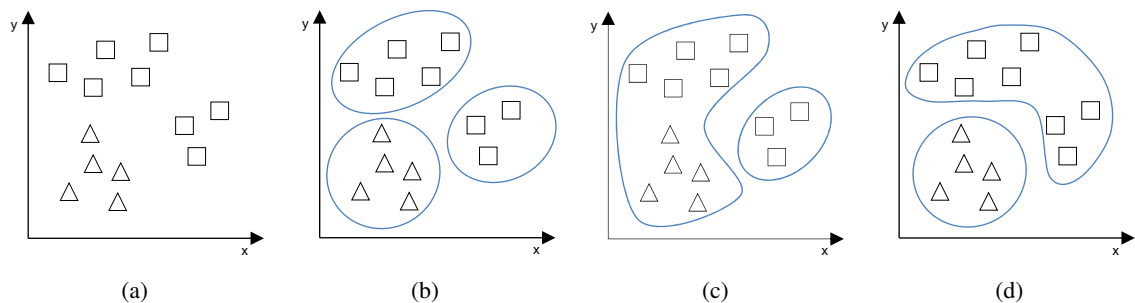


Figure 2.5: Example 1 of clustering problem. (a) Elements to be clustered. (b) Elements clustered optimizing one criterion: the same shape of the elements in each cluster. (c) Elements clustered optimizing one criterion: to minimize the number of clusters. (d) Elements clustered optimizing two criteria: the same shape of the elements in each cluster and to minimize the number of clusters.

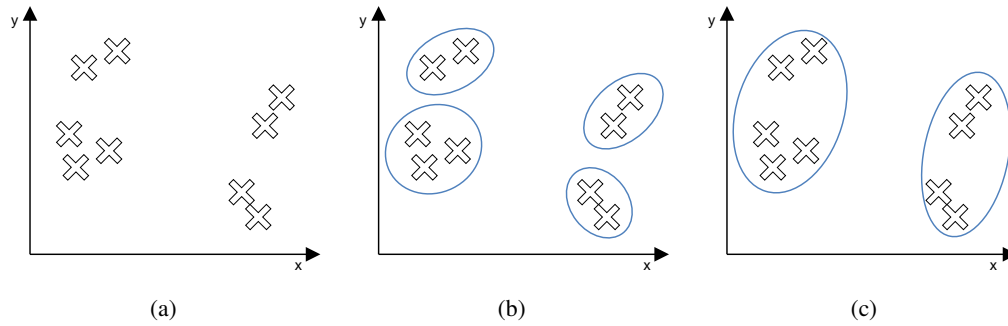


Figure 2.6: Example 2 of clustering problem. (a) Elements to be clustered. (b) Elements clustered optimizing the intra-cluster variance. (c) Elements clustered optimizing the inter-cluster variance.

independently optimized (see Figure 2.6(b) and 2.6(c)) but it is impossible to optimize both criteria at the same time because when one criterion is optimized, the other one gets worse and vice versa. For this reason, each criterion should become an independent objective to be optimized. It is in this kind of problems where conventional clustering algorithms are not suitable due to their impossibility of optimizing several objectives at the same time and other approaches need to be considered.

Ensemble Clustering (Strehl and Ghosh, 2002; Topchy et al., 2004) obtains a clustering solution from the combination of several conventional clustering obtained through different ways, such as for example different optimization objectives, different clustering algorithms, or different samples of a large data set. In the case of objectives optimization, some clustering algorithms can be independently applied to optimize different objectives to subsequently obtain a result with the consensus of all the clustering solutions found. The combination of the solutions is not trivial, and the main drawback is that a good trade-off among objectives may be is not achieved with the combination of the solutions found independently optimizing each objective. Figure 2.7(a) shows a possible application of Ensemble Clustering to the example of Figure 2.6(a), where each objective is independently optimized and the two resulting clusterings are merged in a single clustering solution.

On the other hand, the Multiobjective Clustering technique (Ferligoj and Batagelj, 1992; Handl and Knowles, 2007) is focused on optimizing all the objectives simultaneously. The final solution is a collection of clustering solutions with different trade-offs between objectives represented in a Pareto set based in a Pareto dominance strategy. For example, Figure 2.7(b) shows the Pareto set obtained applying Multiobjective Clustering to the example of Figure 2.6(a) and optimizing the inter-cluster and intra-cluster variances. In a Pareto set all the solutions are non-dominated, this means that there is no solution in it worse than the other ones for all the objectives. In Figure 2.7(b) it can be observed that there is no solution better than the other for the two objectives, and each one offers a different trade-off between the objectives. When a Pareto is graphically represented, each one of the axes represents an objective to optimize. In the aforementioned figure, the axes of the Pareto set represent the two objectives to optimize, in this case they have to be minimized. Due to

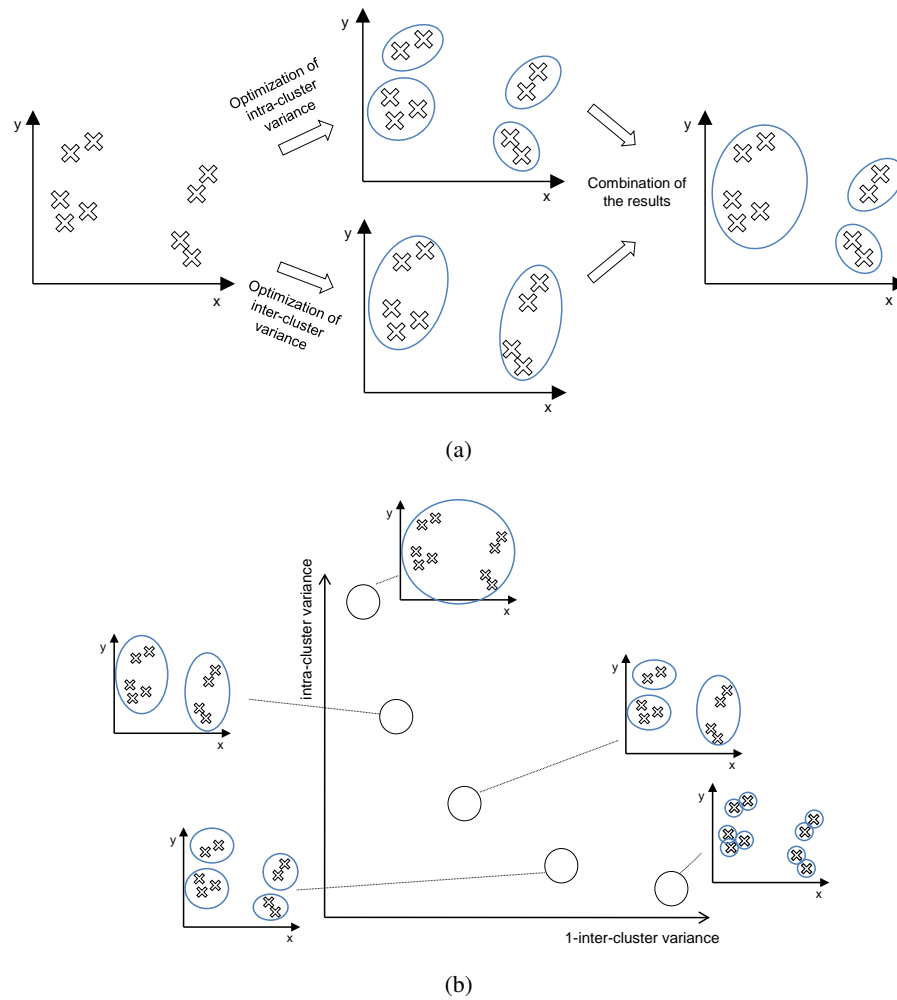


Figure 2.7: Ensemble Clustering and Multiobjective Clustering process for the clustering problem described in Example 2. (a) Process to obtain an Ensemble Clustering solution. (b) Multiobjective Clustering Pareto set of solutions.

the range of the inter-cluster variance is between 0 and 1 and should be maximized, the objective represented is $1 - \text{inter-cluster variance}$.

2.4 When Many Criteria is Too Much

The multiobjective algorithms based on Pareto dominance are useful to optimize problems with few objectives because the Pareto dominance is less effective when the number of objectives is increased and the convergence of the approaches decreases (Farina and Amato, 2002). The reason is because an increment of the number of objectives increases the proportion of non-dominated individuals due to the fact that the majority of the individuals are not worse for all the objectives than the other solutions of the Pareto set (see Figure 2.8). When the search progresses, the Pareto set is rapidly saturated with non-dominated solutions, which cannot be discriminated because there is no solution better than the other ones for all the objectives. Thus, the Pareto set has a big amount

of non-dominated solutions of low quality and the search process is randomly performed.

Multiobjective problems where more than three objectives are simultaneously optimized are called Many-objective Optimization Problems (MOP) (Ishibuchi et al., 2008) in the literature. The main idea of the techniques that solve these problems is to relax the concept of Pareto dominance used in multiobjective techniques in order to obtain a more restrictive Pareto set. There are several strategies to do this (Fabre et al., 2010):

- **Ranking Composition Methods** defines a list of fitness values for each objective and for each solution. These lists are individually sorted, obtaining different ranking positions of each solution for each objective. Finally, the different ranking positions of each solution are composed into a single ranking which reflects its quality. The final ranking can be obtained combining the rankings for each objective in different ways. For example, it can be obtained with the average rank of each solution according to the rankings of all the objectives, called Average Ranking (Bentley and Wakefield, 1998); or taking into account the best ranking position of all the objectives, called Maximum Ranking (Bentley and Wakefield, 1998).
- **Relaxed Forms of Dominance** allows a solution to dominate another one without being better in all the objectives. Generally, these methods can accept a detriment in some objectives if the solution presents a considerable improvement in the other objectives. It exists several strategies to do this such as the α -domination (Kokolo et al., 2001), the L -dominance (Zou et al., 2008), or the Value dominance (Le and Silva, 2007), among others. The α -domination strategy sets upper/lower bounds of trade-off rates among objectives in order to allow a solution S to dominate a solution S' if S is slightly inferior in an objective but largely superior in some other objectives. The L -dominance strategy counts the number of objectives in which

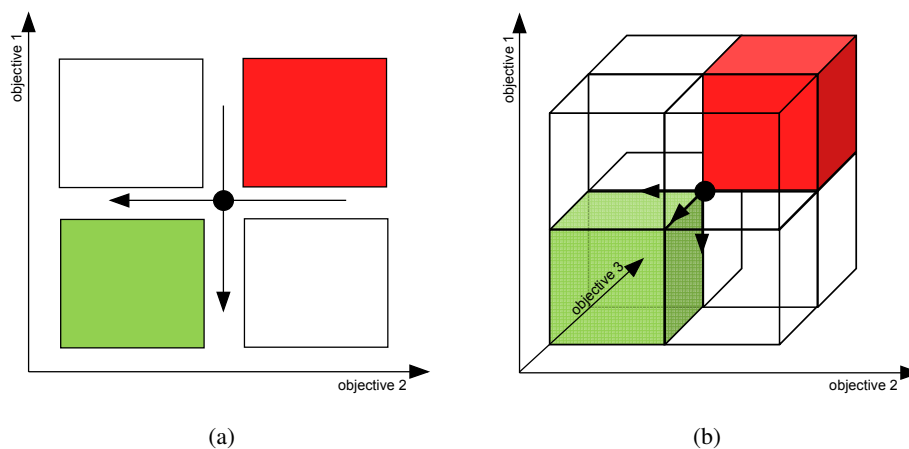


Figure 2.8: Schematic view of Pareto-dominance (Farina and Amato, 2002) based on partial order in problems with (a) 2 objectives and (b) 3 objectives when a candidate solution (●) is considered. Considering that the objectives have to be minimized, the green areas dominate the candidate solution, the red areas are dominated by the candidate solution, and the white areas contain equivalent solutions to the candidate one.

a solution S is respectively better (n_b), equal and worse (n_w) than another solution S' . A solution S L -dominates S' if n_b is higher than n_w and it is not worse than S' considering all the objectives at the same time (average, p-norm, etc.). The Volume dominance strategy is based on the volume of the objective space that a solution dominates. The dominated volume of a solution is defined as the region for which all its feasible solutions are dominated by it.

2.5 Validation Techniques

Although clustering techniques are usually applied to unsupervised problems where the ideal clusters are not known, it is necessary to define mechanisms to evaluate the quality of the solution. For this reason, clustering validation methods are used in order to obtain a quantitative evaluation of the results. However, it must be emphasized that these methods are only a tool at the disposal of the expert in order to evaluate the resulting clustering. The validation methods analyzed in this section are focused on partitional and hard clustering, because hierarchical and fuzzy clustering algorithms are out of the scope of this thesis. The reader is referred to (Theodoridis and Koutroumbas, 2008).

There are three possible clustering validation approaches (Halkidi et al., 2001; Halkidi et al., 2002a; Legány et al., 2006). The first one is called external criteria and the idea is to evaluate a clustering result comparing it with a structure of the data set obtained without applying any clustering algorithm. The second approach is called internal criteria and the objective is to evaluate a clustering result comparing it with only quantities and features inherent to the data set. The third approach is called relative criteria and it is based on comparing a clustering result with other results obtained from the application of the same clustering algorithm with different parameter values, or of other clustering algorithms.

The cluster validation methods based on external or internal criteria are based on statistical hypothesis testing, and their major drawback is their high computational cost. Moreover, these two approaches measure the degree to which a data set confirms an a-priori specified scheme that can be inherent to the data set or an intuitive structure of the data. On the other hand, relative criteria methods find the hypothetical best clustering scheme from several clustering results obtained with different parameters or clustering algorithms without using statistical tests, so they are less computationally expensive. Next sections briefly introduce the statistical testing process followed by the three clustering validation approaches and their description.

2.5.1 Statistical Testing in Clustering Validation

The aim of the statistical testing in Clustering Validation is to test the Null Hypothesis (H_0) that the data set is randomly structured. There are several methods to accept or reject H_0 using simulations (Theodoridis and Koutroumbas, 2008), and we follow the process based on Monte Carlo techniques (Shreider, 1964; Sobo, 1984). Monte Carlo techniques rely on execute the simulation process at hand using a sufficiently large number of computer-generated data. Thus, the procedure has two steps:

1. To generate r reference data sets under the random hypothesis, that is, a collection of data sets that models a random structure. There are different ways to build the reference data sets under H_0 that are summarized in (Theodoridis and Koutroumbas, 2008). In our explanation we can assume that the reference data sets are generated selecting randomly the instances of the original data set (Random Position Hypothesis).
2. To compare the value that results from the original data set (q) with the values (q_i) obtained from the r reference data sets using an appropriate statistic (statistical index), whose values are indicative of the structure of a data set.

The rejection of H_0 is done according to Equation 2.1 (right-tailed statistical test), Equation 2.2 (left-tailed statistical test) or Equation 2.3 (two-tailed statistical test) using ρ as the significance level, which usually has the value 0.05. If the condition corresponding to q is not achieved, H_0 is accepted. If H_0 is rejected, it can be declared that the data set cannot be considered randomly structured.

$$\text{Reject } H_0 \text{ if } q \text{ is greater than } (1 - \rho) \cdot r \text{ of the } q_i \text{ values} \quad (2.1)$$

$$\text{Reject } H_0 \text{ if } q \text{ is smaller than } \rho \cdot r \text{ of the } q_i \text{ values} \quad (2.2)$$

$$\begin{aligned} \text{Reject } H_0 \text{ if } q \text{ is greater than } \frac{\rho}{2} \cdot r \text{ of the } q_i \text{ values and} \\ \text{smaller than } (1 - \frac{\rho}{2}) \cdot r \text{ of the } q_i \text{ values} \end{aligned} \quad (2.3)$$

2.5.2 External Criteria

The external criteria approach can be used for two purposes: (1) to compare a clustering result (C) obtained with a clustering algorithm with a predetermined partition (P) of the data set without using any clustering algorithm; and (2) for measuring the degree of agreement between a predetermined partition (P) and the proximity (similarity) matrix (S) of the data set, which stores the distances between each one of the instances. To make these comparisons it is necessary to define appropriate statistical indexes to be used for the hypothesis test.

Comparison of Clustering C with Partition P

The degree to which C matches P is obtained comparing the assignation of pairs of elements in both structures. Given the pair of elements (x_v, x_u) , we refer to it as SS if both elements belong to the same cluster in C and to the same group in P , DD if both elements belong to different clusters in C and to different groups in P , SD if the elements belong to the same cluster in C and to different

groups in P , and DS if the elements belong to different clusters in C and to the same group in P . Some statistical indexes that carry out this process are the following:

- **Rand index.** (Rand, 1971)

$$R = \frac{a + d}{M} \quad (2.4)$$

- **Jaccard Coefficient.** (Jaccard, 1901)

$$J = \frac{a}{a + b + c} \quad (2.5)$$

- **Fowlkes and Mallows index.** (Fowlkes and Mallows, 1983)

$$FM = \sqrt{\frac{a}{a + b} \cdot \frac{a}{a + c}} \quad (2.6)$$

- **Adjusted Rand index.** (Yeung and Ruzzo, 2001)

$$AR = \frac{\sum_{i=1}^{c_p} \sum_{j=1}^{c_c} \binom{n_{ij}}{2} - \left[\sum_{i=1}^{c_p} \binom{n_i}{2} \cdot \sum_{j=1}^{c_c} \binom{n_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_{i=1}^{c_p} \binom{n_i}{2} + \sum_{j=1}^{c_c} \binom{n_j}{2} \right] - \left[\sum_{i=1}^{c_p} \binom{n_i}{2} \cdot \sum_{j=1}^{c_c} \binom{n_j}{2} \right] / \binom{n}{2}} \quad (2.7)$$

In Equations 2.4 to 2.6, a , b , c and d are the number of pairs of elements considered as SS , SD , DS and DD respectively; N is the number of elements in the data set; and M is the number of possible pairs ($a + b + c + d$). In Equation 2.7, c_c is the number of clusters of the evaluated solution C , c_p is the number of groups in P , n is the number of elements in the data set, n_{ij} is the number of data items that have been assigned to group i in P and cluster j in C , n_i is the number of instances assigned to class i and n_j is the number of instances assigned to cluster j . The value range of these indexes is between 0.0 and 1.0, indicating larger values a higher agreement between C and P , that is, all the corresponding statistical tests are right-tailed.

Furthermore, the degree to which C matches P can be also measured using indexes based on the correlation between two matrices. Depending on the indexes used, the two matrices can be calculated in a different way. For example, Equation 2.10 and 2.11 show how they are calculated in the following two indexes:

- **Hubert's Γ statistic.** (Hubert and Schultz, 1976)

$$\Gamma = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^N X(i, j)Y(i, j) \quad (2.8)$$

- **Normalized Γ statistic.** (Hubert and Schultz, 1976)

$$\hat{\Gamma} = \frac{\frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^N (X(i, j) - \mu_x)(Y(i, j) - \mu_y)}{\sigma_x \sigma_y} \quad (2.9)$$

Where N is the number of elements in the data set; $M = N(N - 1)/2$; and $\mu_x, \mu_y, \sigma_x, \sigma_y$ are the means and deviations of both matrices. The Hubert's Γ statistic has a value range between 0.0 and 1.0 and the Normalized Γ statistic has a value range between -1.0 and 1.0. Larger values of both indexes indicates a higher agreement between C and P , so there are right-tailed statistical tests.

$$X(i, j) = \begin{cases} 1, & \text{if elements } x_i \text{ and } x_j \text{ belong to different cluster in } C \\ 0, & \text{otherwise.} \end{cases} \quad (2.10)$$

$$Y(i, j) = \begin{cases} 1, & \text{if elements } x_i \text{ and } x_j \text{ belong to different group in } P \\ 0, & \text{otherwise.} \end{cases} \quad (2.11)$$

After applying any of the aforementioned indexes, called I , it is necessary to test the null hypothesis H_0 to declare that clustering C is not a random structure of the data. As discussed above, Algorithm 2.1 details the procedure based on Monte Carlo methodology (Theodoridis and Koutroumbas, 2008) to test H_0 . Considering q the value of the corresponding statistical index I for P and C , if H_0 is rejected, it cannot be considered that clustering C is a random structure of the data, and it has the similarity degree q with P . If H_0 is accepted, it is considered that clustering C is a random structure of the data and has to be considered a bad data grouping. It is important to highlight that depending on the statistical index used, the H_0 testing and the similarity degree between C and P can change, obtaining different conclusions for them.

<pre> 1 for $i = 1 \rightarrow r$ do 2 Generate a data set D_i of N elements in the area of interest of the original data set D, so that the vectors are uniformly distributed in it 3 Assign each vector $y_i^j \in D_i$ to the group where the x_j belongs, according to the structure imposed by P 4 Run the same clustering algorithm, used for obtaining C, on D_i and let C_i be the resulting clustering 5 Compute the value q_i of the corresponding statistical index I for P and C_i 6 Accept or reject H_0 according to Equation 2.1, 2.2 or 2.3, depending on the type of statistical index I used </pre>

Algorithm 2.1: Statistical test based on Monte Carlo techniques to test the null hypothesis that clustering C is randomly structured according to a prespecified partition P .

Assessing the Agreement Between Partition P and Proximity Matrix S

This analysis is done in order to measure the degree to which the proximity matrix S of a data set matches a prespecified partition P of the same data set. The statistical indexes used to do this have

to be based on the correlation between two matrices, so we can use the two indexes Hubert's Γ statistic and Normalized Γ statistic, but replacing X with the proximity matrix S .

After select an statistical index I and apply it to P and S to obtain q , it is necessary to test the null hypothesis H_0 to declare that partition P cannot be considered a random structure of the data. In this case, the procedure used to test H_0 is summarized in Algorithm 2.2. If H_0 is rejected, it can be declared that, using the statistical index I , P has the similarity degree q with S .

```

1 for  $i = 1 \rightarrow r$  do
2   Generate a data set  $D_i$  of  $N$  elements in the area of interest of the original data set  $D$ , so that the vectors are uniformly distributed in it
3   Calculate the proximity matrix  $S_i$  of the new data set  $D_i$ 
4   Assign each vector  $y_i^j \in D_i$  to the group where the  $x_j$  belongs, according to the structure imposed by  $P$ 
5   Compute the value  $q_i$  of the corresponding statistical index  $I$  for  $P$  and  $S_i$ 
6 Accept or reject  $H_0$  according to Equation 2.1, 2.2 or 2.3, depending on the type of statistical index  $I$  used

```

Algorithm 2.2: Statistical test based on Monte Carlo techniques to test the null hypothesis that the prespecified partition P is randomly structured according to the similarity matrix S .

2.5.3 Internal Criteria

This kind of methods try to verify if a clustering structure C obtained by a clustering algorithm fits the data set using only information inherent in the data such as the proximity matrix S . Due to the fact that the comparison is made with the matrix S , the statistical indexes used are based on the correlation between two matrices such as the indexes Hubert's Γ statistic and Normalized Γ statistic. In these indexes, the matrix X is the same indicated in Equation 2.10 and Y is replaced with the proximity matrix S .

One statistical index I is selected from the aforementioned indexes and it is applied to C and S obtaining q . Finally, it is necessary to test the null hypothesis H_0 to declare that partition C cannot be considered a random structure of the data. In this case, the procedure used to test H_0 is summarized in Algorithm 2.3. If H_0 is rejected, it can be declared that, using the statistical index I , C has the similarity degree q with S .

```

1 for  $i = 1 \rightarrow r$  do
2   Generate a data set  $D_i$  of  $N$  elements in the area of interest of the original data set  $D$ , so that the vectors are uniformly distributed in it
3   Calculate the proximity matrix  $S_i$  of the new data set  $D_i$ 
4   Assign each vector  $y_i^j \in D_i$  to the cluster where the  $x_j$  belongs, according to clustering  $C$ 
5   Compute the value  $q_i$  of the corresponding statistical index  $I$  for  $C_i$  and  $S_i$ 
6 Accept or reject  $H_0$  according to Equation 2.1, 2.2 or 2.3, depending on the type of statistical index  $I$  used

```

Algorithm 2.3: Statistical test based on Monte Carlo techniques to test the null hypothesis that clustering C is randomly structured according to the similarity matrix S .

2.5.4 Relative Criteria

The main drawback of the last two approaches is that they are computationally expensive due to the fact that they required the use of statistical tests. Relative criteria methods do not need the use of statistical tests to select the best solution from a collection of clustering results obtained with different parameters or clustering algorithms. Nevertheless, they do not identify if a clustering result is not a random structure and if it is good. They can only identify if a solution is better than others. To carry out this identification, the so called validation indexes are used. These validation indexes are usually based on quantities and features inherent to the data set. Some of these indexes are:

- **Modified Hubert Γ statistic.** (Gan et al., 2000)

$$\Gamma = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^N P(i, j)Q(i, j) \quad (2.12)$$

- **Davies-Bouldin index.** (Davies and Bouldin, 1979)

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{\substack{j=1 \\ j \neq i}}^n \left\{ \frac{S_n(C_i) + S_n(C_j)}{d(v_i, v_j)} \right\}, \text{ where} \quad (2.13)$$

$$S_n(C_i) = \frac{1}{|C_i|} \sum_{x \in C_i} d(x, v_i) \quad (2.14)$$

- **Dunn index.** (Dunn, 1974)

$$Dn = \min_{i=1}^n \left\{ \min_{j=i+1}^n \left\{ \frac{d(C_i, C_j)}{\max_{k=1}^n \{diam(C_k)\}} \right\} \right\}, \text{ where} \quad (2.15)$$

$$diam(C_k) = \max_{x, y \in C_k} \{d(x, y)\}, \text{ and}$$

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{d(x, y)\} \quad (2.16)$$

- **Silhouette index.** (Rousseeuw, 1987)

$$Sil = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{|C_i|} \sum_{x \in C_i} \left(\frac{b(C, i, x) - a(C_i, x)}{\max\{a(C_i, x), b(C, i, x)\}} \right) \right), \text{ where} \quad (2.17)$$

$$a(C_i, x) = \frac{1}{|C_i| - 1} \sum_{y \in C_i} d(x, y), \text{ and}$$

$$b(C, i, x) = \min_{\substack{j=1 \\ j \neq i}}^n \frac{1}{|C_j|} \sum_{y \in C_j} d(x, y)$$

- **Calinski-Harabasz index.** (Caliński and Harabasz, 1974)

$$CH = \frac{B \cdot (m - n)}{W \cdot (n - 1)}, \text{ where} \quad (2.18)$$

$$B = \sum_{i=1}^n \sum_{x \in C_i, y \notin C_i} d(x, y), \text{ and}$$

$$W = \sum_{i=1}^n \sum_{x \in C_i} d(x, v_i)$$

In Equation 2.12, N is the number of elements in the data set; $M = N(N - 1)/2$; P is the proximity matrix S of the data set; and Q is matrix whose (i, j) element, $Q(i, j)$, is equal to the Euclidean distance $d(v_{ci}, v_{cj})$ between the representatives of the clusters where elements i and j belong. The value range of this index is between 0.0 and 1.0, indicating larger values a higher quality of the clustering. In the other four equations m is the number of examples in the training data set; n is the number of clusters; C_i is the set of instances belonging to cluster i ; $|C_i|$ is the number of elements in C_i ; v_i is the centroid of C_i ; $d(x, y)$ is the Euclidean distance between x and y elements; $nn(x, i)$ returns the i th nearest element of x according to $d(x, y)$; and ℓ is the amount of nearest elements taken into account. In Calinski-Harabasz index, B and W correspond to the between cluster sum of squares and to within cluster sum of squares respectively. They can be calculated in different ways and we have propose a possible calculation of them.

The value range of the Davies-Bouldin, Dunn and Calinski-Harabasz indexes (Equations 2.13, 2.15 and 2.18) is $[0, +\infty]$ and the Silhouette index (Equation 2.17) has a value range of $[-1, 1]$. Silhouette, Dunn and Calinski-Harabasz indexes have to be maximized, and Davies-Bouldin index has to be minimized. The main difference among these four indexes lies in the calculation of the quality and the shape of each cluster, so they evaluate each clustering solution from different points of view (Halkidi et al., 2002b). Davies-Bouldin index evaluates the clusters taking into account if they are scattered by calculating the distance among the instances of each cluster and their respective centroid. Dunn index evaluates the clusters calculating if they are compact by penalizing the clusters with a long diameter, but they are sensitive to the presence of noisy vectors. Silhouette index calculates the tightness of the clusters by taking into account the distance between the instances of each cluster. Calinski-Harabasz index evaluates the clusters according to their within-cluster variance, i.e. the variance between all the elements of the same cluster.

Other validation indexes are proposed in the literature (Theodoridis and Koutroumbas, 2008) and they can be useful depending on the features of the data set and the characteristics of the clustering algorithm. Moreover, the indexes proposed in Section 2.5.2 can be used to obtain a clustering solution, from a collection of them, that is more similar to a prespecified structure of the data set.

2.6 Summary

Clustering algorithms are useful to group data according to a set of criteria, obtaining a set of groups where each one contains similar elements. There are several clustering algorithms that can be classified in several ways according to their characteristics. One of these ways is the criteria used to build the clusters, which can be categorized in two kinds of clustering: (1) the first one is considered as conventional clustering and it is based on optimizing an objective function for assessing the quality of groups of element, and (2) the second one is focused on using a set of objectives to promote the definition of clusters, being ensemble and multiobjective clustering two of the main approaches. Final ensemble clustering solution results from the combination of the solutions obtained with many clustering algorithms following different single objectives, but the combination of the results is a complex and deceptive task. On the other hand, the final multiobjective clustering solution is a Pareto set of non-dominated clustering results evaluated for each objective simultaneously, where each one has a different trade-off among objectives.

Multiobjective clustering algorithms are useful when several objectives are needed to group the data. However, when there are more than three objectives, these kind of techniques may not properly obtain a trade-off between them due to the concept of non-dominance is not restrictive enough. The solution to this is to reinforce the restriction of the non-dominance concept allowing to discard solutions even when these are not worse.

When a clustering solution is obtained it is indispensable to know if it is accurate, representative and coherent in order to evaluate the quality of its clusters. This is the aim of the clustering validation methods, which offer a quantitative evaluation of the results. These methods can validate a clustering solution using a previous categorization of the data or using internal information of the data. Also, these methods can be used in order to choose the most suitable solution from a collection of clustering results through the use of validation indexes. Finally, it must be emphasized that these methods are only a tool at the disposal of the expert in order to evaluate the resulting clustering but they cannot replace the point of view of an expert very experienced in a specific domain.

Several papers have been published related to the application of single-objective clustering and clustering validation methods to the educational domain of the *Guidelines for Competence Assessment in Engineering and Architecture* project (Garcia-Piquer et al., 2010a; Vernet et al., 2010; Garcia-Piquer et al., 2009a; Garcia-Piquer et al., 2009b; Golobardes and Madrazo, 2009a; Golobardes and Madrazo, 2009b).

Chapter 3

Evolutionary Algorithms

Evolutionary Algorithms are a search and optimization paradigm that simulates the way nature acts with living entities. This process is roughly based on selection, reproduction and mutation. This paradigm makes possible the exploration of the regions of the search space, which is a huge area with a big amount of potential solutions, where the best solutions are placed. This kind of algorithms begin with a set of initial solutions that are improved through an iterative cycle based on evaluating, selecting, recombining and mutating them. The key aspect for finding high quality solutions lies in individual representation and genetic operators. The algorithms based on this kind of techniques that evaluate the solutions according to several objectives are called Multiobjective Evolutionary Algorithms and they return a Pareto set of solutions with different trade-off among objectives, being necessary to retrieve the most suitable solution from it. The main lack of the techniques based on evolutionary algorithms is that they are expensive in terms of computational time and memory usage, and this is clearly noticeable when they are applied to a huge amount of data.

3.1 Natural Principles

Evolutionary Algorithms (EAs) are a paradigm that includes the learning algorithms which is based on the way nature solves the problem of living entities (Cordón et al., 2001; Freitas, 2002) by means of natural selection and evolution. The first one was introduced by Charles Darwin in 1859 (Darwin, 1859) and it tries to explain how living beings evolve from small changes and from the selection of the fittest individuals. The second one was proposed by Gregor Mendel in 1865 (Mendel, 1865) and it explains how the offspring are the results of the combination of the features of the parents creating an individual that should be hypothetically better. The genetic encoding of a living being (or individual) is contained within their chromosomes. Each chromosome is a large number of genes that are composed of several alleles, which are discrete values. The genes of an individual are usually referred to as a genotype, and the physical expression of the genotype is called phenotype. In the EAs, the individuals follow the aforementioned structure

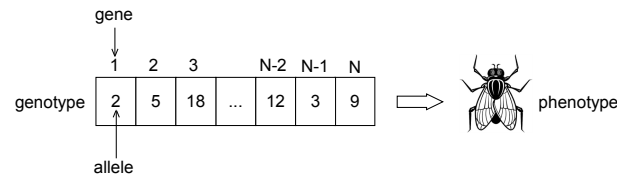


Figure 3.1: Example of an individual representation in a EA. The genotype of N genes represents the genetic information of a fly.

with the expansion that the alleles can be also continuous or nominal values and the genotype is usually represented using a single vector as Figure 3.1 shows. These algorithms reproduce natural selection principle using a fitness function to evaluate the best individuals of a population, being the fittest ones the individuals with more probabilities to survive. Evolution is reproduced crossing the genetic material of the parents to generate new individuals. These new individuals can undergo some random changes, by means of the mutation of some gene.

The natural selection and evolution have found species (solutions) with high degree of adaptation that solve the survival problem in the environment. Conceptually, it can be considered that the natural evolution paradigm is powerful taking into account some actual species. The natural evolution concepts are introduced in the EAs to do a guided search of the solution in huge solution spaces where an exhaustive or random search cannot be sufficiently accurate. This search can be considered directed because the population is guided towards the desired solution through the fitness landscape.

3.2 Taxonomy

There are different EAs approaches and the main differences among them are the individual representation, the definition and the usage of genetic operators, and how the goal has to be achieved. The most representative families are (Freitas, 2002):

- **Genetic Algorithms (GAs).** They were proposed in 1975 by John Holland (Holland, 1992) and his team of the Michigan University. The goal of this approach is to find candidate solutions to an optimization problem. They are theoretically and empirically proven to provide a robust search in complex spaces, thereby offering a valid approach to problems requiring efficient and effective searches (Cordón et al., 2001; Goldberg, 1989).
- **Evolution Strategies (ESs).** They were proposed in the 1960s and developed further in the 1970s and later by Rechenberg and Schwefel (Rechenberg, 1973; Schwefel, 1977) independently from the work of Holland on GAs at the same time. The major difference between ESs and GAs exists in the genetic representation of candidate solutions. The individuals in an ES consists of a vector of real numbers, whereas GAs usually process a population of binary strings (Cordón et al., 2001).
- **Genetic Programming (GP).** It was proposed by Koza in 1992 (Koza, 1992) and they are a

refinement of the GA. It has the the same objective than GA but it is focused on optimizing the parameters of computer programs, which are represented by each individual.

- **Evolutionary Programming (EP).** It was introduced by Fogel in 1960 (Fogel et al., 1966). The goal is focused on the optimization of the combinatorial functions of real values where the optimization surface is abrupt and thus it presents local optimal solutions.

Figure 3.2 shows the cycle of an EA, and their main steps are detailed as follows (Bacardit, 2004):

- **Generation of the initial population.** The individuals used in the first generation of the EA are created according to the individual representation chosen. Each individual is a candidate solution to the problem to solve.
- **Evaluation of the fitness function.** Each individual of the population is evaluated according to the defined fitness function. That is, how good is the individual at solving the problem.
- **Selection of the parents.** Some individuals of the population are selected as parents to produce offsprings. There are many selection approaches (Freitas, 2002), some of them choose individuals based on their proportion of fitness value over the whole population, other methods are rank based, and only take into account if an individual is better than another and not how much better it is.

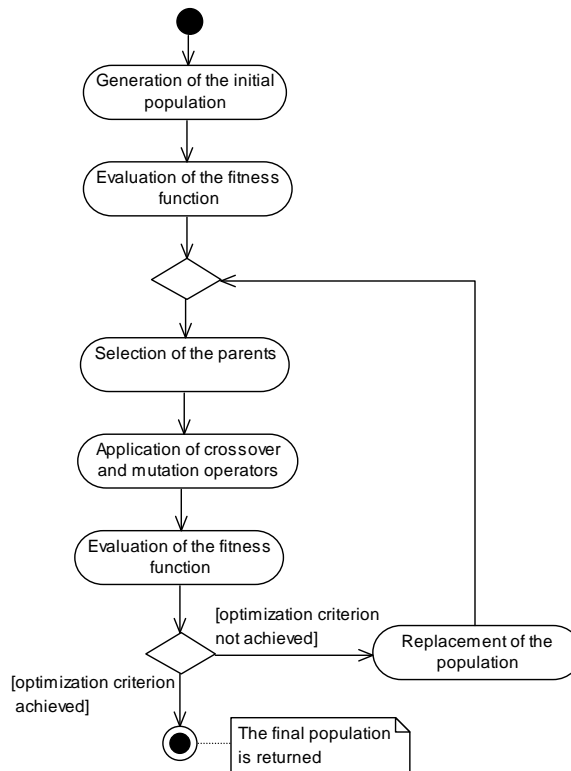


Figure 3.2: Cycle of the steps of an Evolutionary Algorithm.

- **Crossover and mutation.** The individuals selected as parents are combined between them with the crossover operator and the new individuals (offsprings) generated are slightly modified with the mutation operator (Freitas, 2002). Figure 3.3 shows an example of crossover and mutation operators.
- **Replacement of the population.** Given the original population and some new individuals (i.e., the offsprings), the replacement operator is responsible for margin these, obtaining a new set of solutions for subsequent iterations. There are many replacement approaches, being the more extreme ones the next strategies: (1) to erase the previous population and to build a new one with the new generated individuals, and (2) to add the new generated individuals to the population without erasing the previous ones, thus the offspring and the parents being in the same population.
- **Achievement of the optimization criteria.** It defines when the algorithm ends, for example when the evaluated individuals reached a defined likelihood or when the number of generations done by the EA exceeds the maximum number of generations configured. Then, the clustering algorithm returns the best individual of the population according to the fitness function.

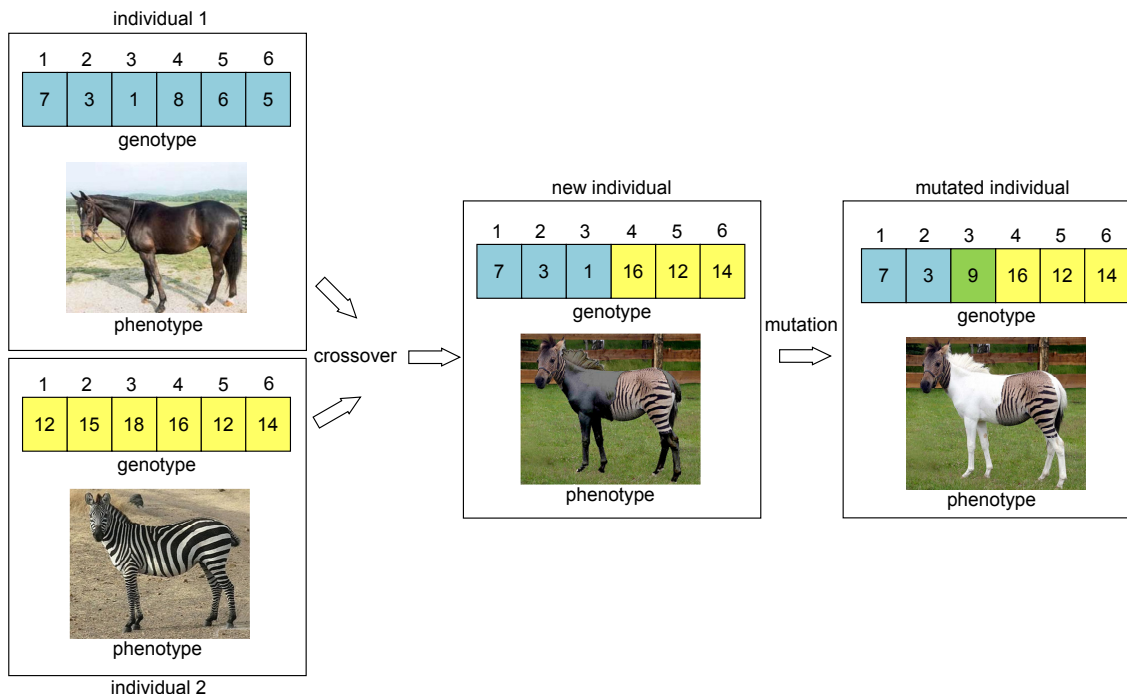


Figure 3.3: Example of crossover and mutation operators used to generate a new individual from two existing individuals. The new individual is generated crossing individual 1 and individual 2 using the crossover operator, which mixes the first half of the genotype of individual 1 with the second half of the genotype of individual 2. Finally, the generated individual is mutated with the mutation operator, which changes the allele of one gene, obtaining the final genetic information of the new individual.

On the other hand, there are two mainly approaches related to the interpretation of the individuals in the EAs (Freitas, 2002): Pittsburgh and Michigan. Simplifying the explanation of the differences between them, on one hand, in the Pittsburgh approach each individual is a candidate solution to the problem, thus the individuals are independent between them. On the other hand, in the Michigan approach each individual is a part of the solution to the problem, so all the individuals make up the complete solution. This approach is useful due to the fact that the individuals can be simpler in comparison with the Pittsburgh approach. Nevertheless, in the Pittsburgh approach the fitness function will measure the performance of an individual without taking into account the other individuals. In the clustering problem, if each individual represents a single cluster, the fitness function only evaluates its quality, but it cannot evaluate the quality of the overall clustering solution. Because of this, Pittsburgh approach is more suitable to solve clustering problems, due to the fact that when the fitness function evaluates an individual it is evaluating the quality of the overall clustering solution.

Next section describes how to adapt the general process of EAs for optimizing several objectives simultaneously.

3.3 Optimizing Several Objectives

The Multiobjective Optimization Problem (MOP) can be defined as the problem of finding (Oszczyka, 1985) a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term “optimize” means finding such a solution which would give the values of all the objective functions acceptable to the decision maker (Coello, 1999). It is rarely the case that there is a single point that simultaneously optimizes all the objective functions of a MOP. Therefore, in these problems it is necessary to look for trade-offs, rather than single solutions. The concept of Pareto Optimality (Pareto, 1896) defines that we can consider a Pareto optimal when there exists no feasible vector of decision variable which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Thus, this concept almost always gives not a single solutions, but rather a set of solutions called the Pareto optimal set. All the solutions included in the Pareto optimal set are non-dominated. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front (Coello, 2001). The reader is referred to (Coello, 1999) for the details.

Multiobjective Evolutionary Algorithms (MOEAs) are EAs focused on optimizing several objectives simultaneously obtaining a trade-off among them. They were proposed by Rosenberg in 1967 (Rosenberg, 1967), but the first algorithm (VEGA) was created by Schaffer in 1984 (Schaffer, 1985). There are two different kinds of MOEA which are described in what follows (Coello, 2003):

- **Non-Pareto based.** These approaches do not incorporate directly the concept of Pareto opti-

num, so they are incapable of producing certain portions of the Pareto front (Coello, 2001). They are efficient and easy to implement, but appropriate to handle only a few objectives. There are several types of non-Pareto based techniques, such as aggregation functions or population based approaches (Coello, 2005). The first ones consist on combine all the objectives into a single one, and they are the oldest mathematical programming methods for multiobjective optimization (Kuhn and Tucker, 1951) but nowadays the evolutionary multiobjective optimization community shows little interest in them. On the other hand, in population based approaches the population of an EA is used to diversify the search by using a different subpopulation for each objective to optimize. The classical example of these approaches is the aforementioned algorithm VEGA, which can deal with a large number of objectives. However, it has several problems such as its selection scheme is opposed to the concept of Pareto dominance (Coello, 2005).

- **Pareto based.** The major step towards this kind of algorithms was given by Goldberg in 1989 (Goldberg, 1989) to solve the problems with VEGA, and they consist of a selection scheme based on the concept of Pareto optimality. The most representative algorithms are NSGA, NPGA and MOGA (Coello et al., 2007; Zitzler et al., 2000), but nowadays they fell into disuse due to some issues which were solved in the following generation of MOEAs. This second generation can be characterized by an emphasis on efficiency and by the use of elitism for retaining all (or some of) the non-dominated solutions found during the evolutionary process, so parents compete with their children and those which are non-dominated (and possibly comply some additional criterion) are selected for the following generation (Coello, 2005). Doing this, all the good individuals can survive until the end of the algorithm and the quality of the population is improved or maintained in each generation in respect of the previous one. The main reason for using elitism in multiobjective optimization is that it has been mathematically proved that it is required in order to guarantee convergence of a MOEA. The most representative algorithms are SPEA, SPEA2, NSGA-II, MOMGA, MOMGA-II, PAES, PESA and PESA-II (Coello et al., 2007; Zitzler et al., 2000).

This thesis is focused on the use of pareto-based MOEAs of second generation in order to obtain a Pareto set of non-dominated solutions with different trade-offs among objectives. The cycle of EAs can be reformulated to obtain the cycle of these kind of algorithms that is shown in Figure 3.4. The new steps or the modified ones are:

- **Evaluation of the objective functions.** Several objective functions, that obtain a single fitness value, are used to evaluate an individual.
- **Selection of the parents.** Some changes in the selection criteria of the parents to produce offsprings are introduced. It is important to highlight that in the population they can be dominated and non-dominated solutions. For this reason, the most common approach is to discard dominated solutions and to choose non-dominated individuals from different regions

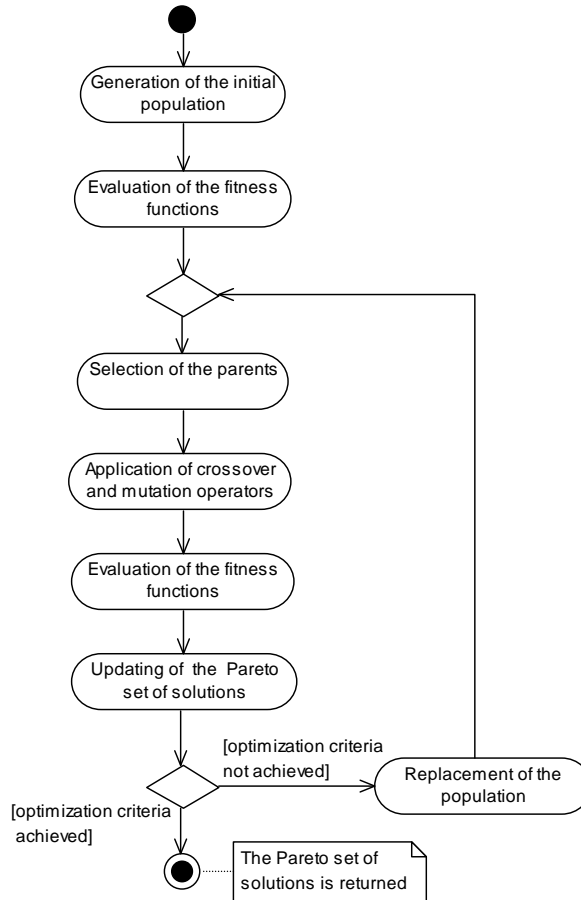


Figure 3.4: Cycle of the steps of a Multiobjective Evolutionary Algorithm.

(niches) of the population, and that is because the individuals of different regions have different trade-offs among objectives. Thus, the selection of the individuals is representative of the overall population.

- **Updating of the Pareto set of solutions.** The Pareto set of solutions is updated adding the non-dominated individuals and erasing the dominated ones. At the end of this step, in the Pareto set of solutions there are only non-dominated individuals.
- **Replacement of the population.** This step is based on the EAs replacement step using some strategy, but in this case the dominated individuals are not introduced in the current population.
- **Achievement of the optimization criteria.** The Pareto set of solutions is returned when the algorithm ends. Usually, in these algorithms is after a certain amount of iterations.

Each one of the above defined phases of an EA and a MOEA have to be adapted according to the family of them used and to the kind of the problem where the algorithm is going to be applied. Chapter 4 explains the clustering algorithm based on MOEAs that is proposed in this thesis, and it

details the aforementioned steps according to it. The next section deals with one of the main lacks of MOEAs, that is, their scalability.

3.4 Scalability

The main lack of EAs is their high computational time and memory usage, being this effect worse when they are applied to large data sets (Freitas, 2002). Two main approaches for speeding up EAs are commonly used. The first one is based on parallelizing some steps of the EA (Cantu-Paz, 2000), and the second one consists in using only a subset of the available data to evaluate the individuals (Bacardit, 2004). There are other strategies less widespread that are based on optimizing the individual evaluation like fitness surrogate, fitness inheritance and exploiting regularities. The first one is based on using a cheaper but less accurate fitness function to evaluate some individuals in order to reduce the computational time of the evaluation (Orriols-Puig et al., 2007). The second one is focused on predicting the fitness of an individual from the fitness of previous evaluated individuals (Llorà et al., 2007). The last one consists in evaluating the individuals taking into account only some relevant attributes (Butz et al., 2008).

3.4.1 Parallelism

One of the main bottlenecks in the EA cycle is the individual evaluation. Due to the fact that EAs can work with a population of independent individuals, it is possible to distribute the computational load among several processors. Therefore, this point is focused on how to parallelize fitness evaluation following the parallel EA taxonomy proposed in (Freitas, 2002) (see Figure 3.5).

There are three kinds of parallel EAs approaches: the control-parallel, the data-parallel and the hybrid control/data-parallel (Cantu-Paz, 2000; Freitas, 2002). The first one distributes the set of individuals of the EA across all the processors and each one evaluates a different subset of individuals. The second one, distributes the data set across all the processors, and each one evaluates each individual of the population with a subset of the data. The last one combines the

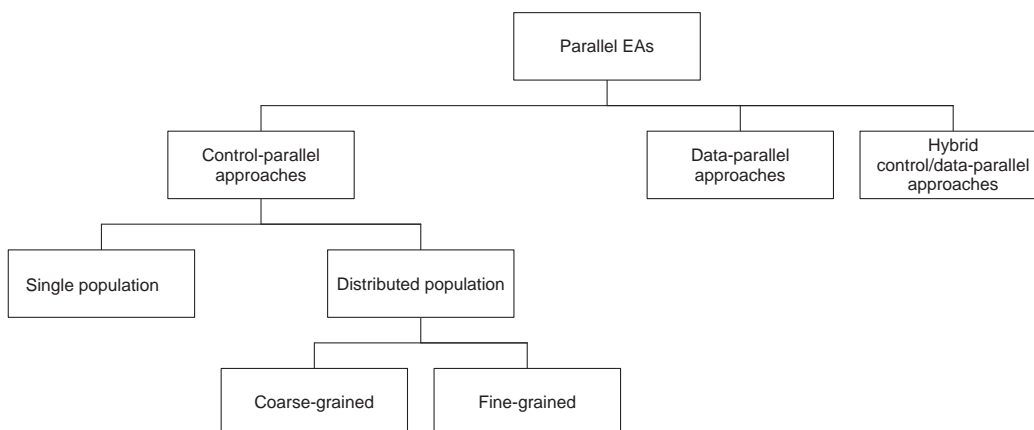


Figure 3.5: Parallel EA taxonomy.

both previous approaches. It is important to highlight that in the three approaches all the processors are working in parallel.

The control-parallel approach can be subdivided into single population EAs and distributed population EAs. The first one applies the selection and genetic operators considering a single population of individuals, and evaluates each subset of the population in different processors. This approach is usually implemented using a master-slave architecture where one master node executes the EA (selection, crossover and mutation) and the evaluation of the individuals is distributed among several slave processors. After the evaluation of each individual, the slaves returns the fitness result to the master that continues with the evolutionary cycle. If the master waits to the evaluation results of all the individuals it is considered a synchronous master-slave EA and if the master waits only for a specific number of individual evaluations it is considered asynchronous. The results of the algorithm are the same than in a conventional EA in the first case and they are not the same in the second case due to the fact that it ignores some individuals in each evaluation process. The control-parallel EAs with distributed populations (also called multiple-population or multiple-deme EAs) consists in distributing several subpopulations (called demes) among multiple processors. Moreover, the coarse-grained algorithms consider a small number of subpopulations, and each one is associated with a single processor. Selection and genetic operators are applied independently at each subpopulation, but the subpopulations can exchange individuals occasionally (migration). The fine-grained algorithms divides the population into a large number of overlapping subpopulations, assigning to each processor a single individual, evaluating all the individuals in parallel, at the same time. Selection and genetic operators are applied in parallel to small neighborhoods around each individual. The neighborhoods overlap, so that the genetic material of good individuals can be spread across the entire spatial structure. Coarse-grained parallel EAs are usually implemented in parallel computers with a relatively small number of processors; however, fine-grained parallel EAs are usually implemented in massively-parallel computers with a large number of processors.

The data-parallel approach distributes the data set across all the processors. The evaluation of each individual is done partially in each processor. Thus, the flow control of the algorithm is sequential due to the fact that an individual cannot be evaluated until all the partial evaluations of the previous individuals were done. In the context of very large data sets, the most important advantage is that it is considerably more scalable than the control-parallel approaches in terms of data. Nevertheless, the number of individuals cannot be scaled-up.

The suitability of each method depends on the cost of the fitness function, the relationship between the reduction of the instances and the reduction of the fitness time. However, it is important to take into account that these approaches require to adapt the algorithms and to introduce communication for coordinating the task executions. The communication cost is related to the quantity of data exchanged between nodes and how the memory is organized (shared-memory or distributed-memory). In shared-memory systems there is a memory that may be simultaneously accessed by multiple processes and the communication costs are related to the access to the global memory

but all the data is available for all the processors. On the other hand, in distributed-memory systems each processor has its own private memory and the communication costs are related to the exchange of data between the processors. Summarizing, these parallel approaches are not useful if the communication costs are higher than the reduction of the evaluation costs regarding to a sequential EA, and this depends on the organization of the memory, the number of processors and the data. The costs of the aforementioned parallel approaches are detailed in Equation 3.1 to Equation 3.4.

$$T_s = g \cdot |P| \cdot T_f \quad (3.1)$$

$$T_{cp-sp} = g \cdot \left[\frac{|P|}{s+1} \cdot T_f + (s \cdot T_c) \right] \quad (3.2)$$

$$T_{cp-dp} = g \cdot \left[|D| \cdot T_f + (r-1) \cdot T_c \right] \quad (3.3)$$

$$T_{dp} = g \cdot |P| \cdot T_{fp} + T_c \cdot s \quad (3.4)$$

$$T_{fp} = \text{time} \left(\text{fitness} \left(\frac{m}{s+1} \right) \right) \quad (3.5)$$

The computational time of a sequential EA (T_s) is shown in Equation 3.1, where g is the number of iterations (generations) of the evolutionary cycle, $|P|$ is the number of individuals of the population and T_f is the time needed to evaluate the fitness of a single individual. The computational time of the control-parallel approach with a single population (T_{cp-sp}) is shown in Equation 3.2, where s is the number of slave processors and T_c is the communication time needed to send the data to a processor. The computational time of the control-parallel approach with a distributed populations (T_{cp-dp}) is shown in Equation 3.3, where $|D|$ is the number of individuals in each deme and r is the number of demes. Finally, the computational time of the data-parallel approach (T_{dp}) is shown in Equation 3.4, where time is a function that calculates the time of a process, fitness is the fitness function that evaluates an individual and m is the number of instances of the entire data set.

3.4.2 Data Subsets Sampling

The individual evaluation is one of the most costly steps in EAs and it is related to the number of instances in the data set. This approach is based on using subsets of data instances from the complete data set to evaluate the individuals instead of using the complete data set. Thus, the

fitness evaluation of the individuals is speeded up but the accuracy could be decreased if subsets are not enough representative. There are several strategies to create data subsets (Bacardit, 2004): the wrapper methods, the prototype selection and the modified learning algorithms.

Wrapper methods run several times a classifier algorithm to correctly select the subset of instances before apply the EA (Olvera et al., 2010; Pudil et al., 1994). The subset of data varies each time the classifier is executed until a stop criterion is achieved, being the last used subset the most suitable one. The stop criterion is based on the properly classified instances using the current subset. After the selection of the suitable data subset, the EA is executed evaluating the individuals according to the selected data subset.

Prototype selection (Aguilar-ruiz et al., 2000; John and Langley, 1996; Salamó and Golobardes, 2002) methods also build the data subset before executing the EA and they do not use a classifier to select the suitable data subset. There are several strategies to select the prototype instances, the most common ones randomly select the data instances with or without replacement. They consists in randomly selecting a predefined number of instances from the entire data set to build the data subset. If there is not replacement, an instance can only be selected once in the data subset. If there is replacement, an instance can be selected more than once in the data subset. Usually, statistical tests are used to determine if the obtained subset is sufficiently similar to the whole training set. In this methods the performance of the EA is strongly related to the behavior of the prototype selection process.

Modified learning algorithms are methods that modify the EA to include the incremental learning inside their algorithm or methods that include or discard instances based on knowledge-representation specific information. The main idea of these methods is to dynamically select a training subset for fitness evaluation. There are several strategies to perform this, and the two most used are explained in what follows. The first one (Fürnkranz, 1998; Gathercole and Ross, 1994; John and Langley, 1996) assumes that EA is used for classification tasks. The idea is to run the EA using a different subset of data in each iteration according to the properly classified instances by the individuals of the current iteration. The second one consists in stratifying the complete data set and alternate between the strata in each generation or individual evaluation. Stratification consists in dividing a data set into homogenous subsets called stratum. The strata has to be disjoint, thus every instance must be assigned to only one stratum. Also, the strata has to be collectively exhaustive, thus no instance can be excluded. The strata can be build randomly or using any strategy to improve the representativeness of each stratum according to the whole data set. Usually, when the data set has a class assigned to each instance, the strata are build with equal class distribution of the instances than in the original data set.

In (Freitas, 2002) these methods are classified in three types according to the frequency of re-sampling. Run-wise is based on selecting a static subset of instances for the whole evolutionary process. Generation-wise is focused on changing the data subset at each generation of the evolutionary process. Individual-wise consists in changing the subset of the data used for each fitness computation. The modified learning algorithms can be considered generation-wise or individual-

wise due to the fact that they change the data subset during the evolutionary process. The prototype selection and wrapper methods are considered run-wise.

The most widely used methods are based on prototype selection and modified learning algorithms. Finally, it is important to highlight that wrapper methods are not useful in unsupervised problems due to the fact that they need to classify the original data set to obtain the data subsets. Nevertheless, the selection of the suitable technique to speed up an EA depends largely on the representation of the individuals and the kind of problem to solve.

3.5 Summary

EAs are a paradigm that includes the learning algorithms which simulates the survival and adaptation to the environment by means of natural selection and evolution. These algorithms reproduce natural selection principle using a fitness function to evaluate the best individuals of a population, being the best ones the individuals with more probabilities to survive. Evolution is reproduced crossing the genetic material of the parents to generate new individuals. These new individuals can undergo some random changes, by means of the mutation of some gene. It is worth noticing that the key aspect for finding high quality solutions lies in individual representation and genetic operators.

There are different evolutionary algorithm approaches such as Genetic Algorithms, Evolution Strategies, Genetic Programming and Evolutionary Programming. The main differences between them are the individual representation, the definition and the usage of genetic operators, and how the goal has to be achieved. Moreover, there are two approaches related to the interpretation of the individuals in the evolutionary algorithms: Pittsburgh and Michigan. In the first approach, each individual is a candidate solution to the problem; and in the second one, each individual is a part of the solution to the problem. The adaptation of the evolutionary algorithms for optimizing several objectives are the MOEAs. They are focused on optimizing several objectives simultaneously by obtaining a Pareto set of solutions with different trade-off among objectives. To obtain a final solution it is necessary to retrieve the most suitable solution from the Pareto set according to the problem to be solved.

The main lack of the approaches based on evolutionary algorithms is their high computational time and memory usage, being this effect worse when they are applied to large data sets. Usually, the evaluation of the individuals is the most expensive step in terms of computational time. Thus, the two main approaches for speeding up these kind of algorithms are: (1) to parallelize the evaluation of each individual, and (2) to use only a subset of the available data to evaluate them. Finally, it must be emphasized that the memory usage of these algorithms is related to the amount of data to manage but it mainly depends on the individual representation used.

Part II

CAOS: Designing a New Multiobjective Clustering Algorithm

Chapter 4

Foundations

This thesis proposes a new MC algorithm to group data according to several objectives. The proposed algorithm is based on MOEAs and it offers the capabilities and the competitiveness with respect to the state-of-the-art of the clustering methods. The algorithm has been designed sufficiently flexible to be subsequently improved to face up the three challenges identified from the literature related to this kind of algorithms, which are the following: (1) definition and exploration of the search space, (2) scalability with large data sets, and (3) selection of the final clustering result.

4.1 Motivation

The objective of this thesis is to propose a new Clustering Algorithm based on multiObjective Strategies (CAOS) for tackling three of the most relevant challenges in MC. CAOS is based on the PESA-II algorithm (Corne et al., 2001) which is a MOEA that reduces the computational cost associated with Pareto ranking (Coello et al., 2007). Concretely, it adapts PESA-II for returning a collection of clustering solutions with optimal trade-offs among objectives evaluated simultaneously, where the best solution is selected using one of the different ways integrated in it such as the Adjusted Rand index (Yeung and Ruzzo, 2001), Davies-Bouldin index (Davies and Bouldin, 1979), the Dunn index (Dunn, 1974), the Silhouette index (Rousseeuw, 1987), or the Calinski-Harabasz index (Caliński and Harabasz, 1974). Moreover, it automatically selects the optimal number of clusters and it allows an easy way to introduce new objective functions and set the individual representations with their associated genetic operators according to the problem. Therefore, the user is able to configure the system using the most specific objective functions, the best selection method of solution and the most suitable individual representation according to the domain requirements.

The next sections present several previous work in MC, and describe in detail the CAOS process, the basic objectives to optimize, the method used to retrieve the best solution and the additional features added.

4.2 Related Work

The Multiobjective Clustering concept was firstly introduced in the year 1992 (Ferligoj and Batagelj, 1992). In 2004, Handl and Knowles proposed a multiobjective clustering algorithm called VIENNA (Handl and Knowles, 2004a), improving it later to obtain the MOCK algorithm (Handl and Knowles, 2004b; Handl and Knowles, 2007) that is one of the most well-known MC algorithms. The multiobjective optimization technique most used is based on evolutionary algorithms since they (1) employ a population based-search evolving a set of optimal trade-offs among objectives, (2) can be easily adapted to the domain typology due to their flexible individual representation, and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions.

The class of evolutionary algorithms that are based on the optimization of several objectives functions that guide the search are called Multiobjective Evolutionary Algorithms (MOEAs) (Coello et al., 2007; Schaffer, 1985). MOCK (Handl and Knowles, 2007) performs partitional hard clustering and it is based on the MOEA algorithm called PESA-II (Corne et al., 2001). In contrast, there are other approaches based on the MOEA algorithm called NSGA-II (Deb et al., 2002) focused on obtaining partitional fuzzy clusters (Bandyopadhyay et al., 2007; Mukhopadhyay et al., 2007). Moreover Faceli et al. (Faceli et al., 2009) proposed MOCLE, a clustering ensemble algorithm which combines several partitions of the data obtained from some conventional clustering algorithms using NSGA-II. In the latter case, the MOEA algorithm is only used for combining some previously found clustering solutions optimizing several objectives.

From the aforementioned algorithms, the most suitable algorithm according our starting point is MOCK due to the fact that it uses the MOEA to obtain the potential clusters and it uses a hard clustering scheme. Nevertheless, it was not conceived for tackling the three challenges suggested in this thesis. This is due to the fact that its internal structure does not allow it to properly explore the search space of several kinds of problems. Moreover, it does not face up scalability and it retrieves the most suitable solution from the Pareto set using only criteria based on the objectives to be optimized or on the quality of the clusters, obtaining solutions that do not have a balance between these two. For this reason CAOS is devised in order to obtain a MC algorithm that overcomes these gaps.

4.3 Process and Design

CAOS inherits its process from PESA-II technique as Algorithm 4.1 describes. We chose this kind of algorithm due to it has similar accuracy performance (Konak et al., 2006) than NSGA-II (Deb et al., 2002) or SPEA2 (Zitzler et al., 2001), two well-tested MOEAs, but it has better computational performance because it is not based on ranking the individuals and does not have the costs associated to this action (Coello, 2005). Moreover, it is very simple to implement (Corne et al., 2001). Concretely, PESA-II works evolving a population of individuals where each one represents a complete clustering solution. This population is recognized as the *external population*

(EP) and it has a fixed maximum size of N_{EP} individuals. In addition to the external population, the system maintains an *internal population* (IP) of fixed size N_{IP} . The purpose of maintaining two detach populations is to separate the exploration from the storage of the best solutions. IP is used to explore new promising solutions by means of applying a typical genetic cycle, i.e., selection, crossover and mutation. On the other hand, EP is employed to store a large and diverse set of non-dominated solutions found so far. A solution S is dominated by S' when S' wins S in all the objectives. Otherwise, the solution is non-dominated. In addition, EP is organized in N_{niches} different niches. That is, a hyper-grid is placed in the objective space, splitting this space in hyper-rectangles where each one is considered a separate niche. Therefore, solutions with similar evaluation of the objectives will be placed in the same niche. This is used by the replacement process, which will make pressure toward balancing the allocation of solutions in different niches thus encouraging solutions to cover all the objective space.

```

1 Let EP be a external population which can store a maximum of  $N_{EP}$  individuals
2 Let IP be an internal population which stores  $N_{IP}$  individuals where  $N_{IP} < N_{EP}$ 
3 Initialize IP with  $N_{IP}$  individuals stochastically created
4 Initialize the EP individuals with non-dominated clustering results from IP
5 Evaluate all the individuals from EP according to the objectives
6 foreach Generation do
7   Select  $N_{IP}$  individuals from EP to generate a new IP
8   while ( $|IP| \neq \emptyset$ ) do
9     Select and remove two individuals from IP
10    Cross and mutate them to obtain 2 new individuals:  $I_{New_1}$  and  $I_{New_2}$ 
11    foreach  $I_{New_i}$  do
12      Evaluate the  $I_{New_i}$  fitness according to the objectives
13      if  $I_{New_i}$  dominates any individual from EP then
14        Remove the dominated individuals by  $I_{New_i}$  from EP
15        Add  $I_{New_i}$  into EP;
16      else if  $I_{New_i}$  is not-dominated and  $I_{New_i}$  not-dominates any individual then
17        if EP is full then
18          Remove an individual from the most crowded niche
19        Add  $I_{New_i}$  into EP
20 Select a individual from EP as a solution

```

Algorithm 4.1: Life cycle of CAOS.

At the beginning, IP is stochastically initialized with N_{IP} individuals and all the non-dominated solutions are copied to EP. Populations are evolved iteratively evolved applying a genetic algorithm for a certain number of generations through the steps of evaluation, selection, crossover and mutation. First, a new population IP_{t+1} is created by selecting N_{IP} individuals from EP. More specifically, the EP objective space is divided into N_{niches} hypercubes of equal size, creating an uniform hyper-grid where each individual is mapped to one of these hypercubes according to its objective values. For selecting one individual it chooses a non-empty niche from EP and randomly selects one of the individuals mapped to the chosen niche. This process is repeated N_{IP} times until filling IP_{t+1} . Then, pairs of individuals are selected from IP_{t+1} without replacement, and they undergo crossover with probability P_c , generating two new offspring. If crossover is not applied, the

offspring are exact copies of the parents. Then, each offspring undergoes mutation with probability P_μ , and the resulting individuals are evaluated. Next, the population EP_t is copied into EP_{t+1} and each offspring of_i is introduced into EP_{t+1} if one of the following two conditions is satisfied:

1. of_i dominates one or several solution/s in EP_{t+1} . In this case, the dominated solutions are removed from EP_{t+1} .
2. of_i does not dominate any solution in EP_{t+1} and none of the solutions in EP_{t+1} dominates of_i . In this case, of_i is inserted into EP_{t+1} if there is room for it. Otherwise, of_i replaces a solution from the most crowded niche in EP_{t+1} , if this niche is more crowded than the niche where of_i is placed.

The working cycle of CAOS does not depend on the individual representation, the genetic operators and the objectives used to evaluate the individuals. But the suitable selection of them can allow to obtain better results depending on the domain of the problem. These topics are analyzed in the next chapters of this thesis.

Finally, some additional features are added to the main process of CAOS in order to obtain a better performance of the algorithm. These additions are introduced in the section 4.6.

4.4 Objective Functions

One of the main keys to obtain good clustering results is the definition of the criteria to optimize in each group of data. One of the strongest points of MC is the possibility of optimize several objectives at the same time trying to obtain compact and separated clusters, this means clusters with elements similar among them and different to the elements of other clusters. This degree of similarity between clusters can be calculated according to general purpose objectives or to specific objectives. The first ones use criteria that are not based on the domain of the problem and the other ones are specifically based on the domain of the problem, improving considerably the results due to the fact that they can adapt better to the problem. Nevertheless, the definition of the specific objectives is not trivial, because it is necessary to perfectly understand the domain of the problem and normally it is essential the help of an expert of the domain. This is an important drawback because in several situations the experts cannot define specific objectives due to the complexity of the domain. For this, the general purpose objectives are frequently used in the majority of the problems, being *Deviation* and *Connectivity* (Hruschka et al., 2009) two of the most objectives functions used in clustering. These objectives measure the compactness and separation of the clusters and they have been successfully used in several problems and domains (Handl and Knowles, 2007).

The *Deviation* (Dev) measures the compactness of the clusters. It is computed as the overall summed distances between data items and their corresponding cluster center as Equation 4.1 shows, where C is the clustering obtained, C_i is the set of instances belonging to cluster i , v_i is the centroid of cluster i , and $d(x, v_i)$ is the Euclidean distance between the element x and v_i . The value of the objective has to be minimized because it computes the overall distance between the

elements of each cluster and we want to obtain compact clusters. The *Connectivity* (Conn) refers to the cluster connectedness. It takes into account the degree to which data points that are close in the feature space have been placed in the same cluster as Equation 4.2 shows, where m is the number of examples in the training data set, C is the clustering obtained, $nn(x, i)$ returns the i th nearest element of x using the Euclidean distance and ℓ is the amount of nearest elements taken into account. Note that, for each instance i , the metric computes a weighted sum of the ℓ nearest neighbors that belong to a different cluster from that of i (the weight is decreased according to how far instances i and j are). It computes the overall number of elements that should be considered in the same cluster due to the proximity to other elements, so the value of the objective has to be minimized because we want well-separated clusters.

$$Dev(C) = \sum_{i \in C} \sum_{x \in C_i} d(x, v_i) \quad (4.1)$$

$$Conn(C) = \sum_{x=1}^m \left(\sum_{i=1}^{\ell} \chi(x, nn(x, i), i) \right), \text{ where} \quad (4.2)$$

$$\chi(x, y, i) = \begin{cases} \frac{1}{i} & \text{if } \neg \exists j : x \in C_j \wedge y \in C_j, \\ 0 & \text{otherwise.} \end{cases}$$

4.5 Selection of the Best Solution

The final step is to select a solution from the Pareto set (composed by all non-dominated solutions) when the evolutionary process ends. This point is not trivial because there is no single individual which is the best in all the objectives and for this reason clustering validation techniques are required. Therefore, the goal of this selection is to obtain the best solution of the Pareto set of solutions taking into account how good are solutions between them. Thus, CAOS integrates many relative criteria methods with the most known validation indexes (Adjusted Rand index, Davies-Bouldin index, Dunn index and Silhouette index).

It is important to highlight that the Adjusted Rand index is based on obtain the best solution according to a prespecified structure of the data set, in our case, the classes assigned to each instance. This index is used to test is CAOS can obtain a solution in the Pareto set very similar to the structure defined by the classes, and it can only be used if the classes of the instances are known. On the other hand, the other four indexes are based on inherent information of the data set in order to obtain solution with cluster of good quality, that is, if each cluster have elements that are similar among them and different to the elements of the other clusters. Each one of these indexes makes different calculations (see Section 2.5.4) and they can return a different clustering solution from the Pareto set, so the final decision of the best solution depend on the point of view of the experts using indexes as a way to help them in this selection. It must be emphasized that Section 7 proposes a

method to improve this process in order to retrieve better solutions from the Pareto set.

4.6 Additional Features

Situations derived from the features of some individual representations could penalize the algorithm performance. For example, a combination of an individual representation and genetic operators that allow to explore a huge search space could obtain good results but may have an elevated run-time due to the number of individuals managed or it may be find solutions with too many clusters. For this reason, two improvements have been included in CAOS to increase the performance. The first one is oriented to control the growth of the individual population and the second one proposes to merge some of the clusters of the final clustering solution. The usage of these enhancements improve the results depending on the individual representation used.

4.6.1 Bloat Control

The bloat effect (Langdon, 1997) is related to the unlimited growth of the size of the individuals and it is useful to prevent the effect and to reduce the amount of individuals in each generation that increase the run-time. Moreover, it can be helpful to introduce generalization pressure into the system, evolving more accurate but compact solutions that will be potentially better test accuracy (Bacardit, 2004). The bloat control is introduced in the following steps:

1. The EP objective space is divided into N_{niches} hypercubes of equal size where each individual is mapped to one of these hypercubes according to its objective values. One of the objectives used is the cluster connectedness, and the growth of the value of this objective is related to the increase of the number of clusters. When in each generation N_{IP} individuals have to be selected from EP to generate a new IP , an individual is randomly selected from a chosen niche. The niches with solutions composed of less clusters (low cluster connectedness) are chosen with more probability.
2. When a new individual is generated in each generation it has to be added to EP if it is not dominated, erasing all the individuals of EP that are dominated by this new individual. Generally, S is dominated by S' when S' wins S in all the objectives. In our case, to prevent the bloat into EP we consider a solution S dominated by S' in the aforementioned case and when S' is similar in all the objectives and has less clusters than S . Two solutions S and S' are similar when the difference of each one of the objectives of S and the correspondent value of the objectives of S' is into a fixed threshold (Bacardit, 2004).

4.6.2 Cluster Merging

This improvement is a post-processing stage applied to the members of the final Pareto set. Solutions are refined with the aim of promoting the separation of clusters without penalizing the compactness too much. The process is based on merging clusters in order to decrease the connectivity

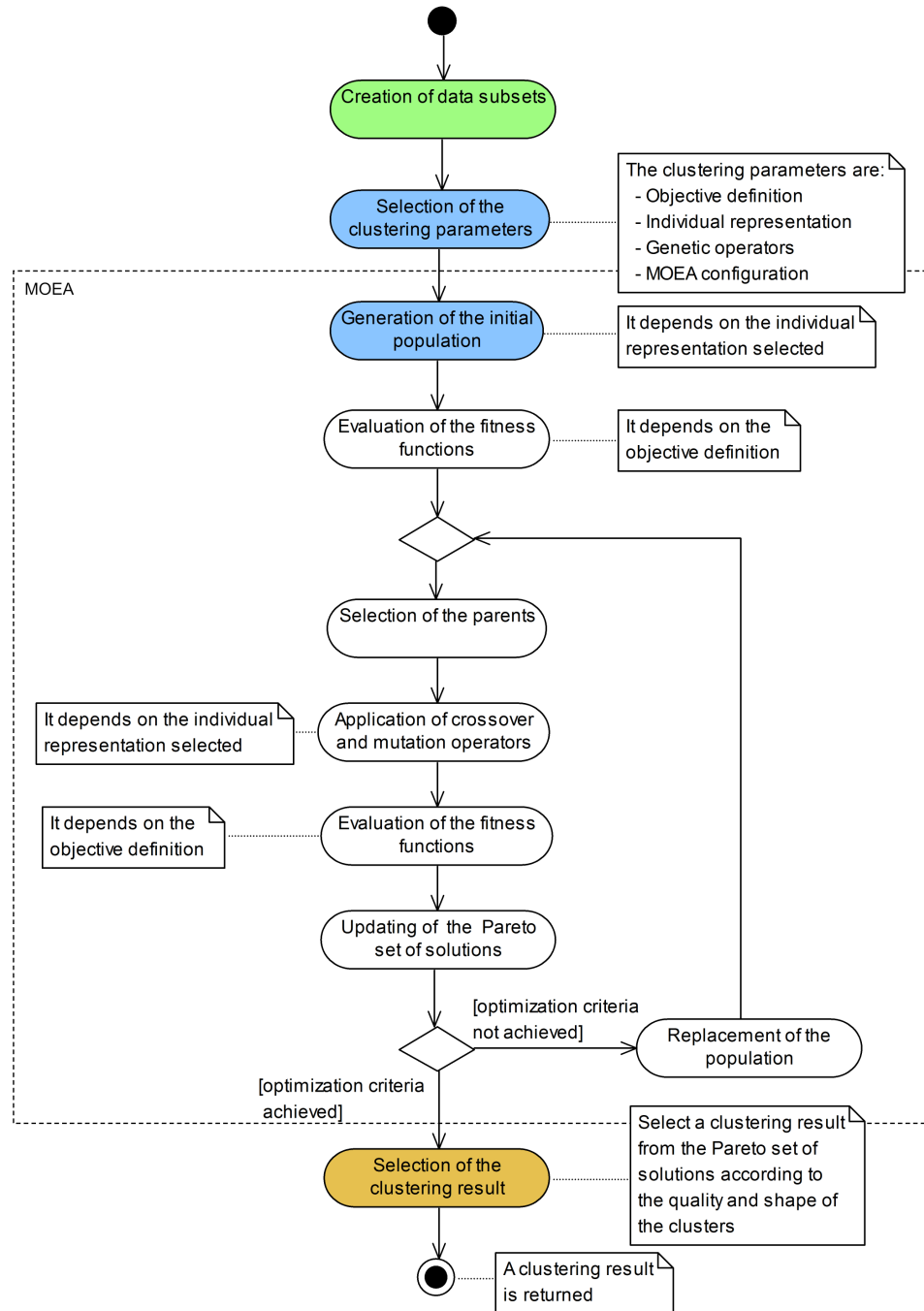


Figure 4.1: Overview of the CAOS process with the steps that will be modified to face up the challenges. The steps related to the challenge that deals with the definition and exploration of the search space are colored in green, the step affected by facing up the scalability with large data sets is colored in blue, and the step related to the challenge that faces up the selection of the final clustering result is colored in orange.

objective without increasing the deviation objective more than a max_{merge} value. Cluster merging is useful when solutions have many clusters with low density, that is, the number of clusters is very high. The procedure is as follows:

1. Two clusters are randomly selected from the clustering solution and they are merged obtaining a new clustering solution.
2. If the new clustering solution does not promote the separation of clusters without penalizing the compactness too much, this last merge is undone.
3. Step 1 and Step 2 are repeated until the resulting clustering solution has only two clusters or there are not more possible combinations of pairs of clusters.

The next chapters describe the research done to face up the proposed challenges related to the definition and exploration of the search space through the most suitable individual representation, the scalability with large data sets, and the selection of the final clustering results. Figure 4.1 depicts a global vision of the CAOS process in order to show the steps that will be affected by dealing with the challenges.

4.7 Summary and Conclusions

CAOS is a new Clustering Algorithm based on multiObjective Strategies designed for tackling three of the most relevant challenges in MC based on EAs. It adapts the MOEA PESA-II that returns a collection of solutions with optimal trade-offs among objectives evaluated simultaneously.

It can be observed that the CAOS process follows the general PESA-II algorithm without adapt it to the clustering problem. The individual representation and the genetic operators are the responsible for adapting it. However, it is necessary to select them accurately in order to properly guide the evolutionary search as it is exposed in the next chapter. This is one of the challenges faced up in this thesis and it is intensely analyzed in Section 5. The process ends returning a collection of non-dominated solutions where it is necessary to select the most suitable solution. This is not a trivial step because the retrieval of the final solution can depend on several features related to the domain and the quality of the clusters. Thus, in the contributions proposed, the final solution is retrieved according to the best value obtained with the clustering validation indexes. However, due to the fact that this is one of the challenges proposed in this thesis, it is deeply tackled in Chapter 7. Moreover, additional features to enhance the performance of the algorithm are implemented such as bloat control and cluster merging. The first one is used to discard individuals that can distort the evolutionary cycle results. On the other hand, the second additional feature is based on merge the clusters of the individuals with small ones in order to obtain more separated clusters.

The CAOS process is based on evolutionary algorithms and the main lack of these techniques is that they have a high computational time and they may have a high memory usage. Thus, it is necessary to adapt the process in order to solve problems with large data sets in a reasonable time and memory usage. This is another challenge faced up by this thesis and it is deeply analyzed in chapter 6.

We can consider as further work all the challenges presented, that are related to CAOS. Moreover, it can be interesting the analysis of other multiobjective evolutionary algorithms or other

multiobjective techniques, instead of the PESA-II algorithm, in order to observe the different capacities of these techniques to explore the search space, such *a priori* or interactive methods instead of the *a posteriori* method used (Branke et al., 2008; Veldhuizen and Lamont, 2000). Finally, it will be interesting the analysis of other general-purpose objectives related to the clustering philosophy in order to improve the optimization process of CAOS with the aim of obtaining better built clusters.

Chapter 5

Definition and Exploration of the Search Space

Multiobjective evolutionary clustering algorithms are based on the optimization of several objective functions that guide the search following a cycle based on evolutionary algorithms. Their capabilities allow them to find better solutions than with conventional clustering algorithms if the suitable individual representation is selected. This chapter analyzes in detail the performance of the three most relevant and useful representations – prototype-based, label-based, and graph-based – through a wide set of synthetic data sets and data sets from the UCI Repository. Moreover, their performance are also compared with respect to relevant conventional clustering algorithms for understanding the situations where a representation is more suitable than the other ones.

5.1 Motivation

To successfully apply evolutionary multiobjective algorithms to real-world problems is important to choose a suitable individual representation according to the domain of the problem, due to the fact that the representation defines the search space. This has motivated many works focused on the analysis and design of several representations that have demonstrated their competitiveness. The three most relevant and useful representations are: (1) prototype-based (Bandyopadhyay et al., 2007; Mukhopadhyay et al., 2007; Ripon et al., 2006) uses genes as clustering prototypes, (2) label-based (Cowgill et al., 1999; Hruschka et al., 2004; Ma et al., 2006) associates genes between them using labels, and (3) graph-based (Handl and Knowles, 2007) links genes between them to represent clusters. The first two are the most used in evolutionary clustering algorithms (Hruschka et al., 2009; Krishna and Narasimha, 1999; Maulik and Bandyopadhyay, 2000), and the last one is used by the most relevant multiobjective evolutionary clustering algorithm (Handl and Knowles, 2007). Nevertheless, a complete discussion and empirical study about the suitability of these representations has not been reported.

The goal of this contribution is to compare the performance of the last three individual rep-

representations and their genetic operators in terms of accuracy, search space and time cost through CAOS. Moreover, the analysis is extended by comparing the results with respect to the most used single-objective clustering algorithms: k -means (Hartigan and Wong, 1979), EM (Dempster et al., 1977), x -means (Pelleg and Moore, 2000) and SOM (Kohonen, 2000). The objective of the comparison is to analyze the performance of these representations with different kinds of data sets to identify the most robust one, to carry out this, the approaches are compared along a wide set of benchmarking synthetic data sets (Handl and Knowles, 2007), where the domains are controlled, and real-world problems from the UCI repository (Asuncion and Newman, 2010). The work is organized as follows. The next sections briefly summarize the related work on individual representations, describe CAOS with the proposed representations and present the experiments done.

5.2 Related Work

Two of the most used individual representations in evolutionary clustering are based on (1) labels (Cowgill et al., 1999; Krishna and Narasimha, 1999) and on (2) prototypes (Bandyopadhyay et al., 2007; Maulik and Bandyopadhyay, 2000). The first representation defines an individual as a list of all the instances of the data set assigning a number of cluster to each one. Thus, this representation can identify cluster structures of arbitrary shape due to each instance can be assigned to any cluster. The second representation defines an individual as a collection of prototypes with the aim of assigning each instance to a cluster. Concretely, it assigns each instance to the nearest cluster prototype and, consequently, this representation can only identify clusters of ellipsoidal shapes. Another representation based on the relationship among the instances in multiobjective clustering is proposed in (Handl and Knowles, 2007), and it is similar to the idea proposed in the evolutionary clustering algorithm proposed in (Tseng and Bien, 2000). In this representation an individual is defined as a graph where all the instances connected among them correspond with a cluster, and it can represent clusters with arbitrary shapes due that there is not restriction between relationships of the instances.

The choices of cluster representation within MC were studied by Handl and Knowles (Handl and Knowles, 2006). They did a theoretical and empirical comparison of some label-based and graph-based representations applied to MC using a framework called MOCK, showing that the graph-based representation performs very well but it greatly depends on the initialization scheme used. Hruschka summarized the main features of the evolutionary clustering algorithms and did a survey of the most used individual representations and their associated genetic operators (Hruschka et al., 2009). However, a complete discussion and empirical study about the suitability of the individual representations in MC has not been reported. The next section describes in detail the individual representations proposed and their application in CAOS.

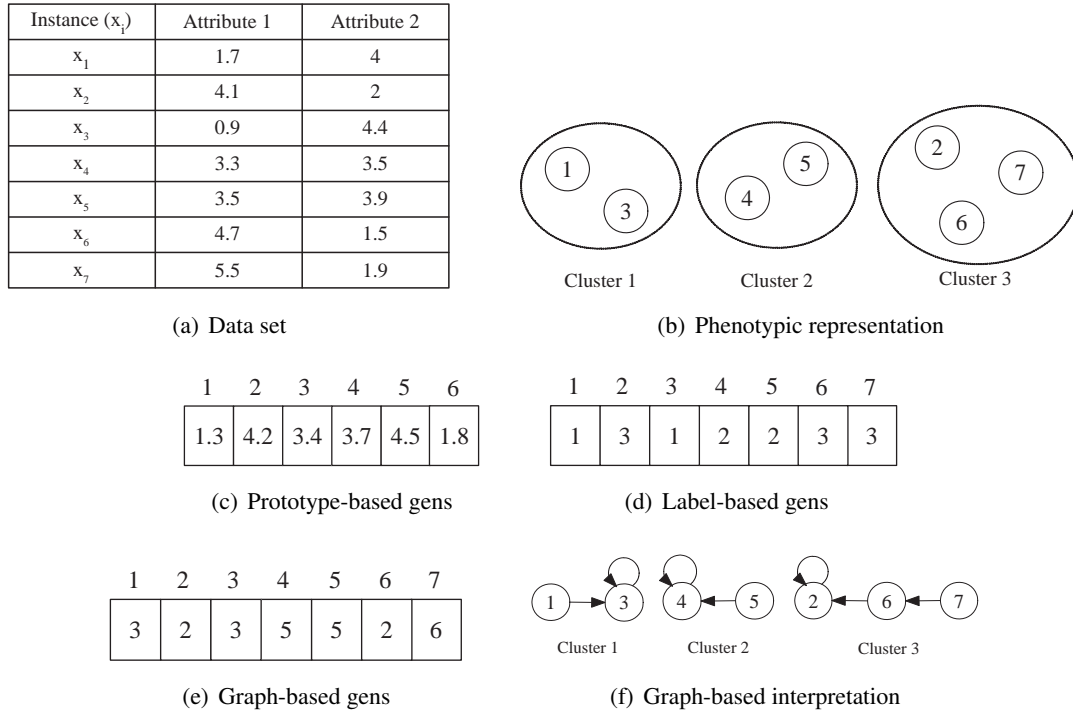


Figure 5.1: Individual representation for a clustering solution using a data set of 7 instances with 2 attributes. The example shows (a) the example data set, (b) the phenotypic interpretation of the clustering, (c,d,e) the individual representation of each approach, and (f) the concept resulting from the graph-based representation.

5.3 Representations

Individual representation and how it is initialized are two important issues in evolutionary algorithms and they selection is directly related to the domain characteristics. The genetic operators have the goal of exploring new areas in the search space, but an uncontrolled application could lost the focus on the right search way in some individual representation. The next points detail for each one of the three approaches the individual representation, the population initialization and the genetic operators used in CAOS.

5.3.1 Prototype-Based Representation

The prototype-based representation is made up of real numbers which represent the coordinates (attributes) of the cluster prototype. The prototypes can be defined by medoids (instances existing on the data set) or by centroids (artificial prototypes). Our implementation uses the concept of centroid to represent each one of the clusters, so this approach can be considered a centroid-based representation (Hruschka et al., 2009). More specifically, each individual consists of $n \cdot t$ genes $\{x_{11}, \dots, x_{1t}, \dots, x_{n1}, \dots, x_{nt}\}$, where n is the number of clusters of the individual, t is the number of the attributes of the data set, and x_{ij} is the value of the attribute j of the cluster centroid i . The genotypic representation is transformed into the phenotypic representation by assigning each instance to the

cluster with the nearest centroid to it. Figure 5.1 shows an example of this genotypic-phenotypic mapping of a clustering solution.

The population initialization step is responsible for filling the population with individuals that contain potentially good clusters. This approach uses a initialization based on medoids to define the initial prototypes, following the same idea than the k -means algorithm (Hartigan and Wong, 1979). The process for each one of the initial individuals is:

1. Randomly select a number k of clusters between a minimum and maximum value.
2. Generate the individual choosing randomly k elements of the data set, where each one represent the prototype of a cluster.

The crossover operator mixes the genetic information of the individuals to obtain new individuals. In this case, a one-point crossover operator is used to generate two offspring from pairs of parents. One point is selected for each parent and the parts are interchanged between them, taking into account that they have to cut the individuals at the same attribute but not necessarily at the same cluster. This is an easy crossover strategy according to the size of each individual can be different.

The mutation operator modifies a piece of the genetic information of an individual to explore new solutions. The probability P_μ determines when the operator is applied. To mutate the individuals, a cluster-oriented mutation operator (Hruschka et al., 2009) is used to promote the right search. The operator defines three different types of mutations and all of them have the same probability to be applied: (1) to merge two clusters, (2) to split a cluster, and (3) to move the centroid of a cluster. Mutation 1 merges a randomly selected cluster $s1$ with its nearest cluster $s2$, adding the new cluster centroid to the individual and erasing both original clusters. The new centroid is calculated with the weighted average between the original cluster centroids and the elements of each one as Equation 5.1 shows, where $|C_i|$ is the size of the cluster i and v_i is the centroid of cluster i . Mutation 2 splits a randomly selected cluster s in two clusters $s1$ and $s2$. $s1$ is equal to s and $s2$ is the most distant element x from s using the Euclidean distance as Equation 5.2 shows. Mutation 3 moves the centroid of a randomly selected cluster s changing the value of a randomly selected attribute sa . If sa is a numerical attribute it is modified by adding or subtracting a δ value to each attribute as Equation 5.3 shows. The range of the v_{sa} is between the minimum and maximum value of the attribute sa in the data set. If the new v_{sa} value is out of the attribute range, the new value is fixed to the corresponding maximum or minimum value. If sa is a categorical attribute, the value is changed by one of its other possible categorical values V_{sa} .

$$v_{s1s2} = \frac{|C_{s1}| \cdot v_{s1} + |C_{s2}| \cdot v_{s2}}{|C_{s1}| + |C_{s2}|} \quad (5.1)$$

$$v_{s2} = \max_{x \in C_s} d(v_s, x) \quad (5.2)$$

$$v_{sa} = \begin{cases} v_{sa} \pm \delta & \text{if } sa \text{ is numerical,} \\ nv, nv \in V_{sa} & \text{otherwise.} \end{cases}$$

After crossover and mutation operators could be inconsistent individuals with empty clusters. These clusters are eliminated from the individual to obtain a new individual where each cluster has at least one instance assigned.

5.3.2 Label-Based Representation

The label-based representation uses an integer encoding scheme assigning a cluster to each instance. Each individual consists of m genes $\{x_1, x_2, \dots, x_m\}$, where m is the number of examples of the training data set and x_i ranges in $[1, max_{clusters}]$. Thence, each gene x_i indicates that instance x_i belongs to cluster i . The value $max_{clusters}$ is the predefined maximum number of clusters of an individual, and cannot be higher than m . The genotypic representation is directly mapped to the phenotypic representation (see Figure 5.1).

The population initialization follows the philosophy proposed at (Handl and Knowles, 2007) using the Minimum Spanning Tree (MST) (Prim, 1957). Thus, the individuals represent the relations between the closest instances. The method consists in:

1. Generate the MST from the undirected, fully connected labeled graph that represents the Euclidean distance between each pair of examples.
2. Generate the first individual of the population removing the link with the highest distance from the original MST.
3. Generate the next individual removing the link with the highest distance from the last individual generated.
4. Repeat last step for generate the P_i individual until there is space in the initial population or until there are not more links between the examples.

It uses uniform crossover to generate two new promising offspring from pairs of parents. For each gene, this operator randomly selects to which offspring the corresponding gene of each parent will be copied. It is worth noting that, as opposed to other selection schemes such one- and two-point crossover, uniform crossover is not biased toward the position of the genes in the genotypic representation, since it is able to shuffle the information of each individual gene.

A neighborhood-biased mutation operator not oriented to clustering is used. If a gene i is selected for mutation (with probability P_μ), then some of the neighbors of the selected instance i are changed to the cluster of it. The numbers of neighbors to change is randomly selected between the range $[1..\ell]$.

5.3.3 Graph-Based Representation

The graph-based representation employs the locus-based adjacency representation proposed in (Park and Song, 1998). That is, each individual encodes a reflexive directed unlabeled graph that connects pairs of examples using an integer encoding scheme. More specifically, each individual consists of m genes $\{x_1, x_2, \dots, x_m\}$, where m is the number of examples of the training data set and x_i ranges in $[1, m]$. Thence, each gene x_i indicates that there exists an arrow connecting instance i with instance x_i . As Figure 5.1 shows, the genotypic representation is transformed into the phenotypic representation by identifying all the connected components of the graph and assigning them to the same cluster. The population initialization follows a MST strategy and uses uniform crossover as it was explained in the label-based representation. A neighborhood-biased mutation operator not oriented to clustering is also used, but in this case it works different. If a gene i is selected for mutation (with probability P_μ), then its value is changed into one of the ℓ nearest neighbors of example i . In addition, each of this ℓ nearest neighbors is given a selection probability proportional the distance between them and instance i .

5.3.4 Search Space and Computational Performance

The capacity of exploring the search space and the behavior of each representation is directly related to the combination of the individual representation and the genetic operators used. The graph-based representation explores a search space of size ℓ^m , due to the fact that each one of the m instances of the data set can only be linked to one of its ℓ neighbors instances. The label-based representation explores a search space of size k^m because each instance can be assigned to one of the k clusters defined; in our case, the maximum number of clusters is m (each instance in one cluster) being this search space of size m^m . The graph-based representation has a huge search space size, due to the fact that the t attributes of each one of the maximum number of clusters defined (m) can have a value between their minimum (max_{att}) and maximum value (min_{att}), that is, $|V|^{m \cdot t}$, where $V = \{x \in \mathbb{R} : x \geq min_{att} \wedge x \leq max_{att}\}$.

The size of the search space is related to the performance of the representations when the individuals are randomly initialized. If the search space of the exploration is little, it could be impossible to obtain good solutions. The graph-based and the label-based solutions lose accuracy when the individuals are randomly initialized, due to the fact that the search space is quite delimited. However, the prototype-based representation explores a larger search space and the quality of the solutions are not affected.

In terms of memory usage, to represent an individual the graph-based and the label-based representations need m integer values, and the prototype-based representation needs $n \cdot t$ real values, where m is the number of instances, t is the number of attributes and n is the number of clusters of the individual. In terms of memory usage, the selection of one of these representation depends on the number of instances and attributes of the data set and the available memory.

Table 5.1 summarize the computational cost, in the worst case, of the three representations. The prototype-based one has long clustering cost than the other two representations. However, the step

Representation	Clustering cost	Merge clusters cost
Prototype-based	$O(g \cdot IP \cdot m \cdot \bar{n} \cdot t)$	-
Label-based	$O(g \cdot IP \cdot m \cdot t)$	$O(n^3 \cdot m \cdot \ell)$
Graph-based	$O(g \cdot IP \cdot m \cdot t)$	$O(n^3 \cdot m \cdot \ell)$

Table 5.1: Cost of each representation broken down in clustering cost and merge clusters cost. Where g is the number of generations, $|IP|$ is the internal population size, m and t are the number of instances and attributes of the data set respectively, \bar{n} is the average of the number of clusters of the individuals, and n is the number of clusters of the retrieved individual.

of merging clusters is done for each one of the retrieved solutions of the graph-based and the label-based representations, and the prototype-based representation does not need this step. In large data sets where a huge number of clusters can be identified, the prototype-based representation could be very costly.

These encoding characteristics have directly implications that have to be taking into account. From a topological point of view of clusters, the label-based and the graph-based representations should allow clusters of complex shapes, on the other hand, the prototype-based representation only allows clusters of ellipsoidal shape. From a computational point of view, the individuals of the prototype-based representation has variable size in contrast with the others two. This makes necessary to apply some bloat control to avoid the degradation of the population. Nevertheless, this representation could be more suitable for large data sets because representation is not directly related to the number of instances in the data set such as in the label-based and the graph-based representations. Finally, cluster merge can be useful for the graph-based representation since it can produce many clusters with low density.

5.4 Experiments, Results and Discussion

The objective of this section is to compare the performance of the three aforementioned individual representations in terms of quantitative and qualitative clustering results and also in terms of computational cost to identify which is the most suitable representation according to the problem typology. Moreover, these results are compared with some of the most used and well-known single clustering algorithms to study the behavior of CAOS.

First, the experimental methodology is described. Next, the comparison between the individual representation is presented. Finally, the comparison is extended with respect to some representative single-objective clustering algorithms. These algorithms are explained in Appendix A.

5.4.1 Experimental Methodology

This section presents the experimental methodology followed to evaluate the CAOS variants and the conventional clustering algorithms. The analysis enables us to emphasize the benefits and the drawbacks of each one. In the followings, we provide details about (i) the data sets collection chosen for the experimentation, (ii) the CAOS configuration, (iii) the conventional single-objective
















Data set	nI	nA	nC	Data set	nI	nA	nC
100d-10c	2198	100	10	balance	625	4	3
100d-4c	1218	100	4	biopsia	1027	24	2
10d-10c	2122	10	10	bpa	345	6	2
10d-4c	1092	10	4	dermatology	366	35	6
2d-10c	2990	2	10	ecoli	336	8	8
2d-4c	1261	2	4	glass	214	9	6
curves1 	1000	2	2	heart-statlog	270	13	2
curves2 	1000	2	2	ionosphere	351	34	2
dartboard1 	1000	2	4	iris	150	4	3
dartboard2 	1000	2	4	liver-disorders	345	6	2
donut1 	1000	2	2	pendigits	7494	17	10
donut2 	1000	2	2	pim	768	8	2
donut3 	999	2	3	segment	2310	19	7
donutcurves 	1000	2	4	segment2c1	2310	19	2
long1	1000	2	2	segment2c2	2310	19	2
long2	1000	2	2	segment2c3	2310	19	2
long3	1000	2	2	segment2c4	2310	19	2
longsquare 	900	2	6	segment2c5	2310	19	2
sizes1	1000	2	4	segment2c6	2310	19	2
sizes2	1000	2	4	segment2c7	2310	19	2
sizes3	1000	2	4	sonar	208	60	2
sizes4	1000	2	4	tae	151	5	3
sizes5	1000	2	4	thyroids	215	5	2
smile1 	1000	2	4	transfusion	748	4	2
smile2 	1000	2	4	vehicle	846	18	4
smile3 	1000	2	4	vehicle2c1	846	18	2
spiral 	1000	2	2	vehicle2c2	846	18	2
spiralsquare 	1500	2	6	vehicle2c3	846	18	2
square1	1000	2	4	vehicle2c4	846	18	2
square2	1000	2	4	wdbc	569	30	2
square3	1000	2	4	wine	178	13	3
square4	1000	2	4	wisconsin	699	9	2
square5	1000	2	4	wpbc	198	33	2
triangle1	1000	2	4	yeast	1484	9	10
triangle2	1000	2	4	zoo	101	16	7

Table 5.2: Summary of the characteristics of the 35 artificial data sets (left block) and real-world data sets (right block) used. The symbol  indicates the handmade data sets. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).

clustering, and (iv) the comparison metrics.

Test Bed. A collection of 70 data sets with different characteristics that may challenge different learning techniques are selected. 35 artificial data sets are extracted from (Handl and Knowles, 2007) and 35 are from the UCI repository (Asuncion and Newman, 2010). From the 35 artificial data sets we can identify 14 handmade data sets that are used to test the representations in domains with arbitrary complex shapes.

CAOS Configuration. Each CAOS representation was run 10 times using different seeds with the following parameters (see Section 4 for notation details): ℓ is 5% of m (the number of data set instances), the maximum size of the initial population is 100, N_{EP} is 1000, N_{IP} is 50, N_{niches} is 5, the number of generations is 400, $P_c = 0.7$ and P_μ is $1/m$. The minimum and maximum number of clusters for the initial individuals at the prototype-based representation is 2 and 20% of m respectively. The bloat control threshold (sim) value is 0.005 and it is applied after the generation

30, and the merge clusters threshold (max_{merge}) value is 0.02. With respect to the selection of the best solution, (1) the Davies-Bouldin index, (2) the Dunn index, (3) the Silhouette index, and (4) the Adjusted Rand index (Yeung and Ruzzo, 2001) are used to test the solutions as Section 4.5 defines. The first three ones are some of the most well-known unsupervised indexes, which take into account the shape of the clusters to evaluate a solution. The Adjusted Rand index is a supervised index that evaluates a clustering solution based on the initial classes of the data set.

Conventional Clustering Configurations. Because the performance of algorithms depend of some parameters, they have been test with several configurations and the best solution for each case has been selected. Thus, K -means and EM are executed with different k in the range [2..15] and SOM is executed with three different maps (3×3 , 4×4 and 5×5) which correspond to different number of clusters. Each algorithm configuration was also executed with 10 different seeds.

Comparison Metrics. The objective of clustering is to group elements according to their attributes' similarity and their evaluation is usually done with clustering validation indexes to measure the quality of the solution. In order to compare among them the best solutions retrieved from the Pareto set using the four aforementioned validation indexes and the solutions provided by the conventional clustering techniques, each solution is quantified using the Adjusted Rand index. This allows us to get a reference measure to compare all clustering strategies using the initial classes of the data set.

Finally, the recommendations pointed out by Demšar (Demšar, 2006) are followed to perform the statistical analysis of the accuracy results, which is based on the use of nonparametric tests. More specifically, the following methodology is employed. First, the Friedman test (Friedman, 1940) is applied to contrast the null hypothesis that all the learning algorithms obtained the same results on average. If the Friedman test rejects the null hypothesis, the non-parametric Nemenyi test (Nemenyi, 1963) is used to compare all results to each other, where the result with lowest rank is considered as the best. The Nemenyi test defines that two results are significantly different if the corresponding average rank differs by at least a critical difference called CD . This method based on the CD is used for showing graphicly the most competitive area. CD is computed as Equation 5.3, where n_{res} and n_{ds} are the number of results and of data sets respectively, and q_α is the critical value based on the studentized range statistic (Sheskin, 2007) with greater confidence ($P < 0.01$).

$$CD = q_\alpha \sqrt{\frac{n_{res} \cdot (n_{res} + 1)}{6 \cdot n_{ds}}} \quad (5.3)$$

5.4.2 Comparison of Individual Representation

As it has been explained in Section 4.3, CAOS includes some additional features to improve its performance such as the bloat control and the cluster merge. However, the application of both improvements depend on the individual representation and then, it would be necessary to identify

when they should be applied before to compare the individual representation between them. In the followings, the selection of the improvements to apply is performed and next the comparison of approaches is done in terms of qualitative and quantitative points of view and also in terms of the search space size and the computational performance. In addition to these analysis, the effect of the process initialization of the individuals is also analyzed to identify if a random initialization could overcome an optimal local solution.

5.4.2.1 Choosing the Best Configuration in Each Representation

Four different scenarios are studied, depending on the activation or not of the bloat control and the clusters merging. The effects of these configurations are analyzed in terms of (1) accuracy using computed by the Adjusted Rand index (see Figure 5.2), (2) the number of individuals dealt in the genetic cycle (see Figure 5.3) and, (3) the number of clusters in the retrieved solutions (see Figure 5.4). Figures present the results obtained by the Nemenyi test using the 70 data set applying a ten fold stratified cross validation with 10 seeds, being the horizontal axis the ranking index. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. For example, *Bdv* represent the solution with the best value of the Davies-Bouldin index from all the solutions obtained applying bloat control to the standard CAOS cycle.

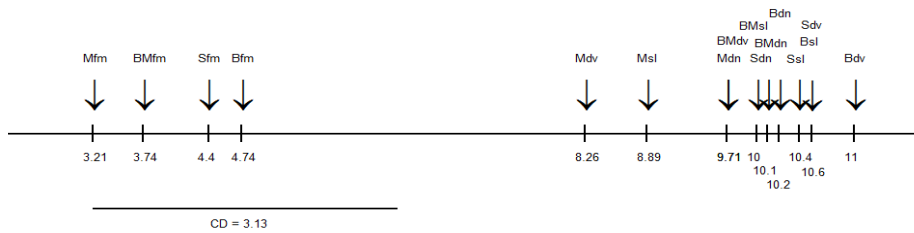
Impact of Bloat Control. The accuracy rank of the retrieved solutions of each representation is slightly worse when the bloat control is applied as Figure 5.2 shows. This is due to the fact that some solutions are removed from the Pareto set because they are similar to others and some good solutions obtained with the combination of them cannot be created. However, Figure 5.3 shows that, applying the bloat control, the number of individuals dealt in the genetic cycle is always inferior for the prototype-based representation, the reduction average is 27% respect the standard cycle individuals. It also shows that for the label-based the number of individuals are significantly reduced (42% of reduction respect the standard configuration) when the bloat control is applied. Nevertheless, to apply the bloat control to the graph-based representation does not always reduce the number of individuals and the average reduction of them are only 8% respect the standard configuration. This means that the prototype-based and the label-based approaches explore more search space, as they obtain more similar individuals that are not dominated. A lot of these individuals are situated at the part of the Pareto set with more connectedness, which it means that they have a lot of clusters. For this reason, in these representations, to eliminate similar solutions it slightly affects to the accuracy results and a lot of individuals are omitted improving the computational time of the algorithm.



(a) Prototype-based

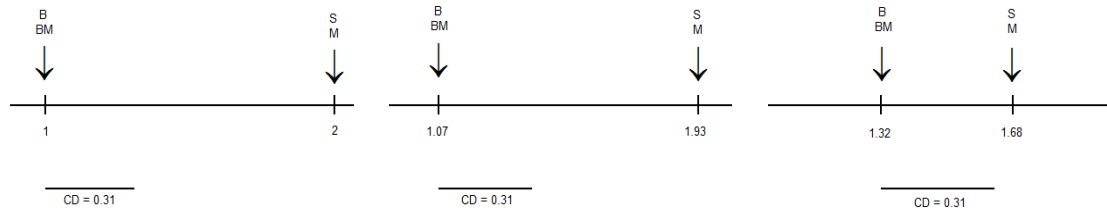


(b) Label-based



(c) Graph-based

Figure 5.2: Accuracy rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank is the best one. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.



(a) Prototype-based

(b) Label-based

(c) Graph-based

Figure 5.3: Number of individuals rank with Nemenyi test of each CAOS configuration of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank indicates that the configuration deals with less number of individuals in the genetic cycle. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.

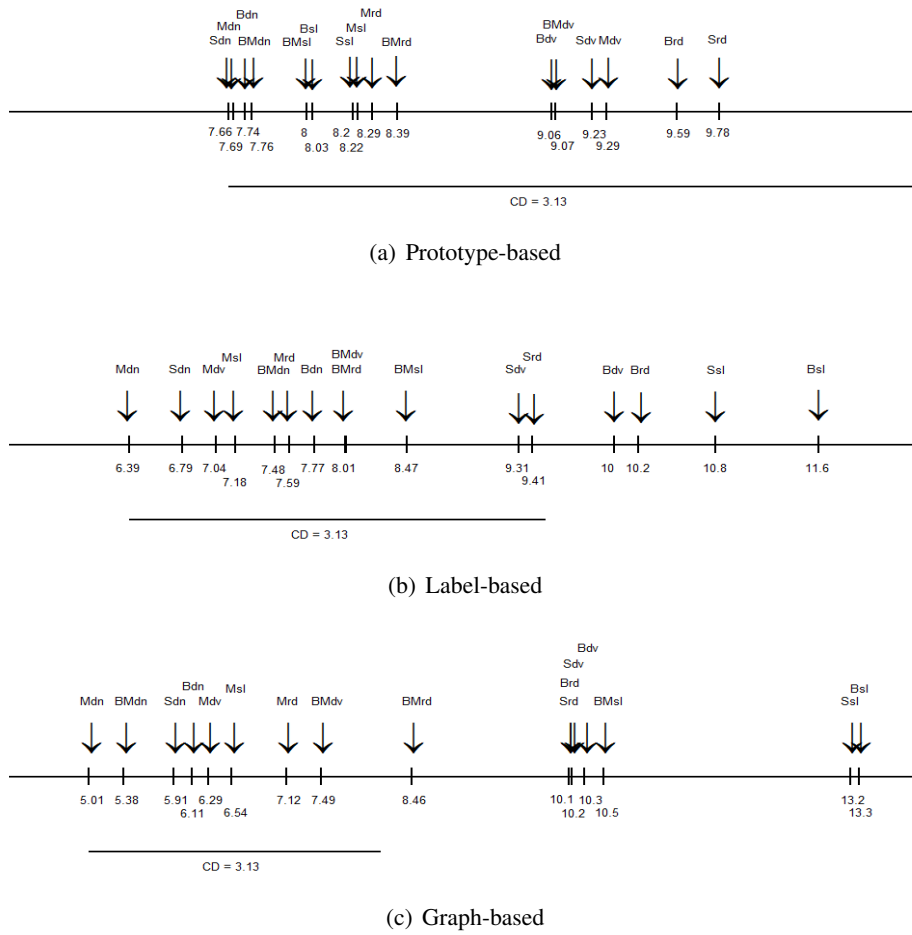


Figure 5.4: Number of clusters rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations. The lower rank indicates that the solution has less number of clusters. *E*, *B*, *M* and *BM* represent CAOS configured with any improvement, with bloat improvement, with cluster merge improvement and with both improvement respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.

Impact of Clusters Merging. The accuracy rank of each representation is slightly better when the clusters of the retrieved solutions are merged as Figure 5.2 shows. In addition, Figure 5.4 shows that the graph-based and the label-based representations obtain solutions with significant less clusters when the final clusters are merged, and the average reduction of clusters respect the non-merged solution is 42% and 22% respectively. Nevertheless, the clusters reduction of the prototype-based is slightly better with the supervised solution, but there is no improvement with the solutions retrieved with validation index. The solutions of the graph-based and the label-based representations are very compact because they have a lot of clusters, but the most of them had few instances. This is due to the fact that the representation tends to isolate the instances that are not very similar to the other ones. Above all, the Davies-Bouldin and Silhouette indexes solutions are

very sensitive to very compact clusters, avoiding the separation between them. This can be solved restricting the maximum number of clusters or the minimum number of instances that a cluster can contain, as is done in (Handl and Knowles, 2007). However, we consider that this restriction decreases the flexibility of the solutions because if there is an outlier or data noise, the system is going to cluster it with other different instances obtaining clusters with a long diameter.

Selection of Best Configuration. The analysis showed that the best configuration for the prototype-based representation is to apply the bloat control to deal with less individuals to improve the computational time of the algorithm, and it is not necessary to merge the clusters of the final solution due to the fact that it does not improve the solutions retrieved with the validation indexes. The best configuration for the graph-based representation is to merge the retrieved solutions because the accuracy of the validation indexes solutions is improved, being unnecessary the bloat control as the computational time is not considerably improved and this does not compensate the lose of accuracy. However, the best configuration of the label-based representation uses the bloat control and merges the clusters of the retrieved solutions, as both actions in this case improved the computational time and reduce the number of clusters.

Effect of Random Initialization. At this point, the best configuration of each representation has been identified. The analysis was done with a heuristical initialization of the initial population as it is described in Section 5.3. Nevertheless, in more complex problems this heuristical initialization of the population cannot be enough to obtain initial good individuals. For this, a random initialization of the population has been applied to analyze if the representations are biased. A representation is considered biased when the search space depends on the individuals initialization. This bias was analyzed with respect to the accuracy of the algorithm. Figure 5.5 shows the results of this analysis for the retrieval solution of each representation. The heuristic initialization is represented by *H* and the random initialization is represented by *R*.

The accuracy rank of the random initialization for the graph based representation is worse than the heuristic one. In the label-based representation, the random initialization is also a little worse than the heuristic one. Nevertheless in the prototype-based representation this difference is lower, being practically negligible with the solutions retrieved with validation indexes. This means that the prototype-based representation is not biased respect to the individual initializations, and is able to explore longer search spaces. The reason of this is that the prototype-based representation when modify a centroid of one individual can modify several instances cluster, allowing to explore a huge search space. However, the tendency of the label-based and the graph-based representations is to affect few instances when a change is done in the individual, and for this the search space is smaller. If a heuristic initialization is applied, the individuals are situated in the suitable search space, avoiding the problem of exploration related to the graph-based and the label-based representations.

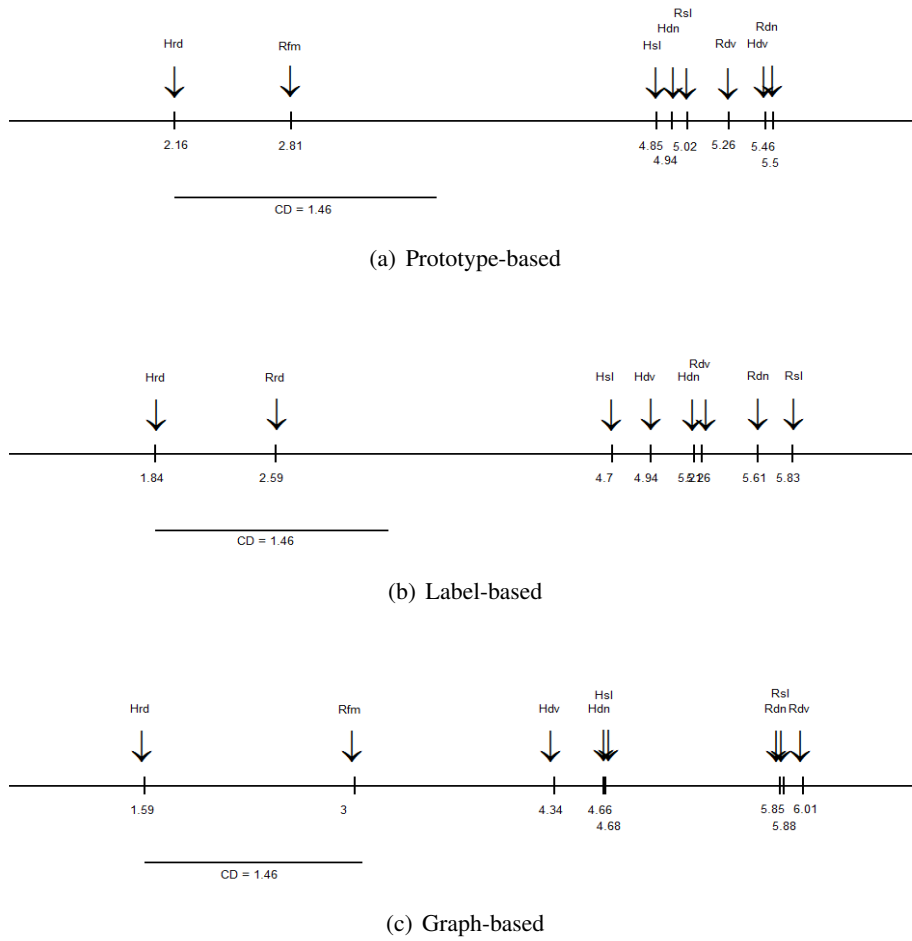
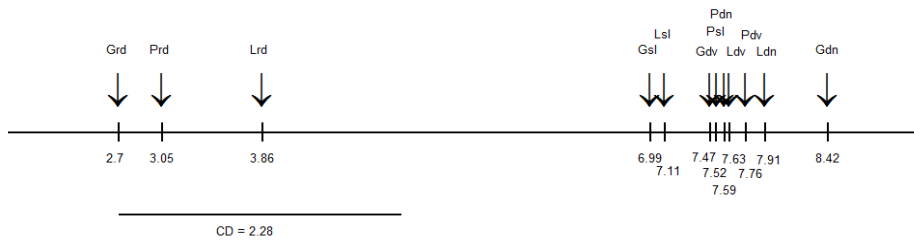


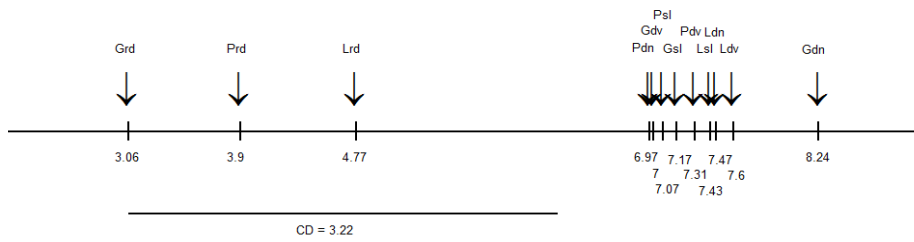
Figure 5.5: Accuracy rank with Nemenyi test of each CAOS solution of the (a) prototype-based, (b) label-based, and (c) graph-based representations obtained with heuristical and random initialization. The lower rank is the best one. *H* represents the heuristical initialization and *R* represents the random initialization. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.

5.4.2.2 Comparing Representations

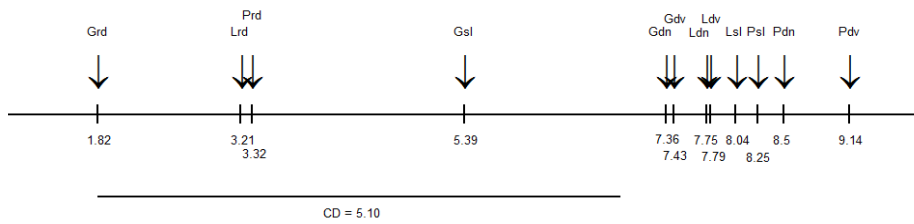
Now, the comparison of the best configuration for each representation found in the previous point is presented. They are compared in quantitative and qualitative terms. The quantitative comparison is made taking into account the accuracy value of the retrieved solution regarding to the original classes of the data set. However, the aim of clustering algorithms is to identify patterns in the data and not to classify it. The accuracy value inform if the instances have been well classified, but do not indicate the quality of the patterns identified. To complement this quantitative analysis, a qualitative analysis based on an external-criteria (Legeny et al., 2006) is used. It consists in the analysis of the clustering results by an expert on the problem domain. In our case, this analysis can be done with the artificial data sets, since the shape of each cluster is known.



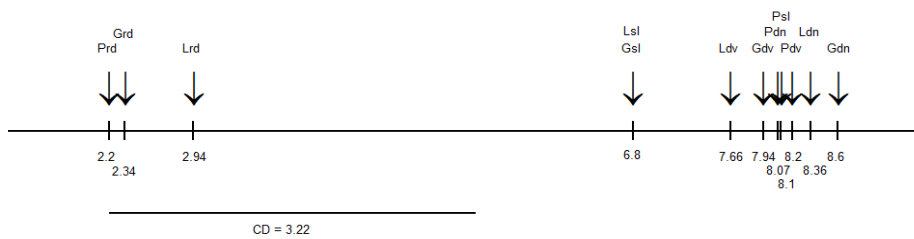
(a) All the data sets



(b) Artificial data sets



(c) Handmade artificial data sets



(d) Real data sets

Figure 5.6: Accuracy rank with Nemenyi test of the solutions of the best configuration of each CAOS representation for (a) all the data sets, (b) the artificial data sets, (c) the handmade artificial data sets, and (d) real data sets. The lower rank is the best one. *P*, *L* and *G* represent the prototype-based, the label-based and the graph-based representations respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.

Figure 5.6 shows the accuracy results obtained with all the data sets. The best configuration of the prototype-based, the label-based and the graph-based are represented at the figure by P , L and G , respectively. With all the data sets, each retrieved solution of the graph-based representation is better than the respective ones of the other representations. If the results are analyzed taking into account the data sets typology, for the artificial data sets the supervised solution of the graph-based representation are better than the other ones, but the non supervised solutions of each representation are very similar. Nevertheless, for the handmade data sets this difference increases, being the worst results these obtained by the non supervised solutions of the prototype-based representation. With the real data sets, the rank accuracy of the supervised solutions are very similar, however each one of the retrieved graph-based solutions are better than the respective ones of the other representations. It is important to highlight that in the four analysis presented, the solutions retrieved with the same method are not statistically different.

The accuracy analysis showed that the graph-based representation can obtain slightly better solutions than the prototype-based and the label-based representations for the artificial data sets. However, for the handmade data sets, which contains clusters of strange shapes, the graph-based and the label-based increases their accuracy regarding to the prototype-based representation. This is due to the fact that the graph-based and the label-based representations are more flexible than the prototype-based one, because they can allow any relationship between the instances. The prototype-based representation does not allow this since the clusters are built according to the Euclidean distances between the instances and the identified prototypes, obtaining clusters of spherical shape, that is a particular case of clusters with ellipsoidal shape. On the other hand, for the real data sets, the accuracy analysis showed that the prototype-based representation is slightly better than the other two representations, since it is less sensitive to noise and outliers, common in these kind of data sets.

Now, the ability of each representation to adapt to different cluster shapes is analyzed. It was explained that the graph-based and the label-based representations are able to adapt to complex shapes due to the type of representation used, on the other hand the prototype-based representation tends to obtain ellipsoidal clusters, being more difficult to it to identify patterns with complex shapes.

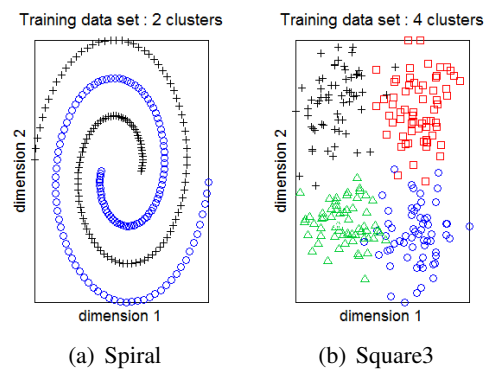


Figure 5.7: Graphical representation of the original classes of the (a) *spiral*, and (b) *square3* data sets.

Figure 5.8 and Figure 5.9 show the clusters found by each retrieved solution of each representation for two data sets, (1) *spiral* and (2) *square3*. The original classes of them are shown in Figure 5.7. These data sets clearly illustrate the aforementioned behavior. Figure 5.8 shows that the graph-based representation can properly identify the two complex shapes, retrieving the optimal solution with each retrieval method. However, even though the label-based can adapt to complex shapes in this case cannot identify the optimal clusters, dividing the spiral in a half with the supervised retrieval and dividing it in three pieces in the validation indexes solutions. The prototype-based representation cannot identify proper solutions, due to the fact that it cannot obtain ellipsoidal clusters with good compactness and separation, returning some solutions with more than 200 very small clusters. On the other hand there are the results showed at Figure 5.9, where the prototype-based representation can properly identify four compact and separate spherical clusters. The label-based representation can also identify four clusters but they are not as compact and separate as the clusters found by the prototype-based. It is important to highlight that the shape of them is not ellipsoidal. The graph-based representation cannot properly identify the four clusters. It tends to group the closest instances, for this here it obtains big clusters with the nearest instances and small clusters with the instances more separated.

This behavior is repeated in the other studied data sets. The graph-based representation can be useful to obtain clusters of complex shapes but it is not recommended to obtain clusters from scattered instances. On the other hand, the prototype-based representation can obtain clusters of good quality from scattered instances, but it only finds ellipsoidal clusters, being useless to identify clusters with complex shapes. The label-based representation can find clusters of several shapes but it does not explore the search space as well as the graph-based representation, and it not obtains as compact and separate clusters as the prototype-based representation.

5.4.3 Performance of CAOS Regarding Single-Objective Clustering Methods

In order to analyze if CAOS is competitive regarding to some single-objective clustering algorithms, they were applied to data sets with diverse features in order to analyze the performance of each one of the algorithms. It is important to highlight that the aim of MC algorithms is to identify patterns in complex domains where single objective algorithms may fail, nevertheless in order to analyze the competitiveness of CAOS it has been also tested with straightforward problems.

Figure 5.10 shows the accuracy rank of the compared algorithms. With all the data sets, the CAOS solutions retrieved with Adjusted Rand index are better than EM and significantly better than the other clustering algorithms. However, the solutions retrieved with validation indexes are worse than EM and k -means, but they are similar than x -means and SOM. With the artificial data sets, the differences between the CAOS supervised retrieved solutions and the not supervised ones are lower. The supervised solution of the graph-based and the prototype-based representations are still better than EM, but the label-based one is slightly worse than it. The solutions retrieved with validation indexes they are similar to k -means and better than x -means and SOM. In the real data sets, the supervised retrieved solutions are better than EM and k -means, and they are significantly

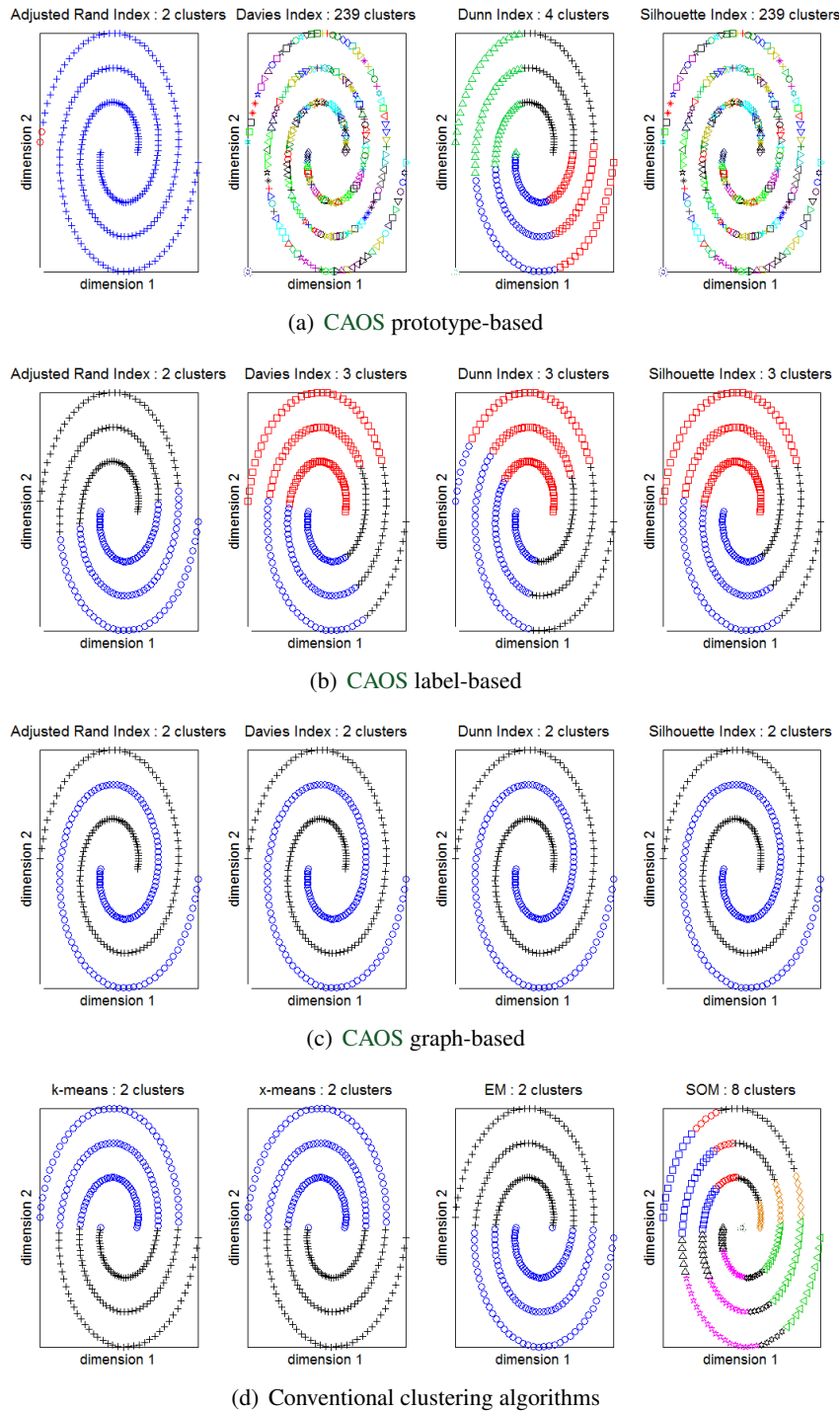


Figure 5.8: Graphical representation of the clusters found in the *spiral* data set by (a) CAOS prototype-based, (b) CAOS label-based, (c) CAOS graph-based, and (d) conventional clustering algorithms. The solutions presented at (a), (b) and (c) were retrieved with Adjusted Rand, Davies-Bouldin, Dunn and Silhouette indexes, from left to right. The solutions presented at (d) are *k*-means, *x*-means, EM and SOM from left to right.

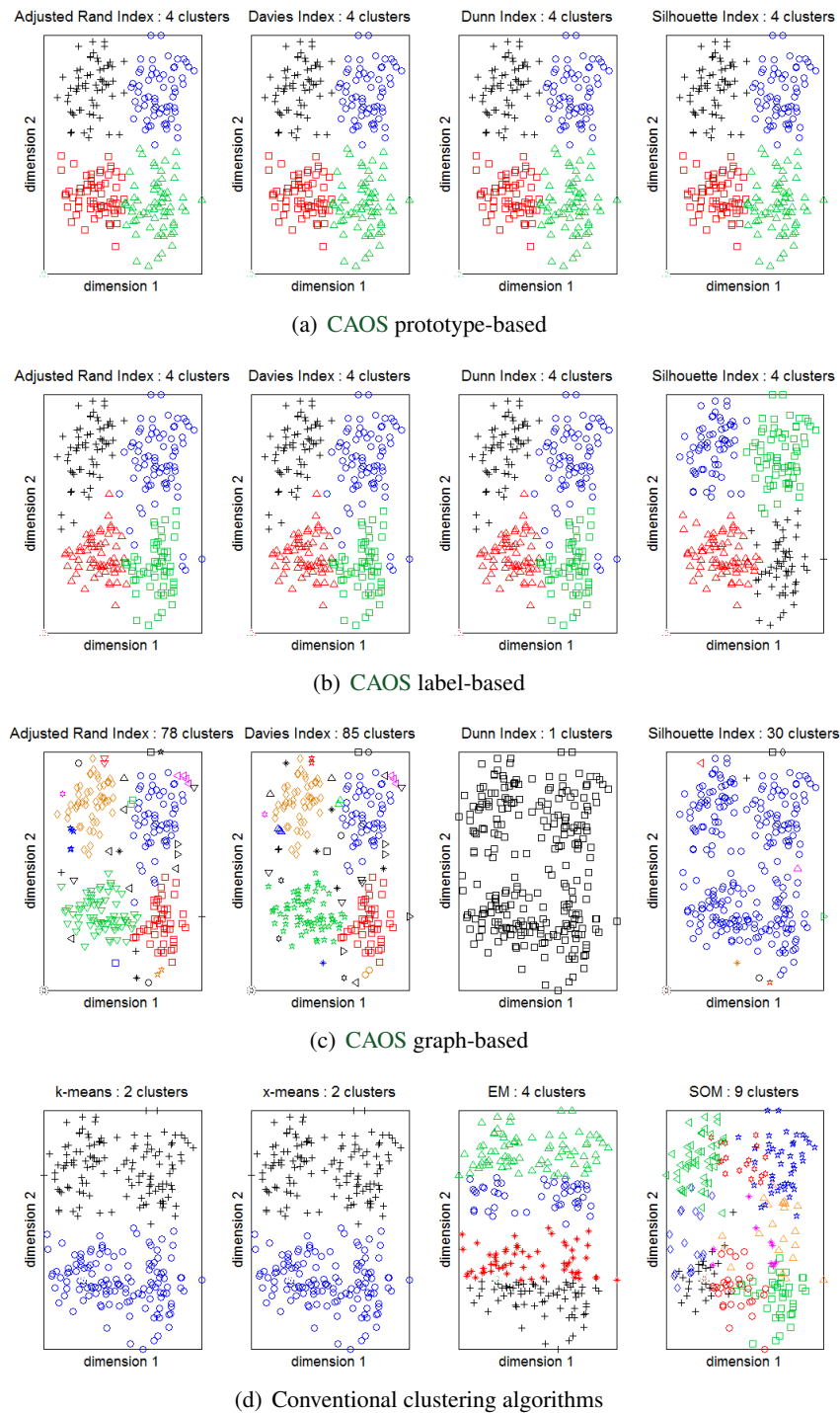


Figure 5.9: Graphical representation of the clusters found in the *square3* data set by (a) CAOS prototype-based, (b) CAOS label-based, (c) CAOS graph-based, and (d) conventional clustering algorithms. The solutions presented at (a), (b) and (c) were retrieved with Adjusted Rand, Davies-Bouldin, Dunn and Silhouette indexes, from left to right. The solutions presented at (d) are *k*-means, *x*-means, EM and SOM, from left to right.

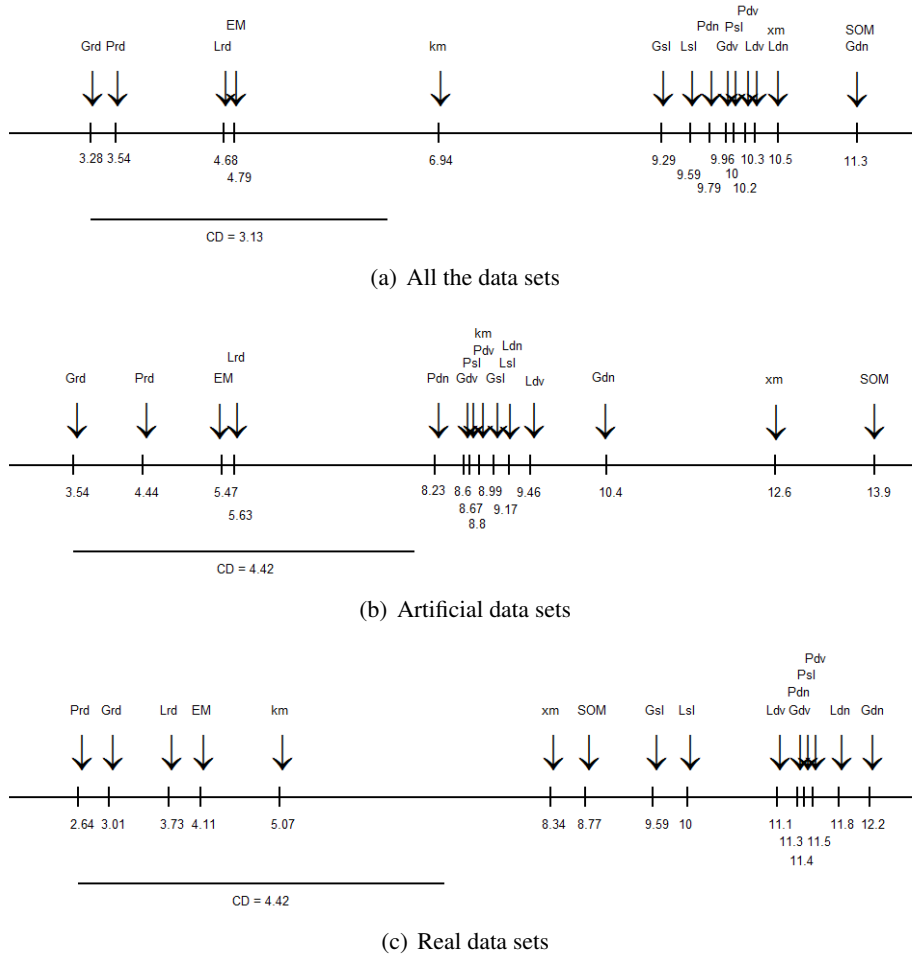


Figure 5.10: Accuracy rank with Nemenyi test of the solutions of the best configuration of each CAOS representation and the solutions of the single objective algorithms for (a) all the data sets, (b) the artificial data sets and (c) real data sets. The lower rank is the best one. *P*, *L* and *G* represent the prototype-based, the label-based and the graph-based representations respectively. In addition to these abbreviations, each retrieved solution is represented by *rd*, *dv*, *dn*, *sl* corresponding to the Adjusted Rand index, the Davies-Bouldin index, the Dunn index and the Silhouette index respectively. *km*, *xm*, *EM* and *SOM* represent the results of the *k*-means, *x*-means, EM and SOM algorithms respectively. *CD* indicates the value of the critical distance, representing with a line the area that is not significantly different respect the best ranked method.

different than the solutions obtained with the other algorithms. However, the solutions retrieved with validation indexes are not very different to the solutions of SOM and *x*-means.

These results show that the Pareto set of solutions obtained with CAOS has solutions of high quality in terms of accuracy. The solutions retrieved with the Adjusted Rand index are quite better than the solutions obtained by the other algorithms, but they take into account the original classes of the data set. Nevertheless, the solutions retrieved with validation indexes are not the better ones in accuracy terms, due to the fact that they are retrieved taking into account only the shape of the clusters. It is important to highlight that CAOS and *x*-means do not need as a parameter the number of clusters to find in the data set, and this is a handicap regarding to the other algorithms analyzed. The difference in accuracy between both algorithms is significantly, due to the fact that CAOS is

better than x -means for the majority of the configurations.

As it was explained previously, our aim is to identify patterns in a data set and not to classify instances. For these, a qualitative analysis is required. Figures 5.8 and 5.9 are representative examples of the performance of the single objective algorithms. K -means, EM, x -means and SOM can only divide the data set space according to one objective that is the intra-cluster variance; in spite of each one calculate the cluster prototypes in a different manner. However, CAOS try to optimize the intra-cluster variance (compactness) and the inter-cluster variance (separation) obtaining more understandable patterns.

For more details about the results, consult Table B.1 to Table B.4 in Appendix B, where the accuracy and number of clusters of each algorithm for each data set are shown.

5.4.4 Discussion

The experiments have shown the performance of the analyzed CAOS representation in quantitative terms according to the Adjusted Rand index and in qualitative terms according to the shape of the clusters. In accuracy terms the graph-based and the prototype-based representations have a similar performance, and the label-based is worse than them. Also, they showed that all the CAOS solutions retrieved with the Adjusted Rand index, which takes into account the original classes of the data set, are better than the solutions obtained by the single-objective clustering algorithms. The CAOS solutions retrieved with some unsupervised indexes lose some accuracy but they are also competitive. It is important to highlight that the solutions retrieved with the Adjusted Rand index are inside the Pareto set and are generated by CAOS, nevertheless they are not recovered with the unsupervised indexes, which take into account the shape of the clusters.

In terms of clusters shape, the graph-based representation is focused on obtaining clusters of complex shapes, but could be more sensitive to data noise. The prototype-based representations is focused on obtaining clusters of ellipsoidal shape, being more robust to data noise but less flexible to identify clusters with complex shapes. For these, it is considered that the prototype-based representation can be useful for scattered data sets, and the graph-based one is useful to identify clusters of complex shape. However, the label-based representation can obtain clusters with more flexible shape than the prototype-based one, and is more robust to the data noise than the graph-based representation. The experiments have also shown that the three approaches try to identify clusters according to the intra-cluster variance and the inter-cluster variance, obtaining understandable patterns that can help an expert on the data set domain. However, the single-objective clustering algorithms used tried to identify clusters according only to the intra-cluster variance, obtaining clusters very difficult to explain.

Moreover, in computational terms, the prototype-based representation and the label-based one are more sensitive to bloat effect. This is because both representations can slightly modify the individuals, obtaining very similar and non-dominated individuals in the Pareto set. On the other hand, the graph-based and the label-based representations need to merge the clusters of the retrieved solutions due to the fact that it could obtain solutions with a very high number of clusters since the

crossover and the mutation operator used can bias the individuals to the extreme of the Pareto set with high connectivity.

Finally, it is important to remark that the prototype-based can obtain good results with a random initialization of the individuals. Nevertheless, the other two representations are penalized when a heuristical initialization is not used. The prototype-based representation can be useful in large data sets where could be difficult to delimit the search space with a heuristical initialization, due to the fact that the exploration search space will be larger than in the other two representations.

5.5 Summary and Conclusions

The goal of this chapter was to identify the most suitable individual representation in MC algorithms using CAOS. To do this, the three most commonly used individual representations in EAs have been analyzed in MC algorithms, those are listed as follows: (1) prototype-based, (2) label-based and (3) graph-based. Moreover, the analysis was extended by comparing the results with respect to the most used single-objective clustering algorithms.

The results showed that, in terms of accuracy, each CAOS representation generates solutions in the Pareto set better than the solutions generated by the single-objective clustering algorithms analyzed. Moreover, in terms of clusters shape, CAOS solutions divides the data set space in more representative and well explained clusters than the single objective algorithms.

Label-based and graph-based representations can identify clusters of arbitrary shapes. However they have problems to explore properly the search space, their performance depends on the individual initialization and they are sensible to scattered data sets. Moreover, the size of the individuals of these representations are related to the size of the data set, therefore they consumes a big amount of memory when they are applied to large data sets. On the other hand, the prototype-based representation can only find ellipsoidal clusters. However it explores more search space than the other two representations, so it is independent to the initialization of the population. Moreover, it is more robust to scattered data sets and the individuals are not related to the size of the data set, so the memory usage is lower. As conclusion, it must be emphasized that these observations show that the selection of the most suitable CAOS representation depends on the domain of the problem because there is not an individual representation that works properly for all kind of problems.

Finally, it is important to highlight that CAOS is based on evolutionary algorithms, and the main lack of these techniques is that they are expensive in terms of computational time and memory usage, therefore it is necessary to scale-up them. As discussed above, the prototype-based representation does not need to scale-up its memory usage and it only needs to scale-up the computational time of the system. Thus, this representation can be useful to scaling-up multiobjective evolutionary clustering algorithms, so this analysis set the basis for the research on this kind of techniques applied to large data sets which is explained in the next chapter. Also, we can consider as further work the analysis of other representations that can be more flexible according to the shape of the clusters than the prototype-based representation but with similar capabilities related

to the memory usage.

A paper related to this contribution has been submitted to a journal under the title "Large-Scale Experimental Evaluation of Cluster Representations for Multiobjective Evolutionary Clustering".

Chapter 6

Large Data Management

The techniques based on evolutionary algorithms are expensive in terms of computational time and memory usage, and this lack is increased when they are applied to large data sets. Multiobjective evolutionary clustering algorithms are not an exception, so specific strategies are required to ensure their successful scalability when facing large data sets. This chapter proposes the application of data subset techniques for scaling-up this kind of algorithms and analyzes the impact of several stratification methods. The experiments show that the use of this technique can improve the performance of these kind of algorithms without considerably penalize the accuracy of the final clustering solution.

6.1 Motivation

One of the present challenges in Data Mining is to allow systems to work with large data sets in a reasonable computational time and memory usage without considerably penalizing their accuracy (Obradovic and Vucetic, 2004; Kargupta et al., 2009). This is especially critical in the approaches based on EAs due to their highly cost in terms of computational time and memory usage when they are applied to a big amount of data since they do an intensive use of computations (Freitas, 2002). From the whole process, the evaluation step is the most time consuming step because each individual has to be assessed with respect to all the instances of the data set. Two of the most used strategies for scaling-up EAs are the usage of a Parallel EA (Cantu-Paz, 2000) and the usage of data subsets (Bacardit, 2004; Cano et al., 2008; Derrac et al., 2010). The first strategy distributes the computational cost of the evaluation step by parallelizing the evaluation of the individuals. Thus, it is necessary to adapt or redefine the algorithm for being able to parallelize it in a environment with several processors. Moreover, the parallelization may imply an additional communication cost that could decrease the performance achieved with the compute distribution. On the other hand, the second strategy use a sample (data subset) from the original data set to evaluate the individuals. Thus, they are easier to introduce in the systems but the definition of the size of subsets and the selection of their elements are not trivial and they influence the performance of the algorithm if the data subsets are not sufficiently representative.

This chapter proposes the application of data subset techniques for scaling-up MC algorithms based on EAs for analyzing their impact in terms of computational time and memory usage. The proposed data subset approach splits the complete data set in several strata and it uses a stratum in each generation following a Round Robin policy to avoid bias problems. An ideal stratified strategy is to map the initial data set into disjoint strata of equal size and with equal class distribution and where the number of strata is defined by the user (Cano et al., 2006). However, clustering problems are unsupervised and classes cannot be used to split the instances into representative strata because they are unknown. For this reason, two strategies are studied to solve this lack: (1) the generation of random strata and (2) the generation of the strata according to clusters distribution using a fast and approximative clustering algorithm.

Stratification strategies are integrated in CAOS and a comparison between the results obtained using the aforementioned data subset strategies and the obtained using the complete data set is presented in order to analyze the performance of the stratification methods with different artificial and real-world data sets. To carry out this, the approaches are compared along 75 synthetic large data sets and with 25 real-world problems from the UCI repository (Asuncion and Newman, 2010). It is important to highlight that each strategy has been applied with data subsets of different size in each one of the tested data sets.

Next sections briefly summarizes the related work on data subsets applied to clustering, describe the stratification strategies, present the experimentation and discuss the results.

6.2 Related Work

There are two main ways to work with data subsets: using only one of the built data subsets, or using alternatively all the data subsets. The algorithm CLARA (Clustering LARge Applications) (Kaufman and Rousseeuw, 1990), one of the most representative algorithms for clustering large data sets, works under the first approach idea. This algorithm is based on selecting randomly a sample from the entire data set and, subsequently, it finds k medoids of the sample using only the built sample. After this, all the instances of the entire data set are assigned to the most similar medoid. The execution of the entire process is repeated five times, and the solution with less dissimilarity is returned as solution. Following this idea, other methods consist in extracting randomly several samples from the entire data set and applying the same clustering algorithm to each one of the samples obtaining several clustering results. After this, all the obtained results are merged in a single clustering solution. Hore et al. (Hore et al., 2009) proposed to use k -means or fuzzy k -means algorithms with large data. The idea is to obtain a set of jointed or disjointed samples and apply one of the two algorithms to each sample to obtain several clustering results. The last step consists in doing a consensus between each clustering result to obtain a final clustering solution as in ensemble clustering. The drawback of using only one sample to obtain the clustering results is that it is necessary to execute the algorithm several times or apply it to different data subsets in order to avoid the bias of using only one sample. Moreover, only a part of the entire data set

is used. For this, it can be useful the approaches based on use all the data subsets to obtain the clustering results in a single execution.

ILAS (Incremental Learning by Alternating Strata) (Bacardit, 2004) is a technique based on Evolutionary Algorithms for classification problems based on dividing the training set into several strata based on using a different stratum in each iteration of the evolutionary algorithm using a round-robin policy. Thus, the individuals are evaluated with all the strata, avoiding any bias of the data and increasing the generalization of the individual. The strategy followed in this chapter is based on the ILAS algorithm but applied to MC problems. The entire data set is divided in several strata that are alternated in each of the generations of the evolutionary algorithm. In each generation, the clustering method is evaluated with the corresponding stratum. This approach seems to be a good start point to scale-up the system.

6.3 Data Subset Strategies

Reducing the amount of data used by an algorithm is a smart approach to reduce the computational cost of evolutionary-based machine learning techniques and it could also improve the accuracy of the system (Bacardit, 2004). In this chapter, we want to scaling-up a MC algorithm based on EAs by dividing a data set in several stratified subsets and using them alternatively during the algorithm process in order to avoid bias.

Next points detail the analyzed approaches to build the strata and how to use it in MC. Finally, the impact of the use of these strategies in the CAOS algorithm in terms of computational cost and memory usage is described.

6.3.1 Creation of Strata

The main idea of data subset strategies based on strata is to map the initial data set into disjoint data subsets (strata) of equal size and with equal class distribution (Bacardit, 2004; Cano et al., 2006), where the number of strata is selected by the user (see Algorithm 6.1). However, the strata cannot be generated according to the classes because in clustering problems the classes are unknown. To avoid this lack, two approaches to divide the data set can be used:

- **Random Strata.** It randomly assigns the instances to each one of the strata as Algorithm 6.2 shows.
- **Strata based on Clusters.** It uses a fast and approximative clustering technique to create a partition of the original data set. Next, the data set is stratified according to the obtained clusters, that is, it assigns the instances to each stratum respecting in it the same cluster distribution of the instances than in the clustered original data set. The process is described in Algorithm 6.3. The clusters are found with the Subtractive Clustering algorithm (Chiu, 1994) applied to the original data set, which is an efficient and non-iterative method for estimating cluster centers. It is usually used to determine the number of clusters and their

initial values for initializing iterative optimization-based clustering algorithms. The lack of this strategy is the $O(m^2)$ computational cost, where m is the number of instances of the original data set, because with very large data sets can be expensive in computational terms. Nevertheless, in terms of spatial cost it only needs the data set information and a list with the prototypes of each cluster.

Finally, both approaches require the definition of the number of strata which will influence in the algorithm performance. As the number of strata increases the computational time decreases but pattern extraction becomes more complex due to the lack of information. It is important to highlight that the idea of these strategies is to obtain data with similar distribution in each stratum, and this only can be possible if the size of each one is not very small.

6.3.2 Evolution Based on Strata

The process consists in changing the stratum in each iteration of the evolutionary algorithm using a round-robin strategy. Thus, the evolutionary process avoid the bias produced when only one stratum is used. If the stratum is changed in each cycle, the final individuals can generalize more than using only one of the strata.

The introduction of these strategies does not modifies the main process of the algorithm. More precisely, the generation of strata is done before line 1 of the Algorithm 4.1, and the change of stratum is done between lines 6 and 7. It is important to highlight that these strategies can be applied to CAOS with the prototype-based representation due to the fact that it defines the individuals by the prototypes of the clusters and then (1) the memory usage of the individuals does not depend on the size of the data set and (2) the individuals are independent of the instances. Thus, the algorithm can work with different instances of the data set in each iteration.

6.3.3 Computational Performance Models

The objective of the strategies based on data subsets is to reduce the computational time and memory usage of a system without considerably penalize the accuracy of the solutions. This section

```

1 Let numStrata be the number of strata to generate
2 Let Strata be a vector of size numStrata where each position is initially an empty list of instances
3 Let I be a vector of size numClasses where each position stores a list of the instances of the same class
4 stratum = 0
5 class = 0
6 while (class < numClasses) do
7   while (I[class] !=  $\emptyset$ ) do
8     Randomly select an instance i from I[class]
9     Add i to Strata[stratum]
10    Erase i from I[class]
11    stratum = (stratum + 1) mod numStrata
12  class = class + 1
13 return Strata

```

Algorithm 6.1: Strata generation based on classes.

```

1 Let numStrata be the number of strata to generate
2 Let Strata be a vector of size numStrata where each position is initially an empty list of instances
3 Let I be a list of all the instances of the data set
4 stratum = 0
5 while (I !=  $\emptyset$ ) do
6   Randomly select an instance i from I
7   Add i to Strata[stratum]
8   Erase i from I
9   stratum = (stratum + 1) mod numStrata
10 return Strata

```

Algorithm 6.2: Strata generation based on random instances selection.

```

1 Obtaining the instances clustered in numClusters clusters by applying the Subtractive Clustering algorithm to the complete data set
2 Let numStrata be the number of strata to generate
3 Let Strata be a vector of size numStrata where each position is initially an empty list of instances
4 Let I be a vector of size numClusters where each position stores a list of the instances assigned to the same cluster
5 stratum = 0
6 cluster = 0
7 while (cluster < numClusters) do
8   while (I[cluster] !=  $\emptyset$ ) do
9     Randomly select an instance i from I[cluster]
10    Add i to Strata[stratum]
11    Erase i from I[cluster]
12    stratum = (stratum + 1) mod numStrata
13  cluster = cluster + 1
14 return Strata

```

Algorithm 6.3: Strata generation based on approximative clusters.

theoretically analyze the improvement in the performance of a MC algorithm based on EAs using the CAOS process as reference. The CAOS process can be divided in two main blocks: the initialization process and the clustering process. The initialization process is focused on precalculating the distances between the instances and the nearest neighbors to speed up the clustering process avoiding the repetition of calculations. On the other hand, the clustering process is referred to the evolutionary cycle that obtains the Pareto set of solutions. CAOS algorithm has the same initialization and clustering cost independently of the data subset strategy used (based on classes, random or based on clusters) but it depends on the number of strata used. However, the time of both processes is extremely lower in comparison with the time spent when the complete data set is used as Table 6.1 describes. It should be emphasized that both times are reduced when the size of the stratum is decreased, that is, when the number of strata increases. Nevertheless, the use of strata requires an additional cost for building them. According to this, the strategies based on random instances selection and based on classes need only one data scan to build the strata and their cost is $O(m)$, where m is the number of instances of the complete data set. In contrast, the strategy based on approximative clusters has a higher cost due to the cost related to the subtractive clustering technique ($O(m^2)$).

As it has been explained above, CAOS precalculate the distances between all the instances of

Algorithm	Initialization cost	Clustering cost
CAOS – CD	$O(m^3 \cdot \ell)$	$O(g \cdot IP \cdot m \cdot \bar{n}_{cd} \cdot t)$
CAOS – DS	$O(numStrata \cdot (\frac{m}{numStrata})^3 \cdot \ell)$	$O(g \cdot IP \cdot \frac{m}{numStrata} \cdot \bar{n}_{ds} \cdot t)$

Table 6.1: Computational cost of CAOS applied to the complete data set and to data subsets (CAOS – CD and CAOS – DS respectively) broken down in initialization cost and clustering cost. Where g is the number of generations, $|IP|$ is the internal population size, m and t are the number of instances and attributes of the data set respectively, \bar{n}_{cd} is the average of the number of clusters of the individuals (the minimum number of clusters is 1 and the maximum m), \bar{n}_{ds} is the average of the number of clusters of the individuals (the minimum number of clusters is 1 and the maximum $\frac{p}{numStrata}$), ℓ is the percentage of the nearest elements taken into account, and $numStrata$ is the number of strata generated.

Algorithm	Memory usage to store distances	Memory usage to store nearest neighbors
CAOS – CD	$m^2 \cdot sizeof(float)$	$(\ell \cdot m)^2 \cdot sizeof(integer)$
CAOS – DS	$(\frac{m}{numStrata})^2 \cdot sizeof(float)$	$numStrata \cdot (\frac{\ell \cdot m}{numStrata})^2 \cdot sizeof(integer)$

Table 6.2: Memory usage of CAOS applied to the complete data set and to data subsets (CAOS – CD and CAOS – DS respectively) to store the nearest neighbors. Where m is the number of instances of the data set, ℓ is the percentage of instances considered neighbors, $numStrata$ is the number of strata generated and $sizeof(data\ type)$ is the size in bytes of the data type.

the data set and the nearest neighbors of each instance to speed up the clustering process. Thus, the memory usage would be extremely high if the complete data set is analyzed when a large data set is used. Applying any of the three strategies of data subset construction, the memory usage is considerably reduced as Table 6.2 shows. Even the computational time and the memory usage of the MC algorithm is considerably reduced, the accuracy of the method can be penalized due to the fact that less data is used to obtain the clustering solutions. In the next section, the accuracy of CAOS using data subsets and the complete data set is compared in order to analyze when strata maintains the accuracy.

6.4 Experiments, Results and Discussion

This section evaluates the impact of reducing the volume of data used for training the system on the quality of the clusters. To carry out this, the CAOS performance using the three different data subset strategies described in section 6.3 are compared, through a collection of 100 data sets, with respect to the approach that uses all the instances. The performance is considered taking into account the accuracy using the Adjusted Rand index value of the solution returned by CAOS and the computational time required to find it. Next, the data sets, the experiments and the results of the comparison are presented.

6.4.1 Experimental Methodology

This section presents the experimental methodology followed to evaluate the data subsets strategies in CAOS. The analysis enables us to emphasize the benefits and the drawbacks of each one. In the followings, we provide details about (i) the data sets collection chosen for the experimentation, (ii) the CAOS configuration, and (iii) the comparison metrics.

Test Bed. The experimentation compares the algorithms performance using different typologies of artificial and real-world problems. Concretely, 75 artificial data sets were created according to different number of instances (from 800 to 24000), attributes (from 2 to 100) and classes (from 2 to 30). They were built adapting the tool used in (Handl and Knowles, 2007) where three parameters are used to create the data sets: the number of attributes, the number of classes related to the number of instances, and the separation between the classes. Each class has a data distribution for each attribute, which can only have numerical values. The distribution can be a normal or uniform distribution, and it is randomly selected to model each attribute. Also, the separation between classes were modeled, obtaining 25 data sets with well-separated classes, other 25 data sets with nearer classes, and the last 25 with overlapped classes. On the other hand, other 25 real-world problems were selected according to different number of instances (from 150 to 58000), attributes (from 2 to 60) and classes (from 2 to 26) from the UCI repository (Asuncion and Newman, 2010) and listed in Table 6.3.

CAOS Configuration. The CAOS representation used is prototype-based due to the fact that the other two representations do not allow to work with different instances of the data set in each iteration, also it is important to highlight that the prototype-based representation uses individuals that there are not related to the size of the data to group because they only store the prototype of each cluster, and this is an important feature when it is applied to large data sets. CAOS with each one of the data subset strategies was run with 10 different seeds and with the following parameters (see Section 4 for notation details): ℓ is 5% of the number of instances used, the maximum size of the initial population is 100, N_{EP} is 1000, N_{IP} is 50, N_{niches} is 5, the number of generations is 400,

Data set	nI	nA	nC	Data set	nI	nA	nC
balance	625	4	3	pim	768	8	2
biopsia	1027	24	2	segment	2310	19	7
bpa	345	6	2	shuttle	58000	9	7
dermatology	366	35	6	sonar	208	60	2
ecoli	336	8	8	thyroids	215	5	2
glass	214	9	6	transfusion	748	4	2
heart-statlog	270	13	2	vehicle	846	18	4
ionosphere	351	34	2	waveform	5000	40	3
iris	150	4	3	wdbc	569	30	2
letter-rec	20000	16	26	wisconsin	699	9	2
liver-disorders	345	6	2	wpbc	198	33	2
magic	19020	10	2	yeast	1484	9	10
pendigits	7494	17	10				

Table 6.3: Summary of the characteristics of the 25 real-world data sets used. The columns are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).

P_c is 0.7 and P_μ is $1/m$. The minimum and maximum number of clusters for the initial individuals is 2 and 20% of m respectively.

Comparison Metrics. The performance of the three approaches based on data subsets (CAOS_{DS}) were compared with respect to the approach that uses the complete data set (CAOS_{CD}) in terms of accuracy and computational time. The accuracy is compared using the Adjusted Rand index that is based on the initial classes of the data set, where 1 is the best accuracy (all the clusters correspond to the original classes) and 0 the worst. The computational time represents the sum of the precalculation time and clustering time. The first one includes the time needed to build the data subsets and to precalculate the distance and nearest neighbors structures necessary to the clustering process. The second one is referred to the time needed to do the evolutionary process that obtains the Pareto set of solutions. Finally, each CAOS_{DS} strategy is executed dividing the original data set in 2, 3, 4, 5, 10, 15, 20 and 25 data subsets which means the 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of the instances of the original data sets are considered in each data subset respectively.

On the other hand, the recommendations pointed out by Demšar (Demšar, 2006) were followed to perform the statistical analysis of the accuracy of the algorithms, which was based on the use of nonparametric tests. More specifically, we followed the process given in (García and Herrera, 2008) to compare them using the software freely provided by the authors¹. First, the Friedman's test (Friedman, 1940) with $\alpha = 0.001$ was applied to contrast the null hypothesis that all the learning algorithms obtained the same results on average. Then, if the Friedman's test rejected the null hypothesis, pair-wise comparisons were performed by means of the Holm's step-down procedure (Holm, 1979). Following this procedure, we could distinguish pairs of learners that performed significantly differently.

6.4.2 Comparison of Results

The performance of CAOS_{CD} and the three CAOS_{DS} strategies were empirically tested with artificial data sets and with real-world data sets separately because we want to analyze the performance of them in different situations.

Table 6.4 shows the results of the three CAOS_{DS} strategies and CAOS_{CD} for artificial data sets from a statistical point of view using the Holm's test. According to this analysis, the three CAOS_{DS} strategies are significantly worse in terms of accuracy regarding CAOS_{CD}, independently of the number of instances considered. Nevertheless, the three CAOS_{DS} are not significantly different in terms of accuracy between them. From a quantitative point of view, Figure 6.1(a) shows the average of accuracy difference among the three CAOS_{DS} strategies and CAOS_{CD}. Globally, the strategies based on classes and on clusters follow a similar pattern, and the accuracy is not considerably decreased until less than the 20% of the instances are used. On the other hand, the strategy based on the random selection seems to underperform the other two due to the fact that the classes structure is complex in some data sets and the random strategy is not able to build representative strata.

¹<http://sci2s.ugr.es/sicidm>

Instances used	Strategies	CAOS – CD	CAOS – DS – Classes	CAOS – DS – Random	CAOS – DS – Clusters
50%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		–	+
	CAOS – DS – Random	⊖	+		+
	CAOS – DS – Clusters	⊖	–	–	
34%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		+
	CAOS – DS – Clusters	⊖	–	–	
25%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	–	+	
20%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	–
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	+	+	
10%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	–	+	
7%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	–
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	+	+	
5%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		+
	CAOS – DS – Clusters	⊖	–	–	
4%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	–	+	

Table 6.4: Comparison of the algorithms in the artificial data sets using Holm’s procedure with $\alpha = 0.05$. The algorithms compared are CAOS using the CAOS_{CD} and the three CAOS_{DS} strategies to generate data subsets: based on classes, random and based on clusters; represented by CAOS – DS – Classes, CAOS – DS – Random and CAOS – DS – Clusters respectively. The results are shown for 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of information used from the complete data set. The symbols ⊕ and ⊖ show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column. Similarly, the symbols + and – denote a non-significant higher/lower results.

Instances used	Strategies	CAOS – CD	CAOS – DS – Classes	CAOS – DS – Random	CAOS – DS – Clusters
50%	CAOS – CD	The Friedman’s test cannot reject the null hypothesis that all the learning algorithms obtain the same results on average.			
	CAOS – DS – Classes				
	CAOS – DS – Random				
	CAOS – DS – Clusters				
34%	CAOS – CD		⊕	+	⊕
	CAOS – DS – Classes	⊖		–	–
	CAOS – DS – Random	–	+		+
	CAOS – DS – Clusters	⊖	–	+	
25%	CAOS – CD		⊕	+	⊕
	CAOS – DS – Classes	⊖		–	–
	CAOS – DS – Random	–	+		+
	CAOS – DS – Clusters	⊖	–	+	
20%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		+
	CAOS – DS – Clusters	⊖	–	–	
10%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		–	–
	CAOS – DS – Random	⊖	+		–
	CAOS – DS – Clusters	⊖	+	+	
7%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		+
	CAOS – DS – Clusters	⊖	–	–	
5%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		+
	CAOS – DS – Clusters	⊖	–	–	
4%	CAOS – CD		⊕	⊕	⊕
	CAOS – DS – Classes	⊖		+	+
	CAOS – DS – Random	⊖	–		–
	CAOS – DS – Clusters	⊖	–	+	

Table 6.5: Comparison of the algorithms in the real-world data sets using Holm’s procedure with $\alpha = 0.05$. The algorithms compared are CAOS using the CAOS_{CD} and the three CAOS_{DS} strategies to generate data subsets: based on classes, random and based on clusters; represented by CAOS – DS – Classes, CAOS – DS – Random and CAOS – DS – Clusters respectively. The results are shown for 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of information used from the complete data set. The symbols ⊕ and ⊖ show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column. Similarly, the symbols + and – denote a non-significant higher/lower results.

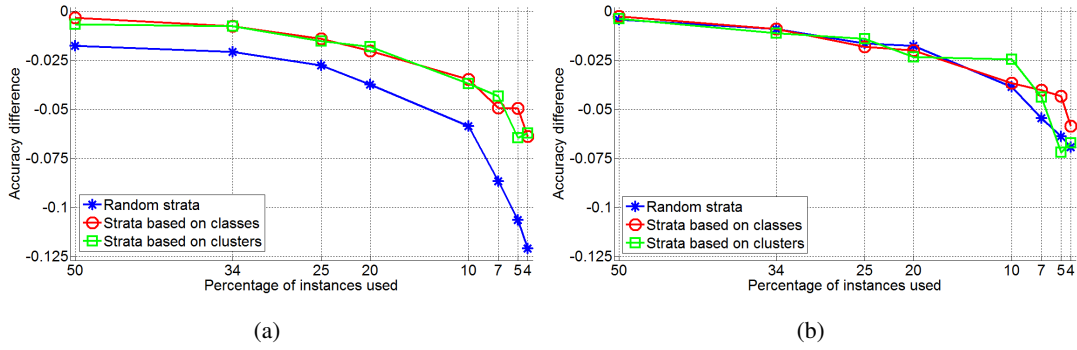


Figure 6.1: Accuracy difference of the three CAOS_{DS} strategies regarding CAOS_{CD}. (a) Artificial data sets and (b) real-world data sets.

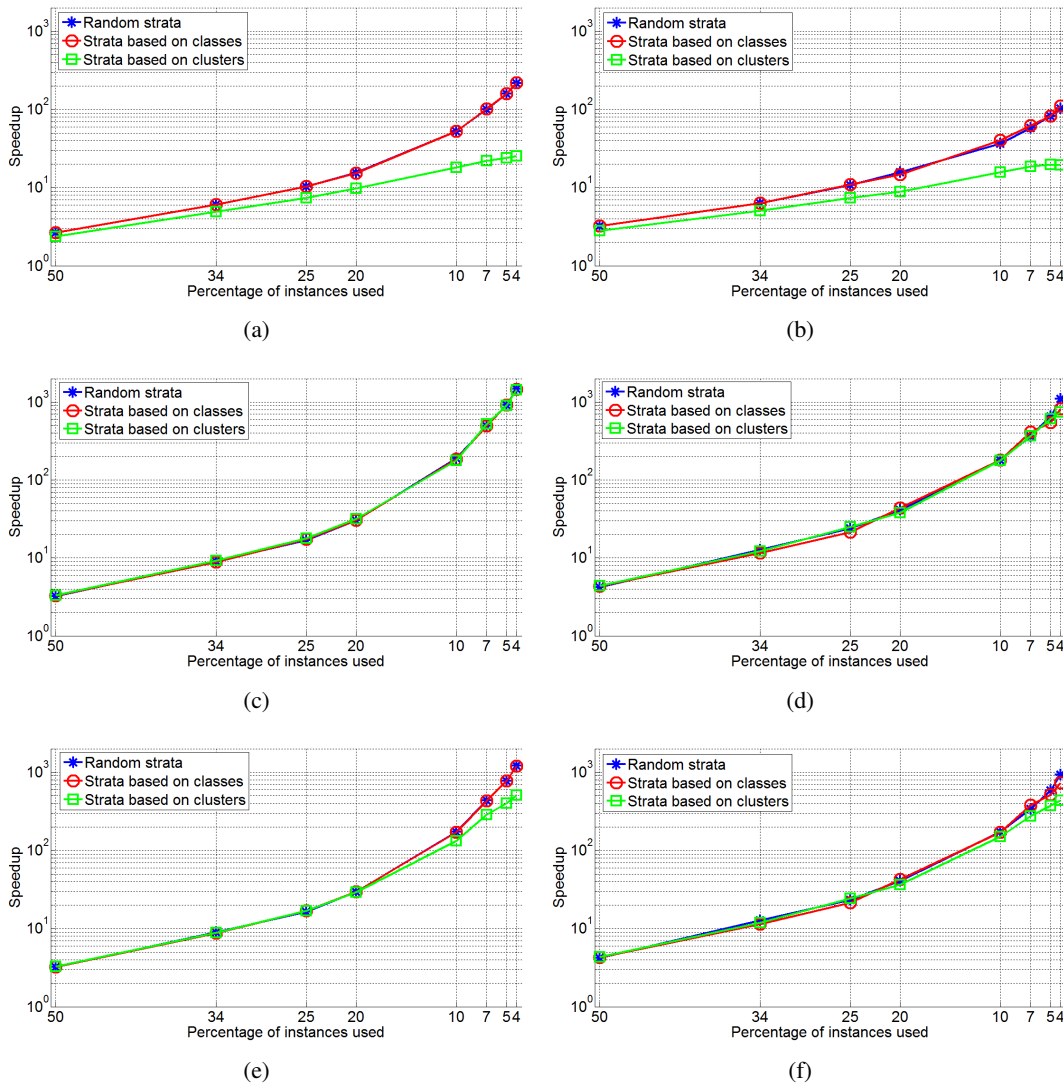


Figure 6.2: Speedup of the three CAOS_{DS} strategies regarding CAOS_{CD} in artificial (a,c,e) and real-world data sets (b,d,f). The first two figures (a,b) are referred to the speedup of the precalculation time. To build the data subsets and to precalculate the distance and nearest neighbors structures. The two following figures (c,d) show the speedup of the clustering time to do the evolutionary process that obtains the Pareto set of solutions. The last two figures (e,f) are related to the speedup of the overall time taking into account both times.

Figure 6.2(e) shows the average speedup of the overall time of the three CAOS_{DS} strategies regarding the overall time of CAOS_{CD} applied to the artificial data sets. This speedup is divided in two partial speedups regarding the precalculation time needed to build the data subsets and to precalculate the distance and nearest neighbors structures (see Figure 6.2(a)), and the second one is referred to the clustering time (see Figure 6.2(c)). The precalculation speedup is the same in the random strata than in the strata based on classes strategies, and it is lower than the time needed in the strategy based on clusters because it needs to cluster the data. On the other hand, the speedup of the clustering time is the same in the three strategies since the strata method does not affect to the clustering process. Analyzing the overall speedup according to the percentage of instances used from the complete data set, it can be observed that using a 50% of the instances, which is the lowest improvement, the speedup is 3, so CAOS_{DS} strategies are three times faster than CAOS_{CD}. Moreover, using a 4% of the instances of the complete data set the speedup is 500 for the strategy based on clusters and 1200 for the other two strategies. Also it can be observed that there are not speedup differences between the strategies until more than a 10% of the instances are used. These analysis showed that CAOS_{DS} strategies applied to the proposed artificial data sets are faster than CAOS_{CD} and do not considerably penalize the accuracy in quantitative terms. Nevertheless, CAOS_{DS} is worse and significantly different, in statistical terms, regarding CAOS_{CD}.

Table 6.5 shows the results of the Holm's test applied to the results obtained with real-world data sets. According to it, when in each data subset is considered a 50% of the instances (2 data subsets) the Friedman's test cannot reject the null hypothesis that all the strategies obtain the same results on average. Thus, the three CAOS_{DS} strategies cannot be considered different than CAOS_{CD} in terms of accuracy. When the 34% and the 25% of the instances are considered, the CAOS_{DS} strategy based on random strata is not significantly different in terms of accuracy regarding CAOS_{CD}. Nevertheless, the accuracy of the other two strategies is worse and significantly different than CAOS_{CD}. When it is used less than the 25% of the instances of the complete data set in each data subset, the three CAOS_{DS} strategies are significantly different in terms of accuracy regarding CAOS_{CD}, but they are not significantly different between them. Figure 6.1(b) shows that there is virtually no accuracy difference between CAOS_{DS} strategies regarding CAOS_{CD}, because the data sets used do not have shapes as complex than the artificial ones used.

Figure 6.2(f) shows that, in terms of speedup, the behavior in the used real-world data sets is similar than in artificial data sets. The maximum speedup is lower because some of the real-world data sets are small and the speedup of using less than a 10% of the instances is not as high than in larger data sets. Also, it can be observed than in the used real-world data sets if it is used a 50% of the instances, it is four times faster than CAOS_{CD} and it obtains the same clustering results.

For more details about the results, consult Table C.1 to Table C.32 in Appendix C, where the accuracy and time of each strategy for each data set are shown.

6.4.3 Discussion

It must be emphasized that the speedup obtained applying this kind of techniques is very high and, consequently, the computational performance of the system is considerably improved. Moreover, assuming that the best strata generation is based on the original classes, the results show that the other two strategies to build the strata are not significantly different in terms of accuracy independently of the kind of data sets tested. In terms of accuracy, the random and the cluster based strategies are as useful as the strategy based on classes but without the requirement of having the original class of each instance. In terms of computational time, the random and classes based strategies have similar speedup regarding $CAOS_{CD}$. Nevertheless, the cluster based strategy has a lower speedup. According to these observations, it seems that the most suitable strategy to build the data subsets in $CAOS_{DS}$ is the random one, because it does not require the original classes of the instances, it is not significantly different in terms of accuracy than the other two strategies and it has a high speedup.

6.5 Summary and Conclusions

The main lack of the techniques based on evolutionary algorithms is their cost in terms of computational time and memory usage when they are applied to a big amount of data since they do an intensive use of computations. This has motivated the necessity of proposing new ways of tackling problems such as stratifying the complete data set in several strata in order to use less data in the algorithm with the aim of reducing the computations while the accuracy is maintained. Thus, the idea of the stratified strategy is to map the initial data set into disjoint strata of equal size and with equal class distribution and allow the system to work with one stratum at the same time. This chapter has analyzed the impact of applying several stratification techniques in a multiobjective evolutionary clustering algorithm using $CAOS$ algorithm. The application of these strategies in $CAOS$ can allow to work with large data sets with a reasonable computational time while the accuracy is not drastically penalized. The approach is based on dividing the data set in some disjoint data subsets (strata) and alternate them in each cycle of the genetic algorithm to increase the generalization of the system and avoiding the bias. Three stratification techniques are analyzed to build the strata (1) according to the original classes of the data set, (2) selecting random instances from the data set, and (3) according to the clusters found applying a fast method called subtractive clustering. The first one is a supervised strategy used to compare the performance of the two unsupervised strategies. These techniques allow to scale-up the system considerably reducing the computational time without drastically penalize the accuracy. They are tested using artificial and real-world data sets in order to analyze their behavior with complex data structures and with some benchmarks from the UCI repository.

The experimentation showed that the speedup of the three strategies is very high, and this considerably improves the computational performance of the system. Moreover, it can be observed that the two unsupervised strategies to build the strata are not significantly different from the strat-

egy based on classes in terms of accuracy. In other words, the random and cluster based strategies can be considered equivalent to the strategy based on classes in terms of accuracy. Furthermore, the strategy based on random strata has a higher speedup than the cluster based strategy, due to the fact that the last one needs to build approximative clusters at the begin of the algorithm and this has a high cost with very large data sets. Thus, the random stratum strategy is the most suitable to scaling-up CAOS. On the other hand, there are statistically significant differences among these three strategies and the approach that use the complete data set, nevertheless the accuracy differences among them are relatively small and they considerably reduce the computational time of the algorithm scaling-up it properly. It is important to highlight that the size of each stratum will affect the performance of the CAOS algorithm. The computational time of CAOS will be decreased as much smaller is the size of the strata, but as much smaller is the size it will be difficult to obtain consistent stratum according to the original data set, affecting to the accuracy of the system.

This chapter sets the basis for further conducting research on multiobjective evolutionary clustering applied to large data sets. This future work will aim at three objectives. First, we will include a validation method to guarantee the consistence of each stratum according to the original data set. Second, we will analyze the consequence of apply stratification methods for scaling-up CAOS with other individual representations. Finally, we will investigate the application of other data mining techniques for large data sets (Bacardit and Llorà, 2009) such as Parallel EAs.

A paper related to this contribution has been submitted to a journal under the title "Scaling-Up Multiobjective Evolutionary Clustering Algorithms using stratification".

Chapter 7

Selection of the Most Suitable Solution

Multiobjective algorithms return a Pareto set of non-dominated solutions where there is not a best solution according to all the optimizing objectives. However, it can be retrieved a solution that can be more suitable to solve a particular problem. To manually find this solution can be a complex task, for this reason automatically methods are needed. There are two main kinds of approaches to automatically retrieve the most suitable solution that are based (1) on the shape of the Pareto set and (2) on specific characteristics of the problem. The first approach returns a solution with a good trade-off among objectives but without taking into account any other characteristic of the solution. The second approach does not takes into account the trade-off between objectives because it only uses the internal characteristics of the solutions. In this chapter, we propose to combine both approaches in order to select the most suitable solution according to the quality of the clusters but obtaining a solution with a good trade-off among objectives. The proposed approach retrieves the best solution according to a clustering validation index from the solutions that are in the region of the Pareto set with better trade-off among objectives. This region is called sweet spot. These approaches are applied to CAOS and they are compared using a wide set of artificial and real-world data sets.

7.1 Motivation

The results in multiobjective algorithms based on Pareto sets (Veldhuizen and Lamont, 2000) are a collection of potential solutions with the characteristic that, for all objectives, a major winner cannot be found. However, in spite of the fact that there is no best solution for all the objectives, it can be found a solution which is the most suitable to solve a particular problem. Therefore, one of the challenges in MC algorithms is the retrieval of the most suitable solution from the Pareto set. This solution can be manually identified by an expert in the domain of the problem, but it results in a subjective method and a non trivial task if there are several solutions in the Pareto set. Thus, automatic methods are needed to help experts and simplify the identification of the most suitable solution.

In multiobjective clustering there are mainly two methods for retrieving the most suitable solu-

tion from the Pareto set: (1) taking into account the shape of the Pareto set (Matake et al., 2007) or (2) taking into account the features related to the shape of the clusters (Handl and Knowles, 2007). The first method tries to identify the knee of the Pareto front for retrieving the solution with best trade-off between objectives, but it does not take into account the clusters shape of the solution, so it can return a solution with human-redeable clusters. The second way retrieves the best solution according to clustering validation indexes (see Section 2.5.4). The main lack of this approach is that the objectives can be unbalanced returning a solution which only optimizes one of the desired objectives. In this chapter we propose a hybrid technique that combine both methods in order to retrieve a solution from the Pareto set with understandable clusters and with a balanced trade-off between objectives. The idea is to find the sweet spot of the Pareto set in order to select the best solution from this region according to a clustering validation index. We call sweet spot the region that includes the solutions that are around the knee of the Pareto front. Thus, the solutions that are outside the sweet spot are not considered as possible solutions due to the fact that they only optimize one of the objectives.

In order to test the benefits of this approach we used CAOS as a base algorithm. To analyze the performance of these strategies we have done a comparison between the results obtained with CAOS retrieving the most suitable solution according to the aforementioned strategies: (1) the shape of the Pareto set, (2) the shape of the clusters, and (3) the shape of the Pareto set and the shape of the clusters simultaneously. To carry out this, the approaches are compared along a set of synthetic data sets (Handl and Knowles, 2007) and real-world ones from the UCI repository (Asuncion and Newman, 2010). The next sections briefly summarize the related work on retrieving solutions from a Pareto set, describe the method proposed and the experimentation done and discuss the results.

7.2 Related Work

One of the most intuitive approaches to identify the most suitable solution is to aggregate all the objectives into some kind of overall objective, but coming up with exact relative objective weights is a daunting task with complicated ramifications (Kasprzak and Lewis, 2001; Messac et al., 2000). Kasprzak proposed a method called collinearity theorem, which goal is to predict the relative objective weighting required to cause any member of the Pareto set to become the optimal solution on the basis of the information contained in the shape of the Pareto set (Kasprzak and Lewis, 2001). Other approaches like ad-hoc methods according to the domain of the problem are used to identify the desirable solution (Liu et al., 2009). For example, if we want to construct a sustainable building minimizing the time and cost of the construction, the most suitable solution can be to use only building materials which respect the environment. Nevertheless, the aforementioned method is not useful when the domain of the problem is not well-known. It is in this situations when it is necessary to use other methods based on general features of the solutions or on the shape of the Pareto set. For example, we can choose a solution from the Pareto set that it is near the knee of

the Pareto front, because it is the region where there should be the solutions with best trade-off between objectives. Branke presented a method based on identifying the knee of the Pareto front by the slopes of the two lines through an individual and its two neighbors. The angle between these slopes can be regarded as an indication of whether the solution is at a knee or not (Branke et al., 2004).

In MC, Handl proposed in MOCK the use of the GAP statistic (Tibshirani et al., 2000) to identify the most suitable solution in the knee of the Pareto front (Handl and Knowles, 2007), this is a solution with a good trade-off between objectives. The main problem of this technique is that the multiobjective algorithm has to be executed several times for finding the solution, therefore it has a high computational cost. Moreover, it does not have into account the shape of the clusters, which is related to the understandability of the clustering result. Mataka proposed in (Mataka et al., 2007) a technique that improves the results and the computational cost of the aforementioned technique. It is based on the angle between the solutions proposed in (Branke et al., 2004) to find a solution in the knee of the Pareto front. However, it also does not take into account the shape of the clusters. Handl also proposed in MOCK the use of some clustering validation indexes that retrieve the solution according to the shape of the clusters instead of taking into account the shape of the Pareto set (Handl and Knowles, 2007). The main problem of this technique is that the validation indexes can return a solution that only properly optimizes one objective, so the solution does not have a good trade-off between the proposed objectives.

7.3 Sweet Spot Selection Technique

Retrieving the most suitable solution from a Pareto set is a complex task. As discussed above, in MC algorithms the approaches based on the shape of the Pareto set can obtain a solution from the knee of it, that is, they are solutions with a good trade-off among objectives, but without any warranty of obtaining a solution with high quality clusters. On the other hand, the methods based on validation indexes can obtain the desirable solution according to the quality of the clusters. Nevertheless, these indexes can be sensitive to outliers and to some specific shape of clusters that are undesirable as a solution. Figures 7.1 and 7.2 show two examples where the clustering validation indexes do not select the most desirable solution, according to the shape of the clusters, due to the existence of outliers. In the first example, the indexes select a solution with a bad trade-off between objectives and it does not generalize, so the solution given does not add any useful knowledge to experts. On the other hand, in the second example it is more difficult to select a solution according to the experts but it can be observed that there is not a consensus between indexes and some of them select the solution with a poorer trade-off between objectives. The limitation produced by outliers can be avoided changing the configuration of some of the index calculations, however they can obtain worse results when there are not outliers. To avoid this problem, we propose a technique based on retrieving the most suitable solution according to the shape of the Pareto set and the shape and quality of the clusters, by retrieving it from the sweet

spot of the Pareto set using clustering validation indexes.

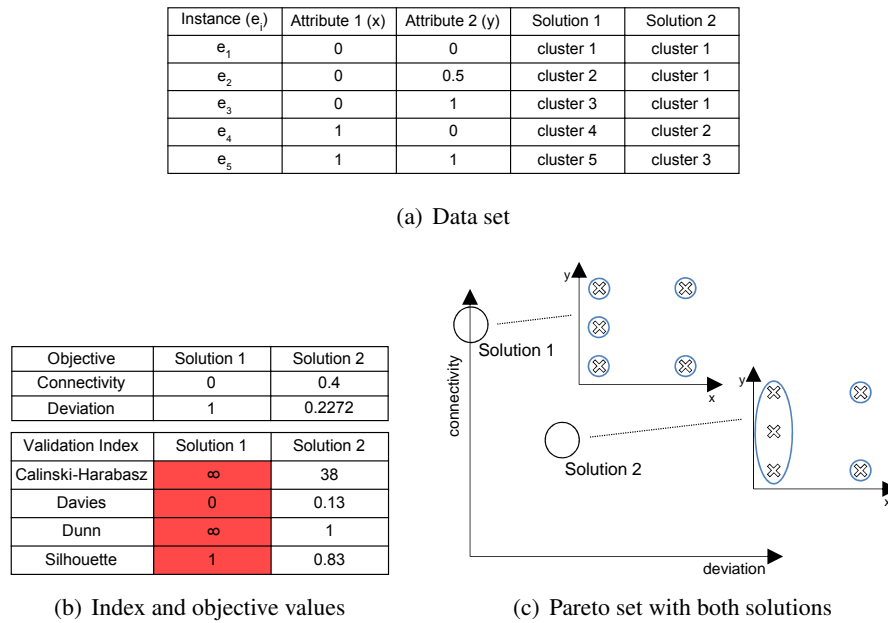


Figure 7.1: Validation indexes results from two non-dominated clustering solutions. The red color indicates the solutions selected by an index but not selected by experts.

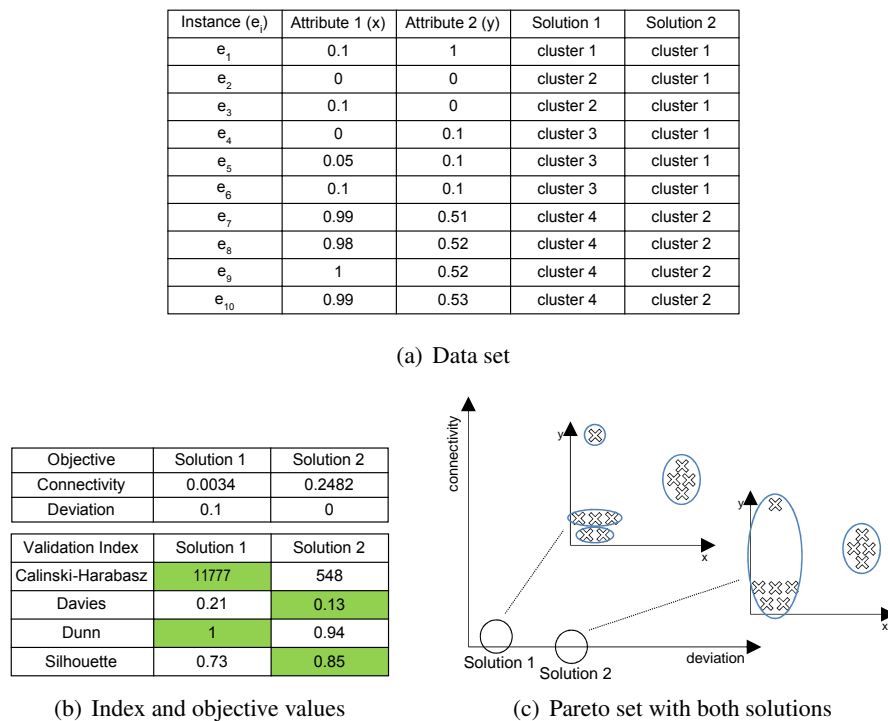


Figure 7.2: Validation indexes results from two non-dominated clustering solutions. The green color indicates the solutions selected by each index.

7.3.1 The Sweet Spot Combined with Clustering Validation Techniques

The objective of the proposed technique is to identify the sweet spot of the Pareto set, which is the region around the knee (see Figure 7.3). This is done by filtering the solutions that are in the boundaries of it, because the solutions included in this regions extremely optimize an objective and they do not properly optimize the other one, usually because they are solutions with very big or small clusters, and they affect to the results of the clustering validation indexes. Taking into account only the solutions contained in the sweet spot of the Pareto set, the indexes can obtain better solutions. The difficulty of this approach is to determine the size of the sweet spot which contains the solutions that will be evaluated through a clustering validation index. This issue is important because if the sweet spot is very small, some valuable solutions from the point of view of shape and quality of clusters can be omitted, but if it is very big, the solutions that are not interesting from the point of view of clustering can be taken into account. For this reason, several sizes of the sweet spot are tested in the experimentation in order to determine the best size for each validation index. To test this technique, it has been applied in the selection of the best solution of CAOS.

7.3.2 Identification of the Sweet Spot in CAOS

Two of the objectives that have been demonstrated more useful to promote the compactness and separation among clusters are the deviation and connectivity objectives (see Section 4.4). In two-objective optimization problems, the Pareto set can be represented in a two-dimensional graph where each axis correspond to each one of the objectives. In this situations, to identify the sweet spot of the Pareto set, two straight lines are drawn from the origin of the axes with an specific angle regarding each axis (α_1 and α_2). The region inside the area comprised between the two straight lines is considered sweet spot (see Figure 7.3). The angle of each line with respect to the axis determines the size of the sweet spot, reducing it when the angles are decreased. If the angles of the two lines are 0 degrees, all the Pareto set is considered as the sweet spot. It is important to

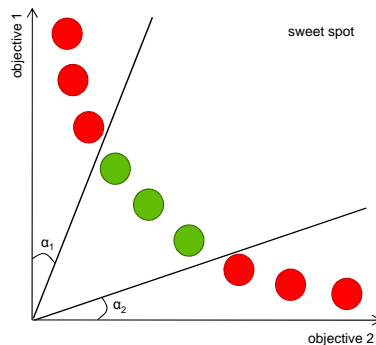


Figure 7.3: Graphical representation of the sweet spot identification. α_1 and α_2 are the angles that determine the size of the sweet spot.

highlight that the two angles cannot be equal or higher than 45 degrees, due to the fact that the area between them cannot comprise any solution of the Pareto set. The result of applying this technique is shown in the following. In other problems where more than two objectives have to be optimized, this specific method needs to be adapted to the dimensions of the problem.

7.4 Experiments, Results and Discussion

This section compares the performance of using the sweet spot selection technique to retrieve the most suitable solution with methods that only use the shape of the Pareto set or the shape of the clusters to retrieve it. Concretely, we compared it with the technique presented by Matake (Matake et al., 2007), which is based on the shape of the Pareto set, and the technique based on clustering validation indexes presented in Section 4.5, which is based on the quality of the clusters. As it has been aforementioned, the technique exposed by Matake is based on the angles of the solutions to identify one of them in the knee of the Pareto front and it has demonstrated that it can obtain interesting results. The comparison was done using the CAOS algorithm with 35 artificial data sets and 35 real-world data sets. The performance is considered in terms of accuracy using the Adjusted Rand index value of the retrieved solution. Next, the experimental methodology, and the results of the comparison are presented and discussed.

7.4.1 Experimental Methodology

This section presents the experimental methodology followed to evaluate the retrieval strategies in CAOS. The analysis enables us to emphasize the benefits and the drawbacks of each one. In the followings, we provide details about (i) the data sets collection chosen for the experimentation, (ii) the CAOS configuration, and (iii) the comparison metrics.

Test Bed. The experimentation assess the algorithms performance using different typologies of artificial and real-world problems (see Table 7.1). Concretely, 35 artificial data sets were selected according to different number of instances (from 900 to 2990), attributes (from 2 to 100) and classes (from 2 to 10). They were built using the tool presented in (Handl and Knowles, 2007). On the other hand, the 35 real-world problems were selected according to different number of instances (from 7494 to 101), attributes (from 3 to 60) and classes (from 2 to 11). These data sets were obtained from the UCI repository (Asuncion and Newman, 2010).

CAOS Configuration. The comparison was done with the Pareto set obtained with the prototype-based representation of CAOS. However, the representation does not considerably affect to the analysis because the goal of this experimentation is to analyze the performance of the proposed retrieval strategies to select the most suitable solution from the Pareto set when it is already build. CAOS was run with 10 different seeds with the following parameters (see Section 4 for notation details): ℓ is 5% of m (the number of data set instances), the maximum size of the initial population

is 100, N_{EP} is 1000, N_{IP} is 50, N_{niches} is 5, the number of generations is 400, $P_c = 0.7$ and P_μ is $1/m$. The minimum and maximum number of clusters for the initial individuals at the prototype-based representation is 2 and 20% of m respectively. The bloat control threshold (sim) value is 0.005 and it is applied after the generation 30, and the clusters are not merged.

Comparison Metrics. The retrieval strategies tested are based on clustering validation indexes (Davies, Dunn, Silhouette and Calinski-Harabasz) and on the strategy based on adjacent angles (Matake et al., 2007). Both strategies are applied to the overall Pareto set and to the sweet spot. Thus, we obtained ten possible solutions from the Pareto set, that is, one solution for each index and for the adjacent angles strategy using the overall Pareto set, and one solution for each index and for the adjacent angles strategy using the sweet spot. In addition to these ten solutions, we have also the best solution from the Pareto set according the Adjusted Rand index, which is the best one according to the classes assigned to each instance. Finally, to identify the suitable sweet spot size we tested several values to the α_1 and the α_2 angles. We used the same values for both angles that range between 1 and 44 degrees, concretely, the values used are: 1, 2, 5, 10, 15, 20, 25, 30, 35, 40,

Data set	nI	nA	nC	Data set	nI	nA	nC
100d-10c	2198	100	10	appendicitis	106	7	2
100d-4c	1218	100	4	balance	625	4	3
10d-10c	2122	10	10	biopsia	1027	24	2
10d-4c	1092	10	4	bpa	345	6	2
2d-10c	2990	2	10	contraceptives	1473	9	3
2d-4c	1261	2	4	crx	690	15	2
curves1	1000	2	2	dermatology	366	35	6
curves2	1000	2	2	echocardiogram	132	12	2
dartboard1	1000	2	4	ecoli	336	8	8
dartboard2	1000	2	4	glass	214	9	6
donut1	1000	2	2	haberman	306	3	2
donut2	1000	2	2	heart-statlog	270	13	2
donut3	999	2	3	hepatitis	155	19	2
donutcurves	1000	2	4	housevotes	435	16	2
long1	1000	2	2	ionosphere	351	34	2
long2	1000	2	2	iris	150	4	3
long3	1000	2	2	liver-disorders	345	6	2
longsquare	900	2	6	mammographic	961	5	2
sizes1	1000	2	4	pendigits	7494	17	10
sizes2	1000	2	4	pim	768	8	2
sizes3	1000	2	4	segment	2310	19	7
sizes4	1000	2	4	sonar	208	60	2
sizes5	1000	2	4	tae	151	5	3
smile1	1000	2	4	thyroids	215	5	2
smile2	1000	2	4	transfusion	748	4	2
smile3	1000	2	4	vehicle	846	18	4
spiral	1000	2	2	vertebral	310	6	3
spiralsquare	1500	2	6	vowel	990	13	11
square1	1000	2	4	waveform	5000	40	3
square2	1000	2	4	wdbc	569	30	2
square3	1000	2	4	wine	178	13	3
square4	1000	2	4	wisconsin	699	9	2
square5	1000	2	4	wpbc	198	33	2
triangle1	1000	2	4	yeast	1484	9	10
triangle2	1000	2	4	zoo	101	16	7

Table 7.1: Summary of the characteristics of the 35 artificial data sets (left block) and real-world data sets (right block) used. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).

43 and 44.

Finally, the recommendations pointed out by Demšar (Demšar, 2006) and the Nemenyi test exposed in Section 5.4.1 have been used to statistically compare the results of each approach.

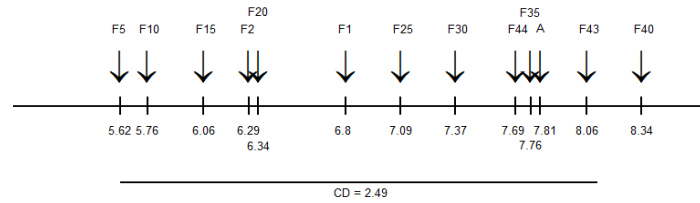
7.4.2 Comparison of Results

The performance of the aforementioned strategies were empirically tested with artificial data sets and with real-world data sets. Also, each solution is quantified using the Adjusted Rand index in order to evaluate them according to the original classes of the problems

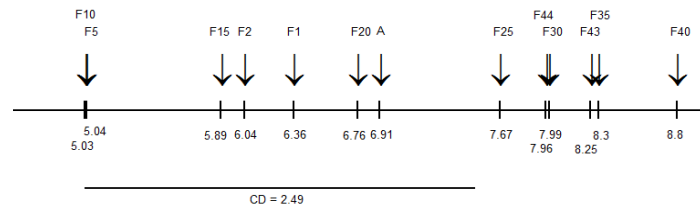
Before comparing the solutions obtained with the Pareto set and with the sweet spot, it is necessary to identify the best size of the sweet spot, which can depend on the data set and on the solutions obtained in the Pareto set. However, we want to identify the size that obtains the best results. For this, we tested several sizes for each one of the clustering validation indexes and for the adjacent angles strategy. Next, we statistically compare the strategies based on the overall Pareto set and on the sweet spot, using the best size for each index and for the adjacent angles strategy.

In order to have a large sample of data sets for selecting the best sweet spot size for each retrieval strategy with more precision, we used all the aforementioned data sets without dividing them into artificial and real-world data sets. The idea is to obtain an approximative size of the sweet spot for each strategy to be used independently of the kind of data set where they are applied. Figure 7.4(a) to Figure 7.4(d) show the rank of the sweet spot sizes analyzed for each index, and Figure 7.4(e) shows the same information for the adjacent angles strategy. The rank is analyzed in terms of accuracy and the best one is the lowest one. It can be observed that the best results for each index are obtained when the sweet spot angles are between 5 and 10 degrees (represented by $F5$ and $F10$ respectively). On the other hand, it is observed that the adjacent angles strategy needs a small sweet spot to obtain good results, concretely, sweet spot angles higher than 20 degrees are necessary (represented by $F20$, $F25$, $F30$, $F35$, $F40$, $F43$ and $F44$). Finally, it is also observed that the solution retrieved from the overall Pareto set (represented by A) are far of the best solution obtained with the suitable sweet spot, but in the majority of the strategies, the solutions using the sweet spot and using the Pareto set are not statistically different. However, it is important to highlight that to apply the validation indexes to the sweet spot is faster than to apply it to all the solutions of the Pareto set and the solutions retrieved from the sweet spot have better trade-off between objectives, which is the main goal of multiobjective clustering.

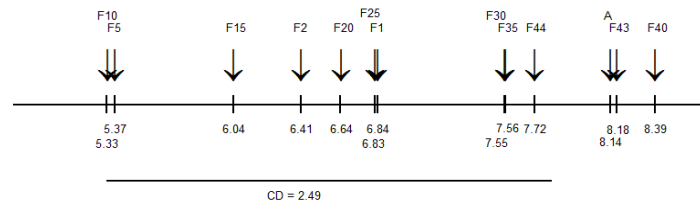
In the analysis of the performance among all the strategies using the overall Pareto set and the sweet spot, we used the artificial data sets and the real-world ones separately because we want to analyze the performance of them in different situations. Figure 7.5 shows the results for both kind of data sets, in it the strategy used to retrieve the most suitable solution is indicated by Dv , Dn , Sl , CH and Ag for the Davies, Dunn, Silhouette, Calinski-Harabasz indexes and the adjacent angles strategy respectively. Also, the symbols of each strategy are preceded by an F when the sweet spot is used and by an A when the overall Pareto set is used. Also, the supervised solution retrieved with the Adjusted Rand index, which is only used taking into account the overall data set,



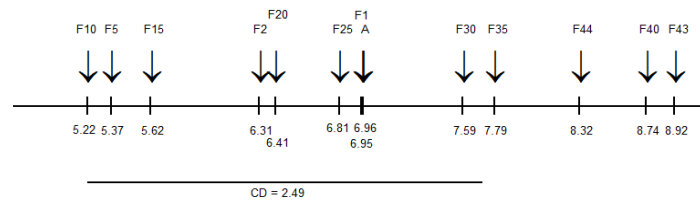
(a) Davies Index



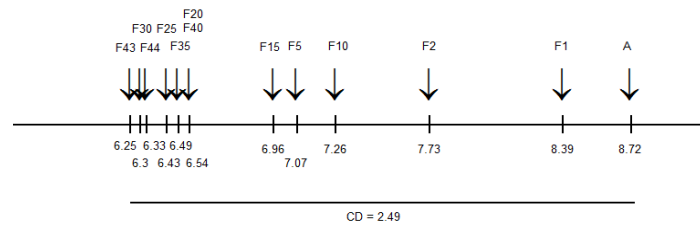
(b) Dunn Index



(c) Silhouette Index



(d) Calinski-Harabasz



(e) Adjacent Angles Strategy

Figure 7.4: Accuracy rank with Nemenyi test of the most suitable CAOS solution obtained with the indexes (a) Davies, (b) Dunn, (c) Silhouette, (d) Calinski-Harabasz, and (e) the adjacent angles strategy with all the Pareto set (A) and with using different angles to define the sweet spot size: 1 (F1), 2 (F2), 5 (F5), 10 (F10), 15 (F15), 20 (F25), 30 (F30), 35 (F35), 40 (F40), 43 (F43) and 44 (F44) degrees. CD indicates the value of the critical distance, representing with a line the area that is not significantly different with respect to the best ranked method.

is represented by Rd . Figure 7.5(a) shows that using artificial data sets there is not virtually any difference among the solutions obtained using the sweet spot and the overall Pareto set. Also, it can be observed that Dunn and Silhouette indexes, using the sweet spot or using the overall Pareto set, obtain solutions that are not significantly different to the best solution Rd obtained using the original classes of the data set. Moreover, with this kind of data sets the adjacent angles strategy and the Calinski-Harabasz index obtain bad results in comparison with the other strategies. It is important to highlight that the artificial data sets used do not have outliers and, in the majority of the data sets, the classes are well-separated and there is no overlapping among them. Thus, it is easier for the indexes to retrieve a suitable solution without leave out any solution, and for this there is not strong differences between the solutions retrieved with the overall Pareto set and with the sweet spot.

Figure 7.5(b) shows the solutions retrieved using the real-world data sets. It can be observed that all the solutions retrieved from the sweet spot with the clustering validation indexes are better than the solutions retrieved by the same indexes using the overall Pareto set. Nevertheless, there are virtually not differences between the results obtained with the adjacent angles strategy using the sweet spot and the overall Pareto set. Moreover, it is important to highlight that there are no statistically differences between the supervised solution Rd and the solution obtained from the sweet spot with the Calinski-Harabasz index. Thus, it seems that this index is very robust with

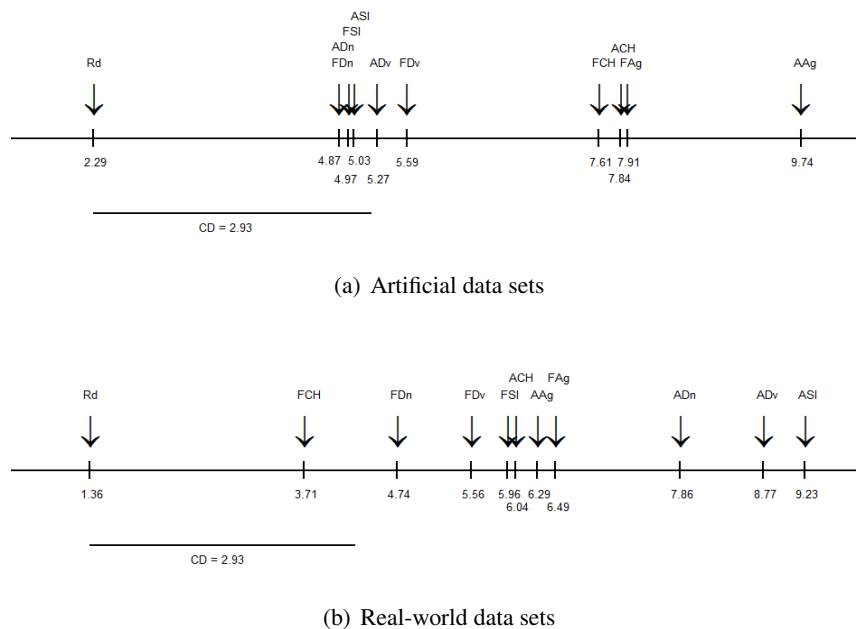


Figure 7.5: Accuracy rank with Nemenyi test of the most suitable CAOS solution obtained with the indexes Davies (Dv), Dunn (Dn), Silhouette (SI), Calinski-Harabasz (CH), and the adjacent angles strategy (Ag) with all the Pareto set (A) and with the best sweet spot size for each strategy (F). Also, the best solution retrieved according the Adjusted Rand index is shown (Rd). The results are obtained using (a) artificial data sets and (b) real-world data sets. CD indicates the value of the critical distance, representing with a line the area that is not significantly different with respect to the best ranked method.

real-world problems due to the fact that it is less affected by noise than the other indexes.

For more details about the results, consult Table D.1 and Table D.2 in Appendix D, where the accuracy of each strategy for each data set is shown.

7.4.3 Discussion

The experiments show that with artificial data sets the solutions retrieved by clustering validation indexes from the sweet spot are virtually equivalent to the solutions obtained from the overall Pareto set. On the other hand, the solutions retrieved by the indexes from the sweet spot with the real-world data sets are better than the solutions obtained from the overall Pareto set. Thus, independently of the typology of data sets used (compact classes, scattered classes, noisy data sets...) the sweet spot solutions improve or maintains the accuracy of the retrieved solutions, so it is an useful retrieval approach. Moreover, it is worth noting that there are some indexes that are not significantly different from the supervised approach that uses the original classes of the data set for retrieving the solution. These indexes are Dunn and Silhouette with the artificial data sets, and Calinski-Harabasz with the real-world data sets. However, even Dunn index applied to real-world problems is statistically different from the supervised solution, it obtains also good results. Thus, if we have to choose only one index, Dunn index can be useful for any kind of data set. After all, each one of the indexes evaluates the quality of the clusters using different features of them and we consider that, in CAOS, the solutions retrieved by each one of the indexes can be offered as final solution to the clustering experts. It must be emphasized that the solutions retrieved with the adjacent angles strategy, independently if they use the sweet spot or the overall Pareto set, are worse than the solutions retrieved from the sweet spot using the indexes. This has sense due to the fact that these solutions do not take into account the quality of the clusters and they only try to retrieve a solution near the knee of the Pareto front.

Finally, regarding the size of the sweet spot, the experiments show that with the clustering validation indexes is not necessary to filter a lot of solutions to improve the accuracy of the method. Concretely, using sweet spot angles of 5 or 10 degrees the results are remarkably better than using the overall the Pareto set. Thus, only the solutions that optimize solely one of the objectives are not taken into account, so potentially good solutions are not discarded. On the other hand, the adjacent angles strategy can discard a big amount of solutions without decreasing the accuracy of the method. This effect is explained as follows: due that the goal of this strategy is to obtain a solution near the knee of the Pareto front, all the solutions that are not around it can be discarded.

7.5 Summary and Conclusions

The solution returned by multiobjective algorithms is a Pareto set where there is no solution better than the others for each objective. Even there is not a winner solution according to all the optimizing objectives, the most suitable solution to solve a specific problem can be retrieved. This has motivated the necessity of proposing methods for automatically retrieving the most suitable

solution. These methods usually select this solution according to (1) the shape of the Pareto set, which correspond to the value of the objectives to optimize, and to (2) the quality of the solutions conforming to specific characteristics of the problem. In clustering problems, the main lack of the first method is that it retrieves a solution without taking into account the shape and quality of the clusters and it can return a solution with a good trade-off between objectives but with incomprehensible clusters. On the other hand, the second method retrieves a solution according to the quality and shape of clusters using clustering validation indexes but it does not take into account the value of the objectives, so it can return a solution with an inadequate trade-off between them and this is not the aim of MC. For these reasons, we propose the combination of both methods to obtain a new one that selects a solution according to a clustering validation index from the region of the Pareto set where are placed all the solutions with a good trade-off between objectives (sweet spot). The main problem of this approach is to define the size of the sweet spot, because potentially good solutions can be discarded if it is very small and not useful solutions can distort the results of the validation indexes if it is very large. In the experimentation we tested several sweet spot sizes to find the most appropriate one and several clustering validation indexes in order to properly analyze the performance of the method. Moreover, the performance of the proposed method was analyzed in comparison with a method that uses a clustering validation index to retrieve the solution and with another one that tries to retrieve the solution in the knee of the Pareto front according to the use of adjacent angles. These tests have been carried out using a wide set of artificial and real-world data.

The results showed that it is not necessary to define a small sweet spot size, so it is not necessary to discard a lot of solutions. Thus, in order to filter the solutions that can distort the value of the validation indexes it is only necessary to discard the solutions that extremely maximize just one of the objectives. These solutions have usually very large or very small clusters that become non-human readable clustering solutions. Moreover, the performance of the sweet spot method combined with clustering validation indexes is better than the performance of the solutions retrieved according to the overall Pareto set and to the adjacent angles strategy, independently of the kind of data sets used. Furthermore, the proposed method can obtain solutions that are not significantly different to the solutions retrieved by a supervised method, so they work as well as a method that uses the classes of the problem to retrieve the best solution. Finally, it is important to highlight that the sweet spot method, combined with quantitative measures that evaluates the quality of the solutions according to specific characteristics of the problem, can be applied in any kind of multiobjective optimization algorithm.

As future work we can analyze the effect of using other strategies to filter the solutions that are not near the knee of the Pareto front, the use of other retrieval strategies based on the quality of the clusters, and the application of the sweet spot technique, or a technique with a similar aim, to Pareto sets with more than two objectives.

Part III

Practical Application of CAOS in Real-World Problems

Chapter 8

Decision Support System for the Analysis of Vulnerabilities in Telematic Networks

Information system security must battle regularly with new threats that jeopardize the protection of those systems. Security tests have to be run periodically not only to identify vulnerabilities but also to control information systems, network devices, services and communications. Vulnerability assessments gather large amounts of data to be further analyzed by security experts, who recently have started using data analysis techniques to extract useful knowledge from these data. With the aim of assisting this process, this chapter uses CAOS to cluster information of security tests. The process enables the clustering of the tested devices with similar vulnerabilities to detect hidden patterns, rogue or risky devices. Two different types of metrics have been selected to guide the discovery process in order to get the best clustering solution: general-purpose and specific-domain objectives. The results of both approaches are compared with the state-of-the-art single-objective clustering techniques to corroborate the benefits of the clustering results to security analysts.

8.1 Motivation

The increase of the dependency of organizations on information and communication technologies, together with the need of securing companies systems in a world where new threats are risks appear daily, has unleashed the demand for new and effective security techniques. Therefore, maintaining a proper level of security is a key challenge in current organizations, even when they have the most advanced technology and trained professionals (Nedjah et al., 2007). Consequently periodic security tests –project-oriented risk assessments of information systems and networks through the application of professional analysis on a security scan– are necessary to assure that security does not degrade below an acceptable risk level. One of the most important analysis included in these tests is the vulnerability assessment, i.e., the process followed to identify and quantify vulnerabili-

ties. Both follow a two-step process: test everything possible and generate a concise report.

The cost and the time involved in a security test may limit its depth, so an automation is essential, specially in the analysis of test results. A complete analysis must also coordinate diverse sources of information to support an intelligent response (Dawkins and Hale, 2004). So security applications demand intelligence to detect malicious data, unauthorized traffic or vulnerabilities (DeLooze, 2004). Machine learning can be applied to process the results of vulnerability assessments. The use of unsupervised learning for discovering hidden patterns through the identification of groups of tested devices with similar vulnerabilities has already been presented in *Analia*. This is the analysis module of the framework *Consensus*, a computer-aided system that automates the processes associated to security tests for information systems and networks (Corral, 2009).

Analia helps security analysts in the task of extracting conclusions from data of security tests. This is due to the integration of different clustering approaches and clustering validation techniques. However, two independent steps are needed before extracting conclusions: analysts have to select (1) the clustering approach and (2) the validity index to return the most appropriate solution. Therefore, the best clustering solution depends on the selected validity index, as each index may evaluate different goals. Moreover, the clustering and the index goals may not be aligned. Analysts also ask for a system where configuration parameters not related to their domain, like the clustering technique or the validation index, are provided automatically.

This chapter presents a new contribution in the domain of information system security. The drawbacks of the clustering process in *Analia* are tackled with *CAOS*. This approach groups tested devices with similar vulnerabilities guided by different goals, as a multiobjective technique allows. So security analysts will obtain the best clustering solution considering different criteria simultaneously. Thus analysts will not need to configure any parameter regarding clustering or validity indexes and will be able to focus only on the obtained clustering results, which is their actual concern. Two different configurations for *CAOS* are studied, depending on the objectives used to evaluate the system: general-purpose and domain-specific objectives. The experimental analysis presented in this chapter demonstrates the improvement of clustering results when using the domain-specific objectives with *CAOS*. Also the process of extracting conclusions from results is simplified, as analysts are now able to extract the best clustering solution and the most adapted to the domain-specific objectives in a single step.

The remainder of this chapter is organized as follows. First, the related work on machine learning in the security domain is introduced. Next, the objectives used to guide *CAOS* are proposed. Finally, the clustering process in *Analia* and the experimentation results are presented.

8.2 Related Work

The increasing frequency of incidents of security breaches in information systems and the ever-increasing reliance of organizations on information technologies involve a constant monitoring of the existing security level for early detection of any negative variation in that control measure. The

last IBM Trend and Risk Report provided an account of vulnerability disclosures in the last few years. It stated that the annual vulnerability disclosure rate appears to be fluctuating between 6-7 thousand new disclosures each year. The most prevalent primary consequence of vulnerability exploitation continues to be gain access (Services, 2009). A study carried out by IDC states that external threats often overshadow the importance of protecting against internal risks (Burke, 2009). Therefore, periodic security tests are needed to check that security is maintained. *Consensus* is a security testing framework created to aid security managers in these regular tasks (Corral, 2009). However, these periodical tests generate large volumes of data that have to be processed to give an alarm signal in case new vulnerabilities or security holes are detected.

The huge amount of data produced by security tests has promoted the use of enhanced techniques to recognize malicious behavior patterns or unauthorized changes in information systems or networks (DeLooze, 2004). These domains are usually defined by sets of unlabeled examples, and experts aim at extracting novel and useful information about the network behavior that helps them detect vulnerabilities, among others. In this context, clustering appears as an appealing approach that allows grouping network devices with similar security vulnerabilities, thence, identifying potential threats to the network.

Several clustering techniques have been applied to the network security domain thus far. For example, k -means (Hartigan and Wong, 1979) has been used to group similar alarm records (Bloedorn et al., 2006) and to detect network intrusions (Leung and Leckie, 2005). Self-organizing maps (SOM) (Kohonen, 2000) have been employed to detect computer attacks (DeLooze, 2004), network intrusions (Depren et al., 2004), and anomalous traffic (Ramadas et al., 2003). Despite the success of these applications, all these clustering techniques guide the discovery process with a single criterion. For example, k -means minimizes the total within-cluster variance and tends to find spherical clusters (Hartigan and Wong, 1979). Our case is different, as we are interested in obtaining clusterings that satisfy different criteria. For this purpose, several authors have proposed to run different clustering techniques to obtain different structures, and then, involve the network expert into the process in order to manually select the best structure according to certain predetermined validation methods.

To automatize this process, we propose the use of CAOS, which guides the clustering process with different objectives. Some MC approaches have been successfully applied to important real-world problems such as intrusion detection (Anchor et al., 2002), formation of cluster-based sensing networks in wireless sensor networks (Yang et al., 2007), and creation of security profiles (Gupta et al., 2006).

8.3 Definition of Specific Optimization Objectives According to the Domain

The quality of each individual, which is a possible clustering solution, is evaluated with a set of objective functions. These are validation indexes, that is, methods to validate how good a clustering

solution is. In our experiments, we used two different types of objectives: (1) objectives typically employed by general-purpose clustering techniques and (2) objectives designed as validation indexes by network security experts, which reflect how clusters should look like. The general-purpose indexes considered here are the *Deviation* of clusters and the *Connectivity* between clusters, which measure the compactness and the connectedness of clusters, respectively (see Section 4.4). The domain-specific indexes employed here are *Intracohesion* (Ita) and *Intercohesion* (Ite) factors (Corral et al., 2006), which also measure the compactness and connectedness, respectively, but now in terms of security vulnerabilities. In other words, the *Intracohesion* factor evaluates the similarity between the elements of each cluster according to the vulnerabilities they share. Values close to 1 indicate that the elements of each cluster are similar each other, and close to 0 indicate that they are different. On the other hand, the *Intercohesion* factor evaluates the similarity between the clusters according to the vulnerabilities of each one. Values close to 0 indicate that the clusters are different and close to 1 indicate that they are similar. Thus, we are interested in maximize the *Intracohesion* factor and minimize the *Intercohesion* factor. These validation indexes are described in Equations 8.1 and 8.2. Where m is the number of examples in the training data set; C is the clustering obtained; n is the number of clusters; C_i is the cluster i ; $|C_i|$ is the number of elements in C_i ; v_i is the centroid of C_i ; $d(x, y)$ is the Euclidean distance between x and y elements; $nn(x, i)$ returns the i th nearest element of x according to $d(x, y)$; ℓ is the amount of nearest elements taken into account; $CommonVulnerabilities(C_i, C_j)$ is the number of the common vulnerabilities between all the elements of C_i and C_j ; and $TotalVulnerabilities(C_i)$ is the number of vulnerabilities that all the elements of C_i have in common. The *Intracohesion* factor *Intercohesion* factor

$$Ita(C) = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{|C_i|^2 - |C_i|} \sum_{x \in C_i} \sum_{\substack{y \in C_i \\ y \neq x}} S(x, y) \right), \text{ where} \quad (8.1)$$

$$S(x, y) = \frac{CommonVulnerabilities(x, y)}{TotalVulnerabilities(x)}$$

$$Ite(C) = \frac{1}{n^2 - n} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n D(C_i, C_j), \text{ where} \quad (8.2)$$

$$D(C_i, C_j) = \frac{CommonVulnerabilities(C_i, C_j)}{TotalVulnerabilities(C_i)}$$

After training, CAOS returns a population of non-dominated solutions which optimize the particular objectives chosen for the problem. Then, the system needs to select one of the solutions and, finally, return it to the security expert.

CAOS uses different validation techniques to recover one of the solutions of the Pareto set. In our experiments, we considered the following ones: the Davies-Bouldin index (Davies and

(Bouldin, 1979), the Dunn index (Dunn, 1974), the Silhouette index (Rousseeuw, 1987), the *Intracohesion* factor, the *Intercohesion* factor, and the *Intra-Inter* (II) factor (Corral, 2009; Corral et al., 2006). The first three indexes have been explained in Section 2.5, the other ones are explained in Equations 8.1 to 8.3. The Intra-Inter factor returns the best solution equally weighting the two objectives used, and to obtain the best solution the factor has to be maximized.

$$II(C) = Ita(C) - Ite(C) \quad (8.3)$$

As we are interested in analyzing the difference in the behavior of the system when it is guided by general-purpose validation indexes and when it is guided by the domain-specific indexes, in the remainder of this chapter we take the two approaches to face the network security problem. For the sake of clarity, the system guided by the two general-purpose validation indexes (i.e., *Connectivity* and *Deviation*) will be referred to as CAOS. On the other hand, the system guided by the two domain-specific validation indexes (i.e., *Intracohesion* and *Intercohesion*) will be addressed as CAOSII.

8.4 Consensus and Analia to Analyze Security Tests

This section describes *Consensus* and its analysis module *Analía*. The benefits of CAOS when processing the results of security tests are described.

8.4.1 Description

Consensus is an information security system that automates processes related to security assessments in order to minimize the time needed to perform a security test (Corral, 2009). *Consensus* gathers data from different network devices, not only computers but also routers, firewalls and Intrusion Detection Systems (IDS). General information, port and vulnerability scanning data, operating system (OS) fingerprinting, routing and filtering rules, IDS response, answer to malicious code, weak passwords reporting, and response to denial of service attacks can be stored for each tested device.

The great amount of data for every device and the different number and type of attributes complicates a manual traffic pattern finding. *Analía* is the data analysis module of *Consensus* and includes unsupervised learning. It finds resemblances within tested devices, and clustering helps security analysts in the extraction of conclusions from data. The best clustering results are selected by applying cluster validity indexes. Explanations of clustering results are provided to security analysts so as to give a more comprehensive response (Corral, 2009). The integration of *Analía* in *Consensus* is shown in Figure 8.1.

8.4.2 Single-Objective Clustering

The manual analysis of data gathered after a security test can become an arduous labor that may even mask relevant information due to the large amount of data. Our previous works validated the incorporation of several unsupervised learning techniques into *Analia*. They were based on single-objective clustering approaches, like *k*-means, *x*-means and SOM (Corral, 2009; Corral et al., 2006).

Analia clusters the data set composed by the network devices that have already been audited. Security experts look forward to a system that groups devices with similar vulnerabilities. However, the knowledge representation of the tested devices is based on the port scanning data and the OS fingerprinting. Features directly related to vulnerabilities have not been used to cluster, as their data formatting is not suitable for input parameters of the aforementioned clustering algorithms. On the other hand, data obtained from these two processes is what a security expert would first analyze to find heterogeneities in tested devices. In fact, devices with similar open ports and OS may share the same security vulnerabilities, so handling this information is also critical.

Two validation indexes (*Intracohesion* and *Intercohesion*) were formulated directly related to the domain, in order to evaluate the clustering results according to existing security vulnerabilities, as detailed in Section 8.3. Thus, the resulting clusters with similar open ports and OS can be appraised according to their common vulnerabilities.

The data set stored in *Analia* is a real unsupervised domain, where the number of the existing classes is not known a priori. Two networks never operate in the same way, so each security assessment produces a new different domain. This is a drawback for several clustering approaches, which require the user to specify the number of clusters of the solution. So many executions need to be run to select the best solution according to a certain validity index. The indexes included in *Analia* are the following: Davies-Bouldin, Dunn, Silhouette, *Intracohesion*, *Intercohesion* and *Intra-Inter*.

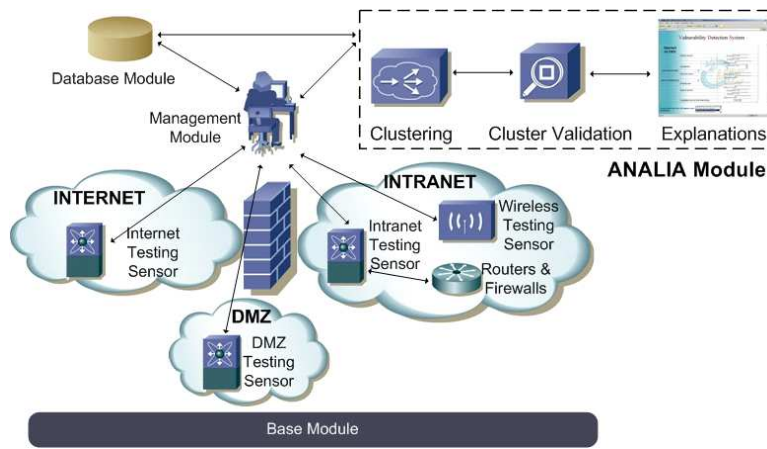


Figure 8.1: Architecture of *Consensus* system and *Analia* data analysis module.

The process to achieve the best clustering solution in *Analia* is summarized as follows:

1. Select the single-objective clustering approach.
2. Run different executions varying input parameters.
3. Calculate validation indexes for each execution.
4. Select the validation index as decision criterion to get the best results.

Security analysts do not usually care about the selected clustering approach and index criteria, but about the best clustering solution. Analysts should easily get the best partition without any previous knowledge about clustering or validity indexes, knowledge not usually related to their expertise area. However, this two-step process of selecting a clustering technique and, afterwards, applying validation indexes may slow down the whole process. Moreover, the choice of any of the aforesaid clustering approaches with the knowledge representation explained before, together with typical validity indexes, does not ensure that the best solution will cluster devices with similar vulnerabilities. This is because neither the clustering nor the validation are considering the information related to security vulnerabilities.

As an improvement, when using *Intracohesion* or *Intercohesion* as validity indexes, the clustering solution with the most compact clusters or with the most separated clusters in terms of vulnerabilities will be shown, respectively. But even this improvement has a drawback: the goal of the clustering approaches was not to optimize these criteria, but the centroids or the director maps. So the best solutions for *k*-means, *x*-means or SOM do not have to coincide with the best solutions in terms of security vulnerabilities, due to the fact that the goals are aimed at slightly different directions. However, the modification of the optimization function of the different clustering approaches would complicate the system. In addition, only one of the cohesion indexes could be included, as they are single-objective algorithms.

Next section presents the contribution to improve these drawbacks. If validation indexes are considered the initial goals of clustering, the clustering solutions will optimize the selected indexes, thus reducing the whole process and achieving better clustering solutions more adapted to the desired goals.

8.4.3 Multiobjective Clustering

Single-objective clustering techniques have shown excellent results in many domains (Corral, 2009; Kaski et al., 2003; Kuo et al., 2005). But they globally optimize a single objective function. This restriction may narrow the search space to find the best solution when more than one goal is pursued. Thus we propose a multiobjective clustering approach to process the data of security tests: CAOS.

We have integrated CAOS in *Analia* in two different ways. The first approach, named CAOS-DC, optimizes two general-purpose objectives: connectivity and deviation. The second approach, named CAOS-II, optimizes the two domain-specific objectives: *Intracohesion* and *Intercohesion*.

In both cases we use complementary objectives so that CAOS-DC and CAOS-II can evolve a set of solutions that move through the tradeoff of both objectives. The goals of both implementations are also aligned. Deviation is related to cluster compactness, so it evaluates clusters with similar open ports and OS. *Intracohesion* evaluates the cluster compactness in terms of the vulnerabilities common in a cluster. On the other hand, connectivity is based on the cluster connectedness. *Intercohesion* evaluates the connection between clusters, considering the vulnerabilities common to different clusters.

The knowledge representation cannot be changed, as it was a design decision of the security experts in *Analia*. Hence, CAOS-DC considers only OS fingerprinting and port scanning data to calculate the deviation and the connectivity. This implies that the best CAOS-DC solutions will group devices with similar OS and ports and, simultaneously, will separate devices with different OS and ports, when optimizing both goals. This methodology is more powerful than single-objective clustering, which only considers one of the goals. But CAOS-DC still does not consider data related to vulnerabilities, so the best CAOS-DC solutions do not imply the best solutions in terms of vulnerabilities, when applying cohesion indexes to select the best solution of the Pareto.

CAOS-II considers not only OS fingerprinting and port scanning but also vulnerability data. This guides the system in the search of the best clustering solutions, as the objective functions are directly related to the *Intracohesion* and the *Intercohesion* factors. Thus, the final Pareto includes the best clustering solutions that maximize the *Intracohesion* and minimize the *Intercohesion*. So, the solutions that group devices with similar vulnerabilities and separate devices with different vulnerabilities are obtained in a single step. These solutions also help security analysts. When detecting a device with a very critical vulnerability, the devices in the same cluster will be also vulnerable and the same solution will have to be applied to them. Moreover, when scanning a group of similar devices, if one of them is in a different cluster, it will mean that the security level of that device has been compromised. Analysts will detect it without having to analyze all the tested devices.

To conclude, when including CAOS-II in *Analia*, the process to achieve the best partition is summarized in a single step: run CAOS-II. It obtains a set of non-dominated solutions that optimize both *Intracohesion* and *Intercohesion* in a single run. Note that, as the validation indexes have been included in the search process, there is no need to calculate those indexes afterwards. As a result, the best executions with the best number of clusters will be automatically obtained. In the next section, we present the experimental analysis with the different single-objective and multiobjective clustering approaches.

8.5 Experiments, Results and Discussion

In this section, we first explain the aim of the experiments and the methodology employed, and then, we present the results.

8.5.1 Experimental Methodology

The aim of the experiments was to answer several questions that move from the accuracy and the efficiency of the different methods to the quality of the clustering solutions from the security experts' viewpoint. Thus, we first analyzed whether CAOS-II resulted in more interesting clustering solutions than original CAOS-DC. Then, we extended the comparison of CAOS-II by including k -means, x -means, and SOM. With these two comparisons we aimed at demonstrating the robustness and competitiveness of CAOS-II with respect to original CAOS-DC and to some of the most used and influential single-objective clustering techniques. Thereafter, we focused on what network security experts would expect from a clustering solution. As a consequence, we first compared the best results obtained by the clustering solutions considering the index *Intra-Inter*, since this metric was specifically designed for this domain. Finally, we moved the analysis to the qualitative side and examined the information that clustering provided to security experts. In the followings, we provide details about (i) the data extraction, (ii) the clustering configuration, and (iii) the comparison metrics.

Data Extraction. The data set used has been extracted from the *Consensus* data, which it contains information of the port scanning, the operating system and the vulnerability tests of the data network. This data set has been built from real security tests executed over 90 devices of La Salle–Ramon Llull University network, including public and internal servers, alumni laboratories and staff computers.

Clustering Configuration. To address these questions, we ran the five systems with different configurations: k -means with $k = \{3, 4, 5, 6, 7, 8, 9, 10\}$; x -means with $min_k = 3$ and $max_k = \{4, 5, 6, 7, 8, 9, 10\}$; SOM with map sizes of $\{3 \times 3, 4 \times 4, 5 \times 5, 6 \times 6\}$; CAOS-DC and CAOS-II with the following parameters: ℓ is 20% of m , $|EP|$ is 1000, $|IP|$ is 50, min_i and max_i are 5% and 10% of m respectively, the number of generations is 400 and P_C is 0.7. We also tested all the validation indexes to recover the best solution, that is, *Intracohesion* (Ita), *Intercohesion* (Ite), *Intra-Inter* (II), Davies-Bouldin (DB), Dunn (Dn), and Silhouette (Sil). All the experiments were repeated with 10 different random seeds.

Comparison Metrics. The results were statistically compared following the recommendations pointed out by Demšar (Demšar, 2006). We first applied multiple-comparison statistical procedures to test the null hypothesis that all the learning algorithms performed equivalently on average. Specifically, we used the Friedman's test (Friedman, 1940). If the Friedman's test rejected the null hypothesis, we performed pairwise comparisons by means of the Holm's step-down procedure (Holm, 1979). Following this procedure, we could distinguish pairs of learners that performed significantly differently.

	CDC _{Ita}	CDC _{Ite}	CDC _{DB}	CDC _{Dn}	CDC _{Sil}	CDC _{II}	CII _{Ita}	CII _{Ite}	CII _{DB}	CII _{Dn}	CII _{Sil}	CII _{II}
CDC _{Ita}												
CDC _{Ite}	+											
CDC _{DB}	+	+										
CDC _{Dn}	+	+	+									
CDC _{Sil}	+	+	+	+								
CDC _{II}	+	+	+	+	+							
CII _{Ita}	+	+	+	+	+	+						
CII _{Ite}	⊕	⊕	⊕	⊕	⊕	+	⊕					
CII _{DB}	⊕	+	+	⊕	⊕	+	+	+				
CII _{Dn}	⊕	+	+	⊕	⊕	+	⊕	+	+			
CII _{Sil}	⊕	+	+	⊕	⊕	+	+	+	+	+		
CII _{II}	⊕	⊕	⊕	⊕	⊕	+	⊕	+	+	+	+	

Table 8.1: Pairwise comparisons of the *Intra-Inter* value of the clustering solutions returned by CAOS-DC (CDC) and CAOS-II (CII) by means of a Holm’s procedure. We have run CAOS-DC and CAOS-II considering all the indexes to recover the best solution, i.e., *Intracohesion* (Ita), *Intercohesion* (Ite), Davies-Bouldin (DB), Dunn (Dn), Silhouette (Sil), *Intra-Inter* (II). The symbol \oplus shows that the method in the row obtained results that were significantly higher than those obtained with the method in the column. Similarly, the symbol $+$ denote a non-significant higher results.

8.5.2 Comparison of Optimization Objectives

Our first concern was to analyze whether guiding the evolutionary search according to the domain-specific indexes resulted in better results than those obtained when guiding the search with general-purpose indexes. For this purpose, we compared the results obtained with CAOS-DC and CAOS-II when the clustering solutions were evaluated with the *Intra-Inter* validation index.

The Friedman’s test rejected the null hypothesis that all learners performed the same, on average, at $\alpha = 0.001$. Thus, we ran the Holm’s step-down procedure (see Table 8.1). The symbols \oplus and \ominus show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column at $\alpha = 0.05$. Similarly, the symbols $+$ and $-$ denote a non-significant higher/lower results. Note that higher values of *Intra-Inter* correspond to better clustering solutions.

The results provided in Table 8.1 highlight the advantages of guiding the search of CAOS with the *Intracohesion* and the *Intercohesion* indexes. That is, all the configurations of CAOS-II outperformed the configurations of CAOS-DC. In addition, in several cases CAOS-II was significantly better than CAOS-DC. This behavior was expected since CAOS-II searches for solutions that optimize the goals specifically designed by network experts for this particular domain. Nevertheless, it is worth highlighting the flexibility of the multiobjective approach, which enables the direct inclusion of these indexes.

8.5.3 Performance of CAOS with Specific Objectives Regarding Single-Objective Clustering Methods

Having demonstrated the competitiveness of CAOS-II with respect to CAOS-DC and the benefits provided by the domain-specific validity indexes, we now extend the study by including some of the most used clustering techniques. For this purpose, we compared the results of CAOS-II with those reached by *k*-means, *x*-means, and SOM. As we wanted to analyze whether CAOS-II

provided competitive results regardless of the validation index used to assess the performance of the clustering solutions, in what follows we compare the results when the general-purpose *Davies-Bouldin* index and the domain-specific *Intra-Inter* indexes are used to evaluate the solution quality.

The multicomparison Friedman's test rejected the hypothesis that CAOS-II, *k*-means, *x*-means, and SOM performed the same, on average, at $\alpha = 0.001$, regardless of whether the quality of the solution was evaluated with the *Davies-Bouldin* or the *Intra-Inter* index. Therefore, we applied the Holm's step-down procedure. Tables 8.2 and 8.3 show the summarized results of Holm's step-down procedure when the best clustering solutions are evaluated with the *Davies-Bouldin* and the *Intra-Inter* validation indexes respectively. The symbols \oplus and \ominus indicate that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column, at $\alpha = 0.05$. Similarly, the symbols + and - denote a non-significant higher/lower results. Note that while good clustering solutions should minimize the *Davies-Bouldin* index, they should maximize the *Intra-Inter* index.

Table 8.2 shows the comparisons of the DB value among CAOS-II with all the retrieval indexes and the different single-objective methods analyzed. It can be seen that all the configurations of SOM, *k*-means and *x*-means obtained worse DB values than any of the configurations of CAOS-II. In addition, in most of the cases CAOS-II provided significantly better results according to the Holm's procedure. For example, among others, all CAOS-II configurations significantly outperformed *k*-means for $k = \{4, 5, 6\}$ and SOM with a map size of 3×3 . Thus, these results encourage the use of CAOS-II even when the results are evaluated with a general-purpose index instead of with the domain-specific index designed for the network domain.

Table 8.3 illustrates the results of the comparisons among CAOS-II with all the retrieval indexes and the single-objective approaches. In this case, the final clustering techniques were evaluated with the *Intra-Inter* index.

The results presented in Table 8.3 are similar to those reported in Table 8.2: CAOS-II always achieved better results than those obtained by the single-objective techniques like SOM, *k*-means and *x*-means. Moreover, in several cases CAOS-II significantly outperformed the other techniques. Furthermore, it is worth highlighting that CAOS-II with *Intra-Inter* as a retrieval index was always significantly better than any of the single-objective methods except for SOM with map sizes $\{5 \times 5, 6 \times 6\}$, *k*-means with $k = \{3, 9\}$, and *x*-means with $max_k = 4$. Therefore, after analyzing the different tables and results, CAOS-II appeared as the best approach to be applied in the given network security domain.

The results shown in Tables 8.2 and 8.3 support the initial hypotheses of the chapter. That is, multiobjective clustering algorithms are worthwhile in domains where different goals, usually opposed, are pursued. When the goals can be customized to the domain, results become more profitable. CAOS-II is a multiobjective clustering approach that includes domain-specific goals, so the search of the best solutions is guided to find the right clustering maps. Furthermore, if the selection function to retrieve the best solution of the Pareto set is properly designed, clustering results will overcome.

	CII _{I_{ta}}	CII _{I_{te}}	CII _{DB}	CII _{D_n}	CII _{S_{il}}	CII _{I_l}
SOM3	⊕	⊕	⊕	⊕	⊕	⊕
SOM4	+	+	⊕	+	+	+
SOM5	⊕	⊕	+	⊕	⊕	⊕
SOM6	+	+	+	+	+	+
KM3	+	+	+	+	+	+
KM4	⊕	⊕	⊕	⊕	⊕	⊕
KM5	⊕	⊕	⊕	⊕	⊕	⊕
KM6	⊕	⊕	⊕	⊕	⊕	⊕
KM7	⊕	⊕	⊕	⊕	⊕	⊕
KM8	+	+	⊕	⊕	⊕	⊕
KM9	+	+	⊕	⊕	⊕	+
KM10	+	+	⊕	⊕	⊕	+
XM4	+	⊕	⊕	⊕	⊕	⊕
XM5	+	+	⊕	⊕	⊕	⊕
XM6	+	+	+	+	+	+
XM7	+	+	+	+	+	+
XM8	+	+	+	+	+	+
XM9	+	+	+	+	+	+
XM10	+	+	⊕	+	+	+

Table 8.2: Pairwise comparisons of the *Davies-Bouldin* value of the clustering solutions returned by CAOS-II (CII) with the different validation indexes to recover the best solution and SOM with 3×3 (SOM3), SOM with 4×4 (SOM4), SOM with 5×5 (SOM5), SOM with 6×6 (SOM6), *k*-means with *k* ranging from 3 (KM3) to 10 (KM10), and *x*-means with $min_k = 3$ and max_k ranging from 4 (XM4) to 10 (XM10). The symbol \oplus shows that the method in the row obtained results that were significantly higher than those obtained with the method in the column. Similarly, the symbol + denote a non-significant higher results.

	CII _{I_{ta}}	CII _{I_{te}}	CII _{DB}	CII _{D_n}	CII _{S_{il}}	CII _{I_l}
SOM3	⊖	⊖	⊖	⊖	⊖	⊖
SOM4	-	⊖	-	-	-	⊖
SOM5	-	-	-	-	-	-
SOM6	-	-	-	-	-	-
KM3	-	-	-	-	-	-
KM4	⊖	⊖	⊖	⊖	⊖	⊖
KM5	-	⊖	⊖	⊖	⊖	⊖
KM6	⊖	⊖	⊖	⊖	⊖	⊖
KM7	-	⊖	⊖	⊖	⊖	⊖
KM8	-	⊖	⊖	⊖	-	⊖
KM9	-	-	-	-	-	-
KM10	-	⊖	-	-	-	⊖
XM4	⊖	-	⊖	⊖	⊖	⊖
XM5	-	-	⊖	⊖	⊖	⊖
XM6	-	-	⊖	-	-	⊖
XM7	-	-	-	⊖	-	⊖
XM8	-	-	-	⊖	-	⊖
XM9	-	-	-	-	-	-
XM10	-	-	-	⊖	-	⊖

Table 8.3: Pairwise comparisons of the Intra-Inter value of the clustering solutions returned by CAOS-II (CII) with the different validation indexes to recover the best solution and SOM with 3×3 (SOM3), SOM with 4×4 (SOM4), SOM with 5×5 (SOM5), SOM with 6×6 (SOM6), *k*-means with *k* ranging from 3 (KM3) to 10 (KM10), and *x*-means with $min_k = 3$ and max_k ranging from 4 (XM4) to 10 (XM10). The symbol \ominus shows that the method in the row obtained results that were significantly lower than those obtained with the method in the column. Similarly, the symbol - denote a non-significant lower results.

	CAOS-II	SOM	KM	XM
CAOS-II				
SOM	⊖			
KM	⊖	–		
XM	⊖	–	–	

Table 8.4: Pairwise comparison of the Intra-Inter value of the clustering solutions returned by CAOS-II with recuperation of the best solution based on Intra-Inter, SOM 6×6 , k -means with $k = 9$ (KM), and x -means with $max_k = 10$ (XM). The symbol \ominus shows that the method in the row obtained results that were significantly lower than those obtained with the method in the column. Similarly, the symbol $-$ denote a non-significant lower results.

8.5.4 Moving on to the Expert Side

Having quantitatively and statistically analyzed the results of CAOS-II, and having shown its excellent results, we now move on to the security expert side. We hence compared the CAOS-II results with the best results of any of the single-objective approaches. The quality of all the results was measured with the *Intra-Inter* index, since it is what analysts are concerned about.

The Friedman’s test rejected the hypothesis that all learners performed the same, on average, at $\alpha=0.001$. Table 8.4 shows the results of the pairwise comparisons according to the Holm’s step-down procedure, at $\alpha=0.05$, when the best clustering solutions are evaluated with the *Intra-Inter* index. The symbols \oplus and \ominus indicate that the method in the row obtained results significantly higher/lower than those obtained with the method in the column, at $\alpha = 0.05$. Similarly, the symbols $+$ and $-$ denote non-significant higher/lower results.

Table 8.4 shows that single-objective approaches obtained significantly lower results than CAOS-II when comparing the best clustering solutions evaluated by the *Intra-Inter* index. High values of this index are preferred, as good solutions should maximize *Intracohesion* and minimize *Intercohesion* simultaneously. So CAOS-II offers the significantly best solution, grouping devices with similar vulnerabilities and separating devices with different vulnerabilities. These results have been achieved even when the knowledge representation did not consider information about vulnerabilities, but about OS fingerprinting and port scanning. Finally, CAOS-II output was selected to be shown to analysts.

The best clustering map calculated by CAOS-II considering the highest *Intra-Inter* index consisted of 6 clusters. The clusters with a single element allowed the location of outlier devices with specific vulnerabilities. The solution created another group of devices with different OS and services, but with many vulnerabilities in common. The clustering map also repositioned some critical information systems into specific clusters due to their distinctive vulnerabilities, like the e-mail server or the radius server, which were located into separated clusters. Finally, the map detected a group of devices that were part of a multiprocessor cluster with very specific features. The distribution of devices in groups according to their vulnerabilities allowed security analysts to label each cluster and classify them depending on their risk level, in order to prioritize the analysis of the most vulnerable clusters.

8.5.5 Discussion

The empirical results have shown that both approaches benefit from the use of multiple objectives. The clustering solutions achieved are better than the results obtained with some of the most used and competent single-objective clustering algorithms, like k -means, x -means and SOM. Moreover, the use of a multiobjective approach with objectives created specifically for this security domain improved the results and allowed security analysts to obtain a clustering solution optimized in terms of vulnerabilities detected. The statistical analysis of the results from the different clustering approaches, not only single but also multiobjective, has shown the benefits of the contributions of this chapter in the information system security field.

In addition, the use of CAOS in *Analia* allows security analysts to reduce the efforts spent in the clustering phase. They do not need to have previous knowledge of cluster validation indexes in order to obtain the best solution of a set of clustering executions. With CAOS-DC, the pursued goals that guide the search towards the best solutions are aligned with the validity indexes. Thus, the obtained clustering solutions comply with validity requirements. Then, a subsequent phase where validity indexes are applied is not necessary. Once the clustering results are shown to security analysts, their task starts analyzing the characteristics of the obtained clusters. Clusters will group devices with similar OS and open ports, and will prioritize security vulnerabilities when applying CAOS-II. Once the clusters are calculated, explanations of clustering results are included to give a more comprehensive response.

8.6 Summary and Conclusions

This chapter has presented the incorporation of CAOS to analyze data from vulnerability assessments in a network security domain. The security analysts that use *Analia* need to have previous knowledge of cluster validation indexes in order to obtain the best solution of a set of clustering executions, this is an important lack in order to obtain a suitable solution. CAOS applied to *Analia* reduces the efforts that the security analysts spent in the clustering phase. The application of CAOS in *Analia* has been presented in two different approaches: (1) CAOS-DC, which includes two generic-purpose indexes as objectives (deviation and connectivity), and (2) CAOS-II, which includes two domain-specific indexes as objectives (*Intracohesion* and *Intercohesion*). The results show that both approaches obtain better solutions than the obtained ones by original *Analia*. However, CAOS-II approach obtains more understandable solutions for experts because it groups devices with similar OS and open ports prioritizing security vulnerabilities.

In general, the overall work has highlighted the importance of applying machine learning techniques to the network security domain in order to accurately and efficiently identify vulnerabilities in our networks. The experimental analysis has also evidenced the importance of the validation indexes selected to guide the search and to recover the best solution. Therefore, in further work, we will focus on the incorporation of new metrics as objectives of CAOS to be optimized related to network features and the improvement of the process to select the best clustering solution with

an easy security analysts interaction.

Two papers related to this contribution have been published in the framework of the MID-CBR project (Corral et al., 2009b; Corral et al., 2011).

Chapter 9

Validation of the Acquisition of Competences in University Degrees

Nowadays the educational methodologies are being reconsidered to allow the successful achievement of the professional skills through the acquisition of competences. This chapter is focused on assessing if a subject offers the competences that are defined in the degree program through the evaluation methods that it uses. Concretely, we propose a procedure based on MC to identify lacks or inconsistencies in the subjects of a degree. The procedure has been successfully tested in the Computer Engineering degree of our institution. The analysis of results has allowed experts to identify some improvements to be made in this degree.

9.1 Motivation

The new needs of the society are promoting the improvement of the educational methodologies in universities. In Europe, these adaptations are a consequence of the European Higher Education Area ([Declaration, 1999](#)), where the educational models have been redefined and now they focus on providing to the student specific competences as a basis for learning outcomes around the world ([Crawley, 2001](#); [Gonzalez and Wagenaar, 2003](#); [Gonzalez and Wagenaar, 2005](#)). A competence is a set of skills, knowledge and attitudes that an individual must possess in order to be capable of satisfactorily perform a specified job ([Hager and Gonczi, 1994](#)). It is important to highlight that the typology competences is closer related to the methodology used for teaching subjects and evaluating students. For example, a technical subject taught through oral presentations will develop competences like 'oral communication' ([Golobardes and Madrazo, 2009b](#); [Golobardes and Madrazo, 2009a](#)). Our institution obtained the European acknowledgement through the European Credit Transfer System Label¹ (ECTS) and the Diploma Supplement Label² (DS) in 2005. Both certificates were the result of specifying the competences that students are expected to acquire in

¹ECTS reference: 29467-IC-1-2005-1-ES-ERASMUS_ECTSL

²DS reference: 29467-IC-1-2005-1-ES-ERASMUS_ECTSDS

the offered degrees. In this process it was necessary to define the competences provided by each subject.

Nowadays, La Salle is assessing if the specified competences in the ECTS Label are really offered in the subjects. This chapter focuses on analyzing the data from the university information system, where teachers have specified the competences, the educational methodology and the evaluation methods for each one of the subjects. This chapter describes a procedure to help the educational experts to determine how to set the new degrees to the requirements of the EHEA. Concretely, we study the cross-relationships between (1) the subjects and the competences, (2) the subjects and the evaluation methods, and (3) the competences and the evaluation methods. The relationships are identified using CAOS with the definition of two new domain-specific objectives in order to adapt better the search of clusters to the educational problem.

Next sections describe the related work, the experimentation done and discuss the obtained results.

9.2 Related Work

There are many studies in the national and in the European context for identifying which are the most important competences of engineers and, consequently, adapting the educational methodologies for promoting the most suitable competences.

Reflex report (Allen and van der Velden, 2007) was elaborated in a project focused on the European context called *The flexible professional in the knowledge society. New demands on higher education in Europe*. The main goal of the study was to identify the most important competences that engineers need to work. 40000 students were surveyed in 14 European countries. Moreover, there are two interesting national studies focused on telecommunication studies. The one performed by the Official College of Telecommunications Engineers (COIT) (Colegio Oficial de Ingenieros de Telecomunicación y Asociación de Empresas de Electrónica, 2007) and the one performed by the Catalan Association of Telecommunication Engineers (Llorens, 2008).

On the other hand, this is not the first time in which Data Mining techniques have been used to help experts in the adaptation of the educational methodologies. In previous works, we applied clustering techniques to identify subject typologies to apply common methodologies to adapt them to the EHEA (Garcia-Piquer et al., 2009b; Golobardes and Madrazo, 2009a) and to validate the competences that the subjects of a specific university degree should offer (Garcia-Piquer et al., 2009a; Garcia-Piquer et al., 2010a).

9.3 Definition of Specific Optimization Objectives According to the Domain

The quality of each individual is evaluated with two domain-specific objectives, which are validation indexes. These indexes are the *Intracohesion* (Ita) and *Intercohesion* (Ite) aforementioned in

Section 8.3 but adapted now to the educational domain. These validation indexes are described in Equations 9.1 and 9.2. Where m is the number of examples in the training data set; C is the clustering obtained; n is the number of clusters; C_i is the cluster i ; $|C_i|$ is the number of elements in C_i ; v_i is the centroid of C_i ; $d(x, y)$ is the Euclidean distance between x and y elements; $nn(x, i)$ returns the i th nearest element of x according to $d(x, y)$; ℓ is the amount of nearest elements taken into account; $CommonAttributes(C_i, C_j)$ is the number of the common educational attributes between all the elements of C_i and C_j ; and $TotalAttributes(C_i)$ is the number of educational attributes that all the elements of C_i have in common.

$$Ita(C) = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{|C_i|^2 - |C_i|} \sum_{x \in C_i} \sum_{\substack{y \in C_i \\ y \neq x}} S(x, y) \right), \text{ where} \quad (9.1)$$

$$S(x, y) = \frac{CommonAttributes(x, y)}{TotalAttributes(x)}$$

$$Ite(C) = \frac{1}{n^2 - n} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n D(C_i, C_j), \text{ where} \quad (9.2)$$

$$D(C_i, C_j) = \frac{CommonAttributes(C_i, C_j)}{TotalAttributes(C_i)} \quad (9.3)$$

To recover the most suitable solution from the Pareto set, the *Intra-Inter* (II) index is used in order to equally weight the two objectives.

9.4 Experiments, Results and Discussion

The objective of this chapter is to found relationships between the subjects and the competences, the subjects and the evaluation methods, and the competences and the evaluation methods; to allow educational experts to extract conclusions that help them to adapt and to monitor the degrees.

9.4.1 Experimental Methodology

This section presents the experimental methodology followed to help educational experts to improve university degrees. In the followings, we provide details about (i) the data extraction from the degree definition, and (ii) the CAOS configuration.

Data Extraction. The competences and evaluation methods of each subject were extracted from the university information system. These information was filled by the teachers and by the degree committee. Specifically, the degree committee selected as relevant 32 competences based on the

Tuning project (Gonzalez and Wagenaar, 2003; Gonzalez and Wagenaar, 2005) (see Table 9.1) and defined 21 evaluation methods (Golobardes and Madrazo, 2009a), which are listed as follows:

- Class exercises
- Class participation
- Classwork
- Computer exams
- Home exercises
- Homework
- Individual works
- Laboratory assemblies
- Laboratory participation
- Laboratory reports
- Oral exams
- Oral presentations
- Others
- Practical computer assignments
- Practical works
- Problems
- Project correction
- Project jury
- Test exams
- Work done with a team
- Written exams

Finally, the evaluation methods that should be used to evaluate each competence were defined by the people responsible of each degree. Thus, we obtained three data sets, which correspond to the three relationships to obtain:

1. Each instance of this data set is a subject described by the evaluation methods used to evaluate it.
2. In this data set, each instance corresponds to a subject described by the competences that they provide.
3. Each one of the instances of this data set corresponds to a competence described by the evaluation methods that should be used to evaluate it.

Each one of the features that describe an instance is a binary value (1 or 0). This value indicates if the instance use the feature (1) or it do not use it (0). It can be observed that the indexes defined in Section 9.3 were designed specifically to this kind of data, because they take into account if the instances provide the defined features.

CAOS Configuration. CAOS has been configured with the following parameters: ℓ is 20% of m , $|EP|$ is 1000, $|IP|$ is 50, min_i and max_i are 5% and 10% of m respectively, the number of generations is 400 and P_C is 0.7.

9.4.2 Comparison of Results

Figure 9.1 shows the results after cluster the subjects according the evaluation methods, and Figure 9.2 shows the subject clusters obtained according to the competences that the subjects provide. The subjects showed in both figures have been selected as example from the Computer Engineering degree of our university. Table 9.1 shows the clusters assigned to each competence according to

the evaluation methods that they use. Table 9.2 indicates the most relevant evaluation methods that each cluster of competences uses.

Competence	Cluster	Competence	Cluster
Ability to communicate with experts in other fields	C1	Environmental sensibility	C1
Ability to work autonomously	C3	Ethical commitment	C1
Ability to work in an interdisciplinary team	C4	Grounding in basic knowledge of the profession	C2
Ability to work in an international context	C1	Information management skills (ability to retrieve and analyze information from different sources)	C5
Appreciation of diversity and multiculturalism	C2	Initiative and entrepreneurial spirit	C1
Basic general knowledge in the educational context	C3	Interpersonal skills	C1
Capacity for adapting to new situations	C1	Knowledge in any educational specialization	C1
Capacity for analysis and synthesis	C4	Knowledge of a second language	C4
Capacity for applying knowledge in practice	C1	Leadership	C4
Capacity for generating new ideas (creativity)	C3	Oral and written communication in your native language	C4
Capacity for organization and planning	C4	Problem solving	C3
Capacity to learn	C3	Project design and management	C1
Concern for quality and long life learning	C1	Research skills	C2
Critical and self-critical abilities	C5	Teamwork	C4
Decision-making	C5	Understanding of cultures and customs of other countries	C1
Elementary computing skills	C3	Will to succeed	C1

Table 9.1: Competences defined in La Salle–Ramon Llull University degrees. For each competence it is indicated the cluster assigned in the clustering result.

Cluster	Evaluation Method
C1	Work done with a team Practical computer assignments
C2	Written exams Work done with a team Homework
C3	Examen oral Oral presentations Work done with a team
C4	Work done with a team Homework
C5	Homework Oral presentations

Table 9.2: Most representative evaluation methods from each one of the competence clusters found. The first column indicates the competence cluster and the second one indicates the name of the evaluation methods.

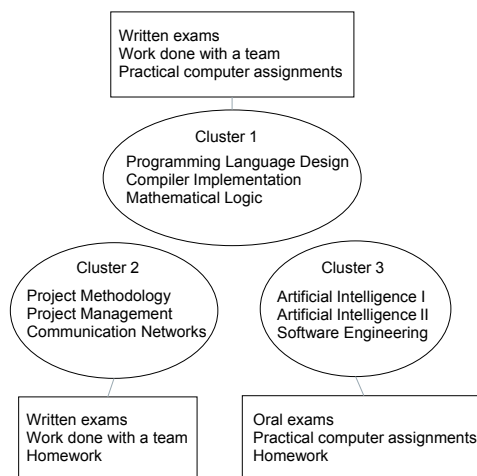


Figure 9.1: Clusters of some example subjects from the Computer Engineering degree according to the evaluation methods that they use. Only the most relevant evaluation methods are shown.

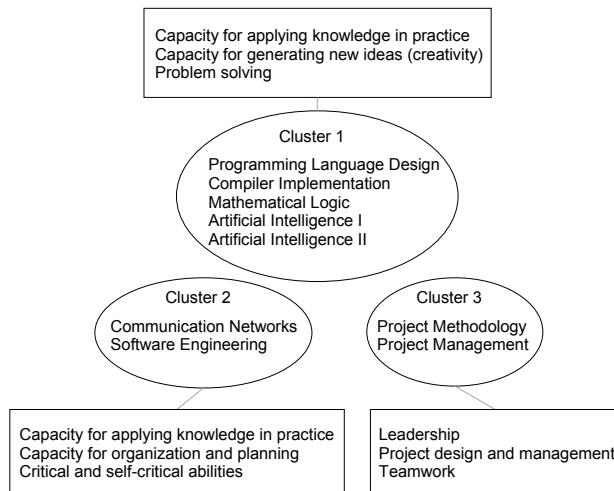


Figure 9.2: Clusters of some example subjects from the Computer Engineering degree according to the competences that they provide. Only the most relevant competences are shown.

9.4.3 Discussion

It can be observed that both subject clustering contain different groups, therefore, it does not exist a direct relationship between the subjects evaluation and the competences that they provide. This means that subjects are not currently being evaluated in a way to assess all the competences that should provide, because the subjects are being evaluated in a similar way to the one used before to define the competences. This is an important lack due to the fact that the acquisition of the competences by the students is not guaranteed. Next step consists in modifying the evaluation methods used in the subjects, and it would be aligned with the results of Tables 9.1 and 9.2. Thus, each competence should be evaluated by some of the evaluation methods of the corresponding

cluster. An example of this is that the competence “Capacity for applying knowledge in practice” (belonging to cluster 1) should be evaluated at least with the evaluation methods “Work done with a team” and “Practical computer assignments”; which are the methods that have to be used to evaluate all the competences of cluster 1.

It is important to highlight that this process is realized to identify any lacks in the subjects of a degree, but it has to be flexible to the specific characteristics of each subject. That is, the goal of this process is to be a support to the educational experts to adapt the degrees to the new requirements. Given these relationships between subjects, competences and evaluation methods, experts have made an adaptation of the subjects of the Computer Engineering degree that they consider correct.

9.5 Summary and Conclusions

The study of the relationship between competences, subjects and evaluation methods is important to the educational experts to adapt the degrees to the requirements of the EHEA. This chapter has presented a procedure to study these relationships based on multiobjective clustering. The use of CAOS with the definition of domain-specific objectives has been positively appreciated by educational experts, and it has allowed them to apply some corrective actions to the Computer Engineering degree.

It is important to highlight that although our institution provides an educational model where competences are shared between subjects, this process is also applicable in those models of education where competences are not shared between subjects, or where there are subjects that do not offer any general competence, apart from the specific competences of the subjects. To check the competences is a hard process that cannot be reduced to this chapter, so as further work we have planned to improve the proposed protocol to make this task easier to the educational experts.

One paper related to this contribution have been published in the framework of the *Guidelines for Competence Assessment in Engineering and Architecture* project (Garcia-Piquer et al., 2010b).

Chapter 10

Multiobjective Knowledge Organization in Case-Based Reasoning Systems

Real-world problems usually present a huge volume of imprecise data. These types of problems may challenge case-based reasoning systems because the knowledge extracted from data is used to identify analogies and solve new problems. Many authors have focused on organizing case memory in patterns to minimize the computational burden and deal with uncertainty. The organization is usually determined by a single criterion, but in some problems a single criterion can be insufficient to find accurate clusters. This chapter describes an approach to organize the case memory in patterns based on multiple criteria. This new approach uses the searching capabilities of CAOS to build a Pareto set of solutions, where each one is a possible organization based on the relevance of objectives. The system shows promising capabilities when it is compared with a successful system based on Self-Organizing Maps. Due to the data set geometry influences the clustering building process, results are analyzed taking into account it. For this reason, some complexity measures are used to categorize data sets according to their topology.

10.1 Motivation

Case-based reasoning (CBR) (Aamodt and Plaza, 1994) systems solve new problems through an analogy-based procedure related to experiences represented by a set of *cases* stored in a *case memory*. The way in which CBR works can be summarized in the following steps: (1) it retrieves the most similar cases from the case memory, (2) it adapts them to propose a new solution, (3) it checks whether this solution is valid, and finally, (4) it stores the relevant knowledge used to solve the problem. As the case memory feeds this process, its size and its organization play an important role in the CBR performance in terms of computational time and accuracy. Traditionally, CBR systems follow a flat organization where cases are stored as a list and the retrieve phase has to sequentially compare the new case with all the previous ones. Therefore, this linear access penalizes the retrieval time in a large case memory and the absence of patterns hinders the detection

of noisy cases. For this reason, structured case memory organization has become a promising research line where clustering techniques (Herrera et al., 2010) are a commonly used approach to speed up the retrieval time and to improve accuracy. The main idea is to promote the selection of the most potentially useful cases based on grouping cases according to their similarities, and next extracting prototypes for characterizing each group (Czarnowski, 2011). Thus, the computational time is reduced because only the cases that belong to the groups where the prototypes fit well with the input case will be used (Fornells et al., 2007b). On the other hand, the suitability of prototypes for representing clusters is crucial for selecting the useful cases and avoiding a reduction of the accuracy. For this reason, clustering methods should promote the similarity between instances of each cluster (low intra-cluster variance) and the difference between instances of the different clusters (high inter-cluster variance).

Nowadays, CBR systems based on clustered case memories optimize only one criterion for organizing the case memory (Chang and Lai, 2005; Corchado et al., 2004; Fornells et al., 2007b; Vernet and Golobardes, 2003) using a distance error for measuring a single property as the combination of the intra-cluster variance and the inter-cluster variance of the clusters created. Nevertheless, the optimization of both conflicting objectives cannot be done using conventional clustering techniques. The purpose of this chapter is to provide a new perspective through the introduction of CAOS for organizing the case memory through an approach called CAOSCBR. This kind of algorithm from the evolutionary computation family (Holland, 1992) allows grouping elements through an iterative process where all the objectives are simultaneously optimized in order to build a Pareto set of solutions. The Pareto set offers an easy and intuitive mechanism for providing a wide range of case memory configurations that draw a tradeoff among objectives. Moreover, the approach automatically selects the best number of clusters to split the case memory. Finally, the work compares the CAOSCBR performance with respect to (1) a CBR based on a flat case memory organization and (2) a relevant clustered CBR based on Self-Organizing Map (SOM) (Kohonen, 2000) called SOMCBR (Fornells et al., 2007b). The comparison is made taking into account the data set geometry because it influences the result of the clustering process. Concretely, methods are tested with three categories of data sets based on complexity measures (Ho and Basu, 2002; Basu and Ho, 2006), which are able to quantify topological properties such as the class separability or how important are the features in the class discrimination. Thus, the suitability of each algorithm with respect to data set topology can be analyzed.

Next sections briefly summarize the related work on case memory organization, describe how to integrate the multiobjective evolutionary clustering approach in CBR, analyze the CAOSCBR approach and compares its performance to a CBR based on a flat case memory organization and to SOMCBR.

10.2 Related Work

Many points of view have been explored concerning the case memory organization. PROTOS (Porter, 1986), one of the early CBR systems, defines categories of cases and also links that make explicit differences among clusters. K-d trees (Wess et al., 1994) define a tree based on the attributes of cases. Case Retrieval Nets (Lenz et al., 1996) organize the case memory as a graph of feature-value pairs. Decision Diagrams (Nicholson et al., 2006) use a directed graph. Other approaches link the cases by means of the similarity between them such as Fish-and-sink (Schaaf, 1995; Yang and Wu, 2001), using relationships defined by the knowledge of the domain such as the CRASH system (Brown, 1994), or applying clustering techniques for grouping cases in the breast cancer domain (Fornells et al., 2008). The organization can also be done by indexing the memory using the knowledge from the domain like in the BankXX system (Risland et al., 1993). Zenko (Zenko et al., 2005) uses the CN2 algorithm to induce rules that determine prototypes for each cluster. Bichindaritz method (Bichindaritz, 2006) is based on constructing generalizations representing subsets of cases. Malek and Amy (Malek and Amy, 2007) organize the case memory in three parts: prototypical, instance, and a third part which contains cases that cannot be classified in any of the other parts. There are also hybrid approaches based on combining ontologies and rules for indexing the knowledge (Strobbe et al., 2011).

10.3 CAOSCBR: Organizing Case Memories Using CAOS

Clustered case memories allow CBR to reduce the number of cases compared. The new case memory access turns into a procedure based on two steps where (1) a set of C clusters is selected by comparing the new input case with the prototypes, and (2) a set of K cases from the selected clusters are used. Figure 10.1 illustrates an example where this procedure is applied to a clustered case memory. It is important to highlight that the success of this approach mainly depends on two aspects. The first one is related to the amount of experience used, which depends on the strategic selection of the C and K values. The second one depends on the representativeness of the defined prototypes, which are responsible for guiding the retrieval. Although both aspects were addressed in the definition of a methodology for selecting the optimal values according to the data set complexity and the expected performance (Fornells et al., 2007b), there is one open issue that has not been addressed yet. If the clustering accuracy is related to the criteria used to group the elements, it could be interesting to use more than one criterion to group the cases in clusters in order to fit better with the domain typology. Thus, CAOSCBR is a CBR system based on organizing the case memory using the CAOS algorithm.

10.4 Experiments, Results and Discussion

This section evaluates and compares the performance of CAOSCBR with respect to SOMCBR and to a CBR based on a flat case memory organization (called FlatCBR) in terms of accuracy

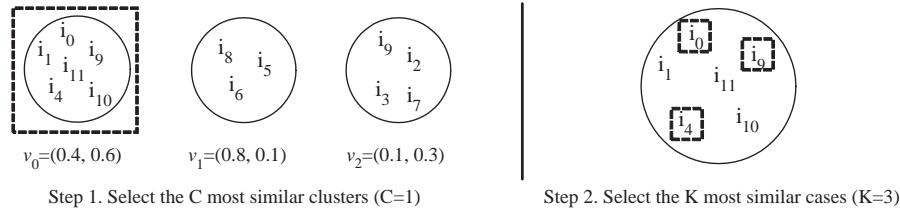


Figure 10.1: Case retrieval using the 3 most similar elements ($K=3$) from the most similar cluster ($C=1$), v_i is the centroid of each cluster. In the example, the similarity is based on the normalized Euclidean distance using an input case composed by two attributes (0.5,0.7).

and retrieval time. The comparison is made considering the data set geometry, due to the fact that the shape and the size of clusters are related to it. This geometry is categorized using complexity measures. These measures allow the system to characterize a data set (Ho and Basu, 2002). First, the experiments are described. Next, the results of the comparison are analyzed. Finally, a more detailed analysis based on complexity measures is discussed.

10.4.1 Experimental Methodology

Test Bed. The objective of the experimentation is to assess the algorithm performance using different typologies of real-world problems. Thus, in order to obtain a test bed similar to that used in previous experiments (Fornells et al., 2007b), 56 data sets were selected according to different number of instances (from 153 to 5000), attributes (from 2 to 60), classes (from 2 to 7) and complexities based on the complexity space defined in Section 10.4.3. Table 10.1 summarizes the data characteristics, where one data set is from a local repository (Bernadó-Mansilla et al., 2002), 23 data sets are from the UCI repository (Asuncion and Newman, 2010) and the other ones are the m -class data sets (where $m > 2$) split in m 2-class data sets, each one discriminating one class against all the others.

CAOS Configuration. CAOS was run with the following parameters: ℓ is 20% of m (the number of data set instances), the N_{EP} is 1000, N_{IP} is 50, min_i and max_i are 5% and 10% of m respectively, N_{niches} is 5, the number of generations is 200, $P_c = 0.7$ and P_μ is $1/m$ (see Section 10.3 for notation details). Summarizing, FlatCBR (*Flat*), SOMCBR 3×3 ($S3 \times 3$), SOMCBR 4×4 ($S4 \times 4$), SOMCBR 5×5 ($S5 \times 5$), CAOSCBR Davies (*CDav*), CAOSCBR Dunn (*CDun*) and CAOSCBR Silhouette (*CSil*) approaches were compared in four different case retrieval processes using all cases of the most similar cluster ($C = 1$), the two most similar clusters ($C = 2$), the three most similar clusters ($C = 3$) and the five most similar clusters ($C = 5$).

Comparison Metrics. CAOSCBR and SOMCBR performances were compared with respect to FlatCBR in order to identify under which conditions the retrieval time was drastically reduced while the accuracy was at least equivalent to FlatCBR. Retrieval time was measured as the mean number of retrieved cases used to solve a new case. The analysis was done through the study

Data set	nCs	nAt	nCl	Comp	Data set	nCs	nAt	nCl	Comp
glass2c1	214	9	2	A	vehicle	846	18	4	B
glass2c2	214	9	2	A	vehicle2c1	849	18	2	B
glass2c4	214	9	2	A	vehicle2c4	849	18	2	B
iris	150	4	3	A	wav2c1	5000	40	2	B
iris2c1	153	4	2	A	wav2c2	5000	40	2	B
iris2c2	153	4	2	A	wav2c3	5000	40	2	B
iris2c3	153	4	2	A	waveform	5000	40	3	B
segment	2310	19	7	A	wbcd	699	9	2	B
segment2c1	2310	19	2	A	wisconsin	699	9	2	B
segment2c2	2310	19	2	A	bal	625	4	3	C
segment2c6	2310	19	2	A	bal2c1	625	4	2	C
segment2c7	2310	19	2	A	bal2c2	625	4	2	C
thy2c1	215	5	2	A	bal2c3	625	4	2	C
thy2c2	215	5	2	A	biopsia	1027	24	2	C
transfusion	748	4	2	A	bpa	345	6	2	C
wdbc	569	30	2	A	glass	214	9	6	C
wine	178	13	3	A	glass2c6	214	9	2	C
wine2c1	178	13	2	A	heart-statlog	270	13	2	C
wine2c2	178	13	2	A	liver-disorders	345	6	2	C
wine2c3	178	13	2	A	monks-1	556	6	2	C
glass2c3	214	9	2	B	monks-2	601	6	2	C
glass2c5	214	9	2	B	monks-3	554	6	2	C
ionosphere	351	34	2	B	pim	768	8	2	C
segment2c3	2310	19	2	B	sonar	208	60	2	C
segment2c4	2310	19	2	B	thyroids	215	5	2	C
segment2c5	2310	19	2	B	vehicle2c2	849	18	2	C
tae	1888	2	2	B	vehicle2c3	849	18	2	C
thy2c3	215	5	2	B	wpbc	198	33	2	C

Table 10.1: Summary of the characteristics of the 56 used data sets. The columns are referred to the number of cases (nCs), to the number of attributes (nAt), to the number of classes (nCl) and to the data set complexity (Comp) defined (A, B, C).

of more to less aggressive case retrieval strategies based on using all cases from: (1) the most similar cluster, (2) the two most similar clusters, (3) the three most similar clusters and (4) the five most similar clusters. It is important to highlight that almost all cases are used in the strategy that recovers five clusters. Each one of the strategies was executed applying a 10-fold stratified cross-validation with the following common configuration: (1) the normalized Euclidean distance was used as distance metric, (2) solutions were proposed using a voting scheme based on the three most similar cases from all the retrieved cases, and (3) no new cases were stored. Moreover, CAOSCBR and SOMCBR results were computed as the mean of several random seeds to minimize the random effects of the clustering techniques. Since SOMCBR needs to set up the maximum number of clusters to find, three configurations of (1) 3×3, (2) 4×4 and (3) 5×5 map sizes¹ were tested. In contrast, CAOSCBR does not need to indicate the number of clusters to discover, but it has to define the strategy for obtaining the best clustering solution from all the non-dominated solutions of the Pareto set. Thus, three different strategies were tested based on (1) the Davies-Bouldin index, (2) the Dunn index and (3) the Silhouette index. As it was explained in Section 2.5.4 the main difference between them is the size of the clusters, where Davies-Bouldin builds the largest clusters and Silhouette the smallest clusters.

Finally, the recommendations pointed out by Demšar (Demšar, 2006) were followed to perform the statistical analysis of the results, which was based on the use of nonparametric tests. More specifically, the following methodology was employed. We first applied the Friedman test (Friedman, 1940) to contrast the null hypothesis that all the learning algorithms obtained the same results on average. If the Friedman test rejected the null hypothesis, we used the non-parametric Nemenyi test (Nemenyi, 1963) to compare all learners to each other. The Nemenyi test defines

¹A map size represents the maximum number of clusters that will be discovered (e.g., 3×3 means 9 clusters).

that two results are significantly different if the corresponding average rank differs by at least a critical difference called CD . We use this method based on the CD for showing graphically the most competitive area.

10.4.2 Comparison of Results

Our first concern was to empirically compare the performance of CAOSCBR with that of SOM-CBR and CBR on the selected collection of real-world problems. For this purpose, we applied the multi-comparison Friedman's test to the results obtained with the seven aforementioned configurations of the three methods. The test rejected the null hypothesis that all the learners performed the same accuracy, on average, at $\alpha = 0.001$. Similarly, the test also rejected the hypothesis that all the learners employed the same number of operations, on average, at $\alpha = 0.05$. Thus, we ran the post-hoc Nemenyi test to evaluate the significant differences among them.

Figure 10.2 ranks the seven configurations for each one of the four case retrieval strategies according to a Nemenyi test at $\alpha = 0.1$ in terms of accuracy and number of operations. The horizontal axis indicates the rank accuracy and the vertical axis indicates the rank operations (in both ranks, the best value is 1 and the worst is 7). Each configuration is represented by an ellipse, where the horizontal and vertical diameters are proportional to the standard deviation of the rank accuracy and the rank operations respectively. Also, for each one of the configurations the average percentage of the case memory used is indicated. Finally, the dashed lines indicate the critical distance (CD) for both ranks regarding to the FlatCBR, being the grey area the optimal one where

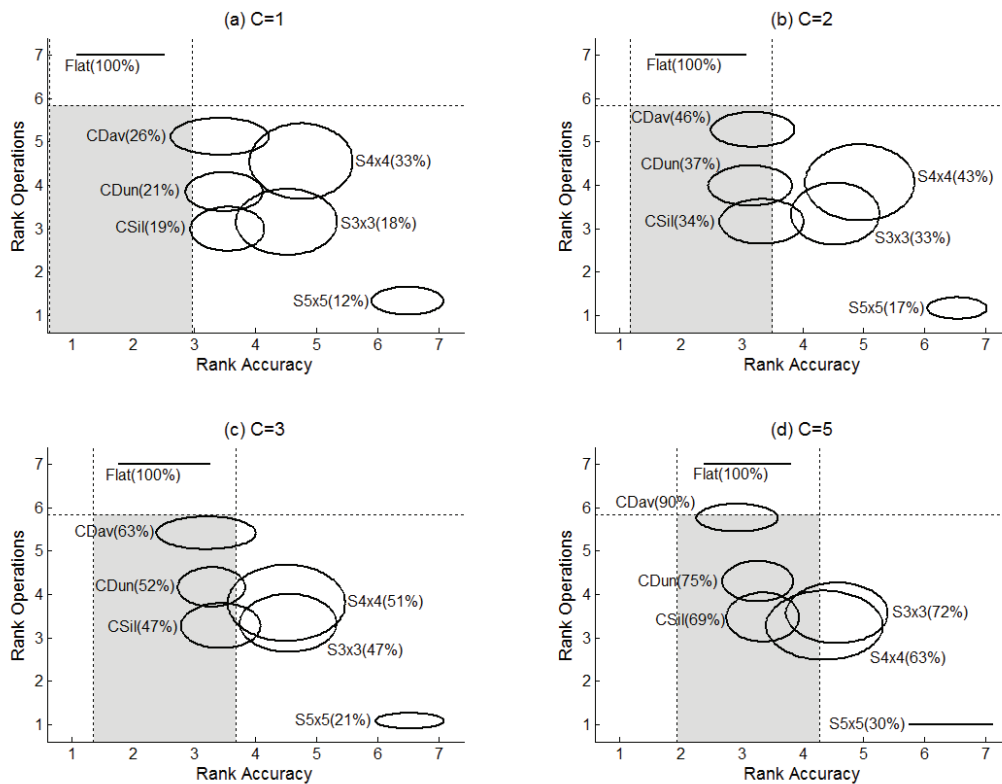


Figure 10.2: Comparison of the average test performance of each configuration against each other with the Nemenyi test for all the data sets. Grey area is the optimal one.

the accuracy is not statistically different to the FlatCBR and the number of operations is statistically different to it. In other words, the configurations in the grey area have the same accuracy than FlatCBR using much less information. A configuration belongs to the area in which the center of the ellipse is placed.

Figure 10.2(a) shows the results when only the most similar cluster is selected. Analyzing the number of operations, CAOSCBR and SOMCBR are statistically different than FlatCBR using among 12% and 32% of the cases from case memory. However, CAOSCBR and SOMCBR configurations are statistically different than FlatCBR in terms of accuracy. Also, it shows the direct relation between the accuracy and the number of cases used from the case memory. In CAOSCBR configurations, *CSil* uses less cases than others and it gets the worst accuracy. In contrast, *CDav* uses more cases and it gets the best accuracy of the three CAOS configurations, being *CDun* an intermediate situation. This behavior in CAOSCBR is due to the way in which the index determines the size of clusters, where Davies-Bouldin index promotes bigger clusters than in Dunn or in Silhouette indexes. On the other hand, $S3 \times 3$ is the best SOMCBR configuration but has an inferior accuracy. $S4 \times 4$ and $S5 \times 5$ split case memory into too many clusters, and this hinders the system from selecting the cases properly. As first conclusion, this kind of retrieval seems to be too much aggressive to offer a competitive performance with respect to FlatCBR.

Figure 10.2(b) and 10.2(c) show the results when the two and three most similar clusters are selected respectively. The performance of the seven configurations follows the global pattern explained in Figure 10.2(a) with the main difference that all configurations use more cases (among 17% and 63% of the case memory). In these new scenarios CAOSCBR configurations are not statistically different than FlatCBR in terms of accuracy, but SOMCBR configurations are still different. CAOSCBR configurations use among 34% and 63% of the case memory, and SOMCBR configurations use among 17% and 51% of it. Finally, Figure 10.2(d) shows the results when the five most similar clusters are selected. Even $S3 \times 3$ uses 72% of the case memory and $S4 \times 4$ uses 63% of it, both are still statistically different than the FlatCBR in terms of accuracy. A similar usage of the case memory happens in *CDun* and *CSil* but they get a performance not statistically different than FlatCBR. In contrast, *CDav* uses near 90% of the case memory and, for this reason, it goes out of the grey area because it behaves like FlatCBR in terms of number of operations.

As a summary of Figure 10.2, CAOS seems to perform better than SOM, concretely *CSil* is the best configuration because it offers the best performance when two, three or five most similar clusters are selected. On the other hand, the worst configuration is $S5 \times 5$, which splits cases into too many clusters and, consequently, the retrieval is not able to properly identify where the cases are.

For more details about the results, consult Table E.1 to Table E.8 in Appendix E, where the accuracy and number of operations of each algorithm for each data set are shown.

10.4.3 Analysis of Results on the Complexity Space

The study performed in the last section has highlighted many important issues on the *average behavior* of CAOSCBR, SOMCBR and CBR. The different methods and configurations excelled in different problems, which may have different characteristics. Thus, the identification of which algorithm can be more suitable to solve a concrete kind of problems is an appealing exercise. In order to obtain a deeper understanding of the problem characteristics that SOMCBR may be most affected by, in (Fornells et al., 2007a) the authors used a set of complexity measures (Ho and Basu, 2002) to study the relation between the retrieval performance and the apparent data set complexity. The results enabled identifying different problem characteristics that negatively affected SOMCBR. Therefore, here we use the same methodology and compare CAOSCBR, SOMCBR and CBR on the complexity space. In what follows we first introduce the original methodology in detail and then we apply it to CAOSCBR.

In the original work (Fornells et al., 2007a) the authors used three complexity measures, defined by Ho and Basu (Ho and Basu, 2002), that tried to capture different sources of problem difficulties: $F3$, $N1$, and $N2$. $F3$ is the maximum feature efficiency, which returns the proportion of instances that a single attribute can discriminate; thus, higher values denote that the instances of different classes can be separated by a single attribute. $N1$ is the fraction of points on the class boundary, which denotes the boundary length; high values of this measure are usually associated to complex problems. $N2$ is the ratio of average intra/inter class nearest neighbor distance, which compares the within-class spread with the distances to the nearest neighbors of other classes; high values of $N2$ indicate that the examples of the same class are disperse. In (Fornells et al., 2007a), the different data sets were mapped onto a two-dimensional complexity space defined by $F3$ and the product of $N1$ and $N2$; thereafter, conclusions about the problem difficulties that affected SOMCBR were extracted.

We followed the same approach to map the collection of 56 real-world problems onto the complexity space. The resulting map is depicted in Figure 10.3. As in the original work, the problems were classified according to three different complexities, resulting in three types of problems of increasing complexity: type *A*, type *B*, and type *C*. The analysis conducted in (Fornells et al., 2007b) showed that SOMCBR (1) outperformed FlatCBR in complex domains (type *B* and *C*) and (2) underperformed FlatCBR in less complex data sets (type *A*). Taking into account this point of view, we wanted to analyze the impact of data set complexity in CAOSCBR.

Figure 10.4 ranks the results of CAOSCBR, SOMCBR and CBR only for the data sets of less complexity (type *A*). All CAOSCBR configurations cannot be considered statistically different to FlatCBR in terms of accuracy. On the other hand, SOM approaches do not obtain representative clusters with this data set complexity as it was identified in (Fornells et al., 2007b). $S5 \times 5$ splits the data sets into a high number of clusters and because of this the selected clusters have not enough information to correctly solve the problem. In contrast, $S3 \times 3$ and $S4 \times 4$ split the data sets into less clusters. SOMCBR improves its accuracy when more clusters are selected, and $S4 \times 4$ is not statistically different to FlatCBR in terms of accuracy with the less aggressive strategy. Therefore,

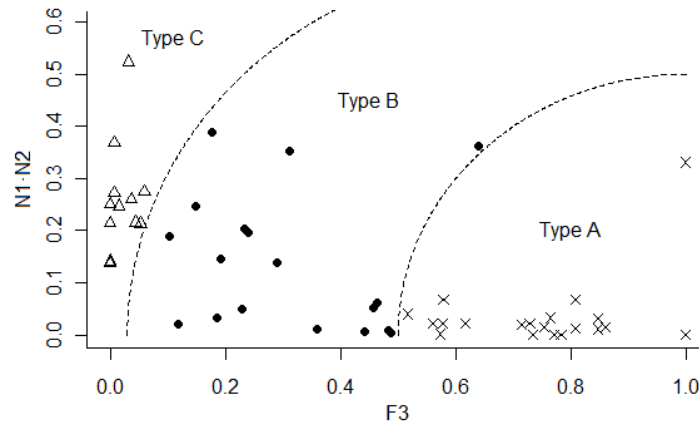


Figure 10.3: Complexity map of the analyzed data sets. Complexity grows from A to C in radial shape.

CDun and *CSil* configurations selecting one cluster have the same accuracy than FlatCBR using only 23% and 24% of the case memory respectively, and $S4 \times 4$ has FlatCBR accuracy using the 79% of the case memory with the less aggressive strategy. With this type of data set complexity, SOMCBR configurations underperform, fact which supports the conclusions pointed out in (Fornells et al., 2007b).

Figure 10.5 ranks the results for problems with intermediate complexity (type *B*). Although the results follow the pattern described in Figure 10.4, there is no configuration that works always better than others and the suitable configuration depends on the problem. Although any configuration is not statistically different in terms of accuracy to FlatCBR with the most aggressive retrieval,

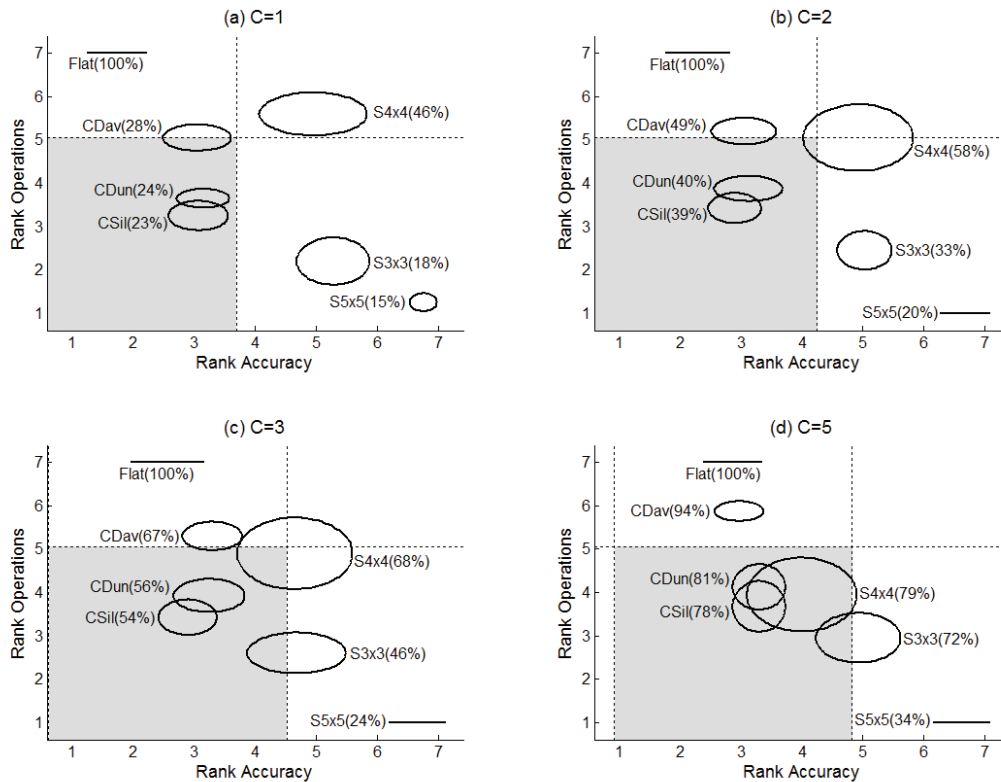


Figure 10.4: Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type A. Grey area is the optimal one.

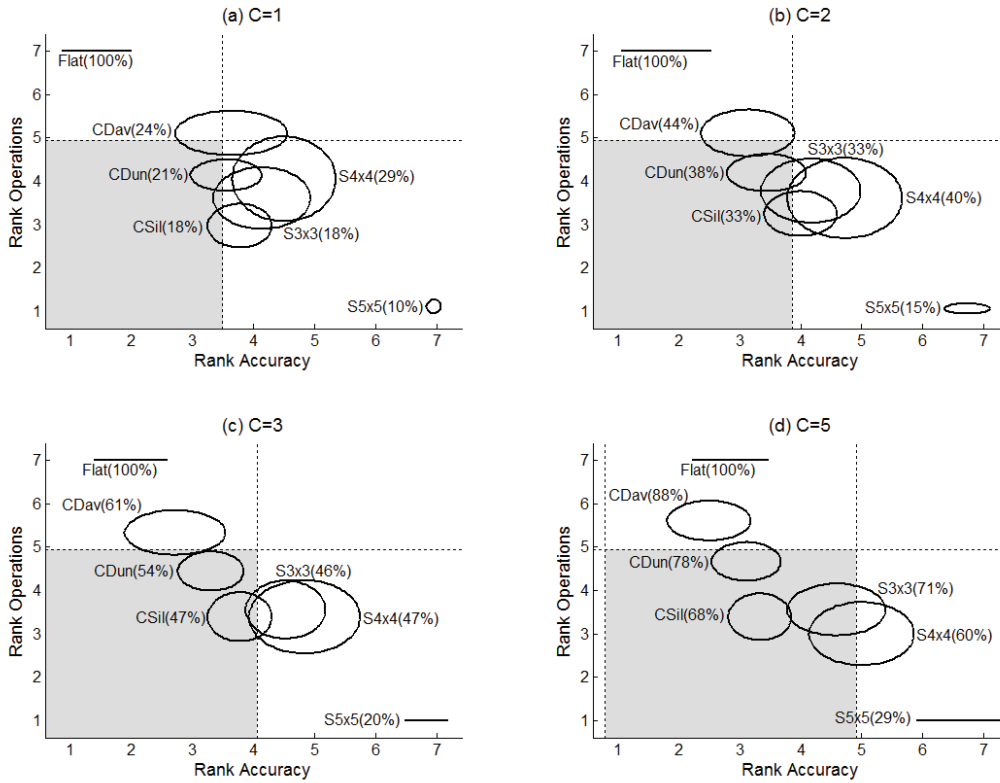


Figure 10.5: Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type B. Grey area is the optimal one.

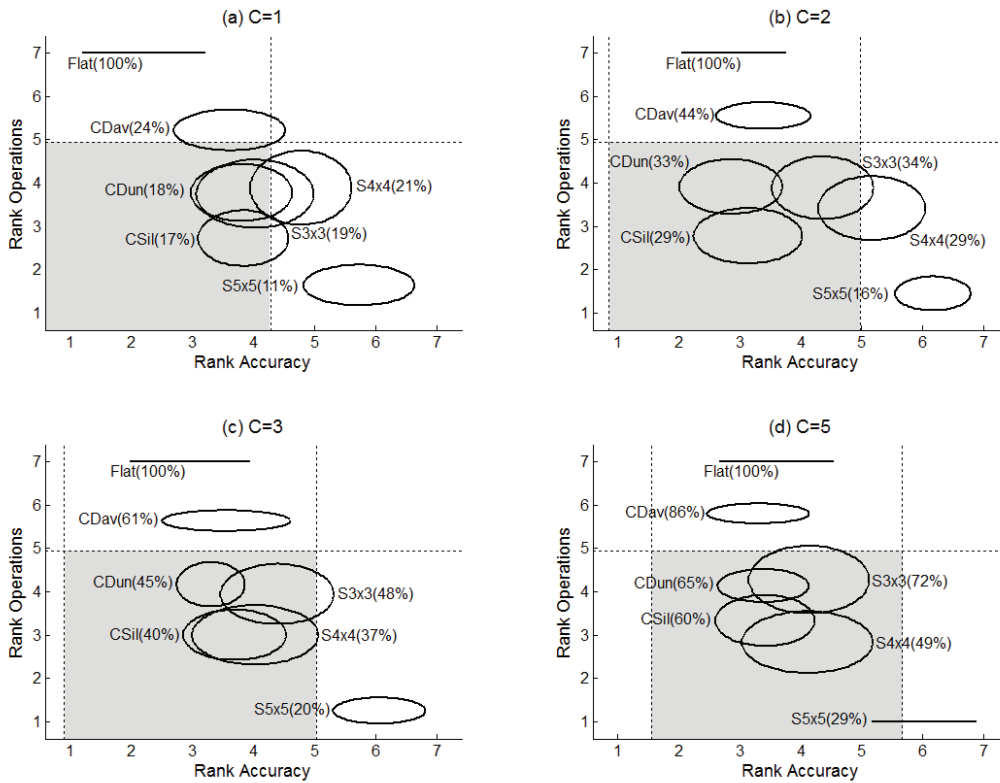


Figure 10.6: Comparison of the average test performance of each configuration against each other with the Nemenyi test for the data sets of type C. Grey area is the optimal one.

CAOSCBR configurations are not statistically different to FlatCBR when more than one cluster is selected. Moreover, *CDun*, *CDav* and *CSil* use only 38%, 44% and 47% of the case memory respectively. On the other hand, $S3 \times 3$ using five clusters is the unique SOMCBR configuration that is not statistically different to FlatCBR in terms of accuracy, and it needs the 71% of the case memory. Therefore, CAOS approach splits the data better into the clusters, and this allows the system to perform a successful selective retrieval. In contrast, SOM splits the data into more clusters and the recovered information is not representative.

Figure 10.6 ranks the results for the most complex problems (type *C*). The results are similar to these described for Figure 10.5. With the most aggressive retrieval strategy, CAOSCBR and some SOMCBR configurations are not statistically different to FlatCBR in terms of accuracy. *CDun*, *CSil* and $S3 \times 3$ configurations have the same accuracy than FlatCBR using only among 17% and 19% of the case memory. SOMCBR configurations have a good performance for this kind of complexity, as it was studied in (Fornells et al., 2007b). Therefore, SOMCBR and CAOSCBR are equivalent to FlatCBR recovering only one cluster for these hard problems, due to both approaches split the cases into representative clusters.

10.4.4 Discussion

Overall, the analysis has shown that CAOS approach splits the data into more representative clusters than SOM, due to the fact that CAOS promotes the deviation and the compactness of the clusters. SOMCBR improves its accuracy when more clusters are selected at the retrieval. This feature is highlighted with the data sets of complexity *A* and *B*, where SOMCBR is not as competitive as CAOSCBR. For the data sets of type *C* SOMCBR improves its performance, however CAOSCBR continues performing better than it. Finally, it is important to emphasize that, for all complexity types, CAOSCBR performs better than FlatCBR because it has the same accuracy using less information.

10.5 Summary and Conclusions

CBR systems solve new problems based on experiences stored in a case memory, whose size and organization are key issues for optimal performance in terms of computational time and accuracy. If the case memory is organized as a flat memory, it is necessary to look up all the cases. This is an important drawback because the retrieval time increases with the size of the case memory. When the case memory is structured, the computational time is improved due to the possibility of doing selective retrievals. In this chapter we have presented a CBR system based on the organization of a case memory using CAOS, a multiobjective evolutionary clustering algorithm. CAOS builds a set of clustering solutions that optimize two complementary objectives based on the intra-cluster and the inter-cluster variance, and uses an automatic procedure that uses clustering validation indexes to select the most suitable one.

The performance of the CBR based on CAOS (CAOSCBR) has been compared to a CBR with a

flat memory (FlatCBR) and with another CBR based on the clustering algorithm SOM (SOMCBR), using several retrieval strategies from more to less aggressive. The experimental results have shown that CAOSCBR improves the performance with respect to FlatCBR and to SOMCBR. This is because CAOSCBR splits the case memory into representative clusters, and this makes it possible to use aggressive retrieval strategies obtaining the same accuracy as FlatCBR and SOMCBR with less information. Also, we have analyzed whether CAOSCBR performance is influenced by the data set complexity, following the same procedure done in previous studies with SOMCBR. We have observed that CAOSCBR is not significantly affected by the data set complexity, concluding that it is more stable than SOMCBR.

This analysis sets the basis for further conducting research on CAOSCBR. This future work will target three objectives. First, we want to study whether new objectives may result in better clustering solutions. For example, complexity metrics could be used as new objectives in the clustering process, in order to generate the best distribution of clusters for a given problem. Second, we will build new methods to retrieve the best solution in the Pareto set considering new variables in addition to the approach used. Finally, the use of data subsets (Bacardit and Llorà, 2009) or parallel evolutionary algorithms (Cantu-Paz, 2000; Freitas, 2002) could be interesting in order to reduce the computational time in the clustering process when the case memory needs to be updated in the CBR retain phase (Plaza and Arcos, 1990).

One paper related to this contribution have been published in the framework of the MID-CBR project (Garcia-Piquer et al., 2011).

Part IV

Conclusions and Further Work

Chapter 11

Work Done, Lessons Learned and Future Work Lines

Throughout this thesis several challenges related to MC algorithms based on MOEAs have been faced up. First, a MC framework has been defined in order to set the research bases of the thesis. Then, several contributions related to the challenges to be faced up have been presented and tested. To sum up, the MC algorithm proposed has been tested in some real applications of projects in which I have been involved during my PhD studies. This chapter summarizes the conclusions obtained from each contribution and exposes the global conclusions of the thesis. Finally, some further research is proposed.

11.1 Recapitulation and Key Conclusions

This thesis started by presenting clustering techniques as mechanisms to identify patterns and discover relationships with the aim of obtaining wisdom from a large amount of data. Then, the main lacks of the conventional clustering algorithms were identified when it is necessary to obtain clusters according to several criteria simultaneously and they cannot be summarized in a single objective obtaining a collection of objectives to be optimized. Thus, multiobjective clustering (MC) techniques were presented in order to be applied to complex domains where it is necessary to use several objectives to identify understanding patterns in the data. Later, multiobjective evolutionary algorithms (MOEAs) have been presented as a capable technique to be used by this kind of clustering algorithms for optimizing several objectives simultaneously obtaining a collection of clustering solutions with different trade-off between objectives. Finally, three challenges related to MOEAs are identified to be faced up in the thesis: (1) the definition and exploration of the search space where the evolutionary process is going to search the solutions, (2) the scalability of the method with large data, and (3) the retrieval of the most suitable solution from the collection of solutions with different trade-off between objectives.

In the second part of the thesis, **CAOS** algorithm was proposed as a new multiobjective evolutionary clustering algorithm. First, the framework of **CAOS** was presented highlighting that it is based on the multiobjective evolutionary algorithm PESA-II with slightly modifications for

improving the performance of the algorithm. Next, we defined the objectives to optimize in the clustering process with the aim of obtaining clusters with elements similar among them and different to the elements of other clusters. After that, the aforementioned challenges were faced up with the aim of obtaining a robust algorithm. As proceeds, we summarize the contributions related to each one of the challenges and provide the key conclusions extracted from each one.

Definition and exploration of the Search Space. Three of the most relevant and useful individual representations in EAs were analyzed in order to identify the representation that properly explores the search space. These three representations were: (1) prototype-based, (2) label-based and (3) graph-based. We tested the performance of these individual representations in MC algorithms, using CAOS, by comparing the results quantitatively and qualitatively. The quantitative measure of a solution was the accuracy value calculated with the Adjusted Rand Index according to the original classes of the problem. On the other hand, the qualitative analysis was related to the shape and features of the clusters. Furthermore, the analysis was extended by comparing the results with respect to the most used single-objective clustering algorithms.

The experimental comparison between the three representations showed that label-based and graph-based representations can identify clusters of arbitrary shapes but they have problems to explore properly the search space. This last issue can be clearly observed when the initial population includes low quality individuals because the final solutions are not adequate, so the performance depends on their individual initialization. Moreover, they are sensible to scattered data sets due to a small change in an individual gene can split up clusters that should not be divided. On the other hand, the prototype-based representation can only find ellipsoidal clusters but it explores the search space more accurately than the other two representations, so it is independent to the initialization of the population, and it is more robust to scattered data sets due to a change in a gene of an individual only moves the prototype of a cluster without changing all the clustering structure. It is important to highlight that the individuals of the label-based and graph-based representations are related to the size of the data set, so they consume a big amount of memory when they are applied to large data sets. On the other hand, the individuals of the prototype-based representation are not related to the size of the data set, so the memory usage is lower and it is the most suitable representation to be applied to large data sets. Nevertheless, the experiments showed that the selection of the most suitable representation depends on the domain of the problem because there is not an individual representation that works properly for all kind of problems.

Moreover, the experimentation showed that, in quantitative terms, the differences among the solutions obtained by the studied CAOS representations and by the analyzed single-objective clustering algorithms are not statistically significant. Furthermore, in qualitative terms, CAOS solutions divides the data set space in more representative and well explained clusters than the single objective algorithms.

Large data management. The main lack of MC algorithms based on MOEAs is that the evolutionary techniques are expensive in terms of computational time and memory usage when they

are applied to a big amount of data since they do an intensive use of computations, therefore it is necessary to scale-up them. The method proposed to be applied to MC algorithms is based on stratifying the complete data set in several strata in order to use less data in the clustering process but without drastically penalize the accuracy of the system. We analyzed three stratification techniques based on: (1) the original classes of the data set, (2) random instances from the data set, and (3) the clusters found applying a fast approximate clustering method. The first one is a supervised strategy and it was used to compare the performance of the other two strategies. The impact of applying several stratification techniques in MC algorithms was tested using CAOS. Specifically, the approach is based on dividing the data set in some strata and alternate them in each cycle of the evolutionary algorithm to increase the generalization of the system and avoiding the bias.

The experimental analysis showed that the two unsupervised strategies are not statistically different from the strategy based on classes in terms of accuracy. Thus, the random and cluster based strategies can be considered equivalent to the strategy based on classes in terms of accuracy. In terms of speedup, the strategy based on clusters has a lower speedup than the other two strategies because it needs to find the approximate clusters before to stratify the data set. Taking into account that the strategy based on classes cannot be used in clustering problems because the original classes are unknown, the most suitable strategy to scaling-up CAOS is based on random strata due to it is not different in terms of accuracy according to the other two strategies and it has a higher speedup than the strategy based on clusters.

Moreover, the experimentation showed that there are statistical differences among the proposed stratification strategies and the approach that uses the complete data set. Nevertheless, it seems that the accuracy differences among them are relatively small and, moreover, they considerably reduce the computational time of the algorithm scaling-up it properly.

Selection of the Most Suitable Solution. The algorithms based on MOEAs usually return a Pareto set where there is no solution better than the others for each objective. Thus, to return the solution that solves a particular problem it is necessary to retrieve it from the Pareto set according to a specific point of view. There are two main strategies to automatically retrieve the most suitable solution. The first one is based on retrieving the solution according to the shape of the Pareto set, but it does not take into account the specific characteristics of the problem. Thus, in clustering problems on the one hand, this strategy retrieves a solution without taking into account the shape and quality of the clusters and it can return a solution with a good trade-off between objectives but with incomprehensible clusters. On the other hand, the second strategy is based on specific characteristics of the problem. Therefore, in clustering problems, it retrieves a solution according to the quality and shape of clusters using clustering validation indexes but it does not take into account the value of the objectives, so it can return a solution with a bad trade-off between them and this is not the aim of MC. As a result, in this contribution we proposed the combination of both methods in order to select the solution according to a clustering validation index from the region of the Pareto set where all the solutions with a good trade-off between objectives are placed. We

called this region sweet spot.

The experimental results showed that in the proposed method it is necessary to discard only the solutions that extremely maximize each one of the objectives, because these solutions have usually very large or very small clusters that become incomprehensible clustering solutions. Moreover, the results showed that to apply the clustering validation indexes to the sweet spot of the Pareto set improves the results of the other retrieval methods. Moreover, these methods are not significantly different from the solutions retrieved by a supervised method, so they work as well as a method that uses the classes of the problem to retrieve the most suitable solution. It is important to highlight that the proposed method can be applied in any kind of MOEA.

The third part of the dissertation presented real-world applications of CAOS in order to test its capabilities. Specifically, it was applied in three problems listed as follows: (1) the identification of vulnerabilities in networks for helping security analysts to analyze security tests, (2) the identification of some improvements to be made in university degrees for helping educational experts to adapt the degrees to some specific requirements, and (3) the organization of the case memory in case-based reasoning systems in order to improve the performance of the classifier. In the three problems, CAOS demonstrated its ability to obtain solutions that properly organizes a collection of data obtaining understandable clusters that can help experts.

The global conclusions that can be extracted from the thesis are positive. All the three kinds of contributions studied are able to improve MC algorithms based on MOEAs and can be combined among themselves (see Figure 11.1) in order to obtain a robust algorithm focused on solving real-

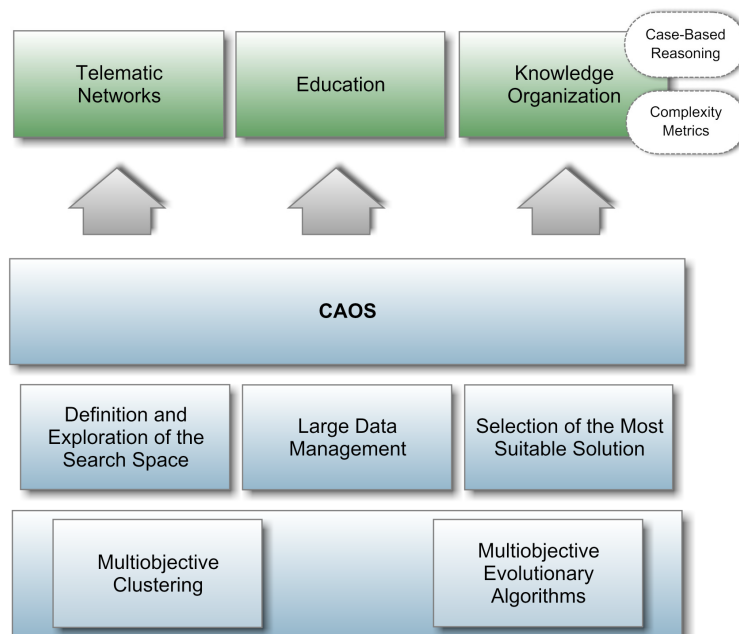


Figure 11.1: This thesis has faced up three challenges –the definition and exploration of the search space, the scalability with large data sets and the retrieval of solutions– in the context of MC based on MOEAs to develop the CAOS algorithm, which has been successfully tested in different real-world projects. Moreover, CAOS has been used in combination with other techniques such as Case-Based Reasoning and complexity measures. Starting from the bottom we can see the contexts of the thesis, the challenges faced, the system obtained and the domains of application in real-world projects.

world problems with large data. In the introduction of the thesis, we have exposed the description of the content of the human mind made by Russell Ackoff. From this description we can extract that knowledge is the appropriate collection of useful information but it must be understood in order to obtain it into wisdom. The main objective of the thesis revolves around this idea, trying to extract understandable patterns from data that allow experts to obtain wisdom. Instead of conventional clustering algorithms, MC algorithms can find patterns according to several objectives simultaneously. Thus, experts can define several criteria to be taken into account in the clustering process with the goal of obtaining more understandable clusters.

Therefore, the result of this thesis has been the design of a new algorithm which is able to face up the three aforementioned challenges, and also which has demonstrated the capability of returning competitive and differentiated results throughout the experimentation done in a large amount of artificial and real-world data sets. Furthermore, the contributions were not limited only to the scientific framework and have also been applied to real world in the projects focused on networks security, on education and on the design of CBR systems. Finally, it must be emphasized that several publications, related to the projects that I have participated in, have been written in order for the scientific community to learn about this research. These publications are listed in the following:

- Garcia-Piquer, A., Fornells, A., Orriols-Puig, A., Corral, G., and Golobardes, E. (2011). Data Classification through an Evolutionary Approach Based on Multiple Criteria. *Knowledge and Information Systems* (ISI index in the upper quartile), 10.1007/s10115-011-0462-9.
- Corral, G., Garcia-Piquer, A., Orriols-Puig, A., Fornells, A., and Golobardes, E. (2011). Analysis of Vulnerability Assessment Results based on CAOS. *Applied Soft Computing Journal* (ISI index in the upper quartile), 11:4321–4331.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., Orriols-Puig, A., and Cugota, L. (2010b). Análisis de titulaciones universitarias basadas en competencias mediante una técnica de clustering evolutiva multiobjetivo. In *Proceedings of the III Congreso Español de Informática (CEDI 2010). Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA)*, pages 345–354.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2010a). Validación de competencias en titulaciones universitarias usando minería de datos. *Revista Iberoamericana de Tecnologías del Aprendizaje (RITA), Capítulo Español de la Sociedad de la Educación del IEEE*, 5(1):23–29.
- Vernet, D., Nicolas, R., Golobardes, E., Fornells, A., and Garcia-Piquer, A. (2010). Intelligent tutoring system framework for the acquisition of knowledge and competences. In *The 40th Annual Frontiers in Education (FIE) Conference*, pages T4G 1–2. IEEE.
- Corral, G., Garcia-Piquer, A., Orriols-Puig, A., Fornells, A., and Golobardes, E. (2009b). Multiobjective evolutionary clustering approach to security vulnerability assessments. In

Hybrid Artificial Intelligence Systems, volume 5572, pages 597–603.

- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2009a). Assessment of competences in university degrees using data mining techniques. In *FINTDI 2009: Fomento e Innovación con Nuevas Tecnologías en la Docencia de la Ingeniería*, pages 45–49. IEEE.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2009b). Identification of subject typologies through artificial intelligence techniques to study the competences achievement of the new computer engineers. In *The 39th Annual Frontiers in Education (FIE) Conference*, pages T3D 1–2. IEEE.
- Garcia-Piquer, A., Fornells, A., Golobardes, E. and Cugota, L. *Anexo 4. Estudio sobre las tipologías de asignaturas*. In Golobardes, E. and Madrazo, L. (2009b). *Guía para la evaluación de competencias en el área de Ingeniería y Arquitectura*. Guías de evaluación de competencias. AQU Catalunya, Barcelona.
- Garcia-Piquer, A., Fornells, A., Golobardes, E. and Cugota, L. *Anexo 4. Estudi sobre les tipologies d'assignatures*. In Golobardes, E. and Madrazo, L. (2009b). *Guia per a l'avaluació de competències en l'àrea d'Enginyeria i Arquitectura*. Guies d'avaluació de competències. AQU Catalunya, Barcelona.

Moreover, two papers referred to the contribution related to the definition and exploration of the search space and the contribution related to the large data management have been submitted to other two ISI indexed journals under the titles "Large-Scale Experimental Evaluation of Cluster Representations for Multiobjective Evolutionary Clustering" (Applied Soft Computing Journal) and "Scaling-Up Multiobjective Evolutionary Clustering Algorithms using stratification" (Soft Computing Journal). The paper referred to the contribution related to selection of the most suitable solution will be submitted in the next few months.

In the next section, we expose the further work related to the presented contributions.

11.2 Forthcoming Research

Throughout this thesis different multiobjective clustering applications in real-world problems have been analyzed. However, these techniques can be applied in a large amount of problems from decision making in situations where several criteria have to be considered or where it is necessary to obtain a consensus with different points of view. An example of the first case is to help in the project management grouping the tasks to be performed and the tasks to be discarded, obtaining solutions with different trade-offs among three essential criteria in this domain: time, resources and quality. This could help project managers to properly plan and justify their decisions. An example related to take into account several points of view is, in the medical context, the categorization of particular tumors in which the experts are not agree. Using a multiobjective clustering technique, a categorization of these tumors can be obtained with a trade-off among the different points of view

of the experts. Nevertheless, it would be necessary to carefully analyze the feasibility of these applications and, if they are viable, to carefully study in detail the obtained results.

In technical terms, further research has been identified at the end of the chapter related to each contribution. Moreover, we can identify some global further work that is described as follows.

Types of clustering. Currently, CAOS is a MC algorithm that obtains partitioned hard clusters. However, experts can consider that the application domain can be characterized by hierarchical or fuzzy clusters. To adapt the system to these kinds of clustering can be interesting in order to obtain more understandable clusters. This adaptation will affect to the evolutionary and clustering validation processes. Specifically, to adapt the system to hierarchical clustering it will be necessary to propose a new individual representation based on represent the clusters hierarchy (Chis, 2008; Chakrabarti et al., 2006; Castellanos-Garzón et al., 2009). To adapt the system to find fuzzy clusters is only necessary to change the assignation of the instances to each cluster. Nevertheless, the graph-based and label-based representations proposed in this thesis will need some modifications in order to assign an instance to several clusters (Le, 1995). On the other hand, the prototype-based representation can be adapted to find fuzzy clusters without major changes (Dong et al., 2010). Moreover, it will be necessary to select other validation indexes to be used in the validation step that are able to evaluate hierarchical and fuzzy clustering (Halkidi et al., 2001).

Data dimensionality. We can identify two improvements related to the data dimensionality. The first one is referred to better identify overlapped clusters. Concretely, in some data sets the suitable clusters can be overlapped and it is difficult to identify them. Thus, some methods like kernel methods (Vapnik, 1995) can map the instances in inner product spaces in order to identify nonlinearly separable clusters in input space. Despite kernel methods are typically used in classification problems, we can extrapolate them in order to calculate distances between instances in kernel spaces (Dhillon et al., 2004; Tzortzis and Likas, 2008). The second improvement is related to the reduction of the data dimensionality. We have faced up to the scalability lack in MC algorithms based on MOEAs related to the number of instances. However, we have not tested our method with any unsupervised dimension reduction technique such as PCA (Hotelling, 1933) or Kernel PCA (Schölkopf et al., 1998). This can be interesting in order to scale-up the method in problems with high dimensionality and to better generalize in some noisy domains.

Cluster explanations. The main objective of the thesis is to extract understandable patterns from data that allow experts to obtain wisdom. To allow experts to better understand the obtained clusters can be interesting to add a cluster explanation to the final solution (Bélanger and Martel, 2005; Corral et al., 2009a). Rule-based classifier systems can be useful techniques in order to identify the relations between the attributes of each cluster. Specifically, rules are typically used to classify but the process can be extrapolated to clusters in order to interpret them as the concept to be learned (Hotho et al., 2003).

Obviously, these ideas are only the tip of the iceberg of further research in MC and MOEAs. Both techniques have successfully demonstrated their capabilities to solve several problems and they are promising techniques that can be applied in different fields of life with a pinch of innovation and creativity.

11.3 Summary

This chapter has concluded the dissertation of the present thesis. It has summarized the challenges confronted to achieve the main objective of the thesis: the definition and implementation of a new MC algorithm based on a MOEA that faces up (1) the definition and exploration of the search space, (2) the ability to work with large data sets with reasonable computational time and memory usage, and (3) the selection of the best solution from the Pareto set of potential solutions. On the other hand, the new MC algorithm has been tested in three particular problems of life related to (1) education, (2) information systems and networks, and (3) case-based reasoning systems.

Part V

Appendix

Appendix A

Description of Single-Objective Clustering Algorithms

K-means (Hartigan and Wong, 1979), EM (Dempster et al., 1977), and SOM (Kohonen, 2000) represent three of the most influential paradigms of clustering: the inductive, the statistical and the neural network-based. Moreover, the *x*-means (Pelleg and Moore, 2000) algorithm is added to the analysis since it automatically determines the number of clusters to find, unlike the aforementioned ones, and it is based on *k*-means. All approaches are focused on minimizing the intra-cluster variance.

K-means. This technique is one of the most influential clustering techniques (Hartigan and Wong, 1979) due to its simplicity and the competent results demonstrated in real-world applications. *k*-means splits a given training data set into *k* disjoint clusters specified by the user through the following procedure. Initially, it randomly selects *k* instances of the training data set, and each one is considered the *centroid* of one of the clusters. Then, the algorithm iteratively performs the following two steps. First, each example is assigned to the cluster with the closest centroid using a similarity measure such as for example the Euclidean distance. Second, each cluster centroid is re-allocated to the center of all the examples assigned to it. These two steps are repeated until none of the cluster's centroids change. Its main drawbacks are the selection of the number of clusters, the dependency of results due to the cluster initialization, the presence of outliers negatively affects the results and it tends to falter when data cannot be well described by reasonably separated spherical balls.

X-means. The definition of the number of clusters in *k*-means was overcome in *x*-means (Pelleg and Moore, 2000). Given a minimum number min_k and a maximum number max_k of clusters specified by the user, the approach starts running *k*-means with *k* equal to min_k and it continues dividing the existing centroids where they are needed until max_k . In each iteration, the algorithm splits some centroids in two and, for each new centroid, it runs a local *k*-means only for the data points that were associated with the original centroid. The best partition is calculated by using

the *Bayesian information criterion*. Finally, the centroid set that achieves the best score and the k value are returned. Although x -means preserves the advantages of k -means and it simplifies the k value selection, the initial distribution and an incorrect choice of initial centroids may have a great impact on both performance and distribution.

EM. It is an efficient iterative procedure based on statistically methods to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data (Dempster et al., 1977). In clustering problems, the objective of the algorithm is to estimate the model parameters for which the observed data are the most likely. EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters and try to maximize de likelihood of the clusters. Each iteration of the EM algorithm consists of two processes: the expectation step (E-step), and the maximization step (M-step). In the E-step, the probability of each instance to belong to a cluster is estimated with the observed data. In the M-step, the likelihood function is maximized assigning the instances to each cluster according to the probabilities found in the previous step. The algorithm is guaranteed to increase the likelihood at each iteration, therefore the convergence is assured. EM has similar drawbacks as k -means since the definition of the number of clusters, dependency of initialization and result may be a local minimum.

SOM. Self-Organizing Map (Kohonen, 2000) is one of the most well-known and used neural networks techniques in clustering and data visualization (Kaski et al., 2003) tasks. Its architecture is composed by two layers to project the high-dimensional input data space into a low-dimensional space with simple geometric relationships. The input layer represents the original N -dimensional space of the data set using N neurons. On the other hand, the output layer is usually a 2-dimensional grid of $K \times K$ neurons where each one represents a cluster described by a director vector of N dimensions. The maximum number of clusters to find is $K \times K$ and has to be fixed by the user. The approach iteratively maps elements into neurons according a function error between director vector and input data and, in each step, the director vectors and their neighbors are updated to represent better the data groups. The process ends when a maximum number of iterations is reached or until the global error between each director vector and its elements is less than a threshold value. The main characteristics of SOM is its suitability for working with high-dimensional problems, it provides an easy way to present data groups, it is not biased towards class imbalances, it maintains the relation of the original topology; and, it intrinsically performs feature selection. In contrast, the SOM adjustment is complex and the learning process depends on the order of the training examples.

Appendix B

Full Results of the Experimentation of Chapter 5

This appendix details the results obtained by each one of the **CAOS** representations and by the single-objective clustering methods in the experimentation described in Chapter 5. Table B.1 and Table B.2 show the accuracy results for each one of the analyzed approaches for the 35 artificial data sets and the 35 real-world data sets, respectively. Table B.3 and Table B.4 show the number of cluster found by each one of the analyzed approaches for the 35 artificial data sets and the 35 real-world data sets, respectively.

Data set	k-means	Single-objective clustering		r-means	CAOS Graph				CAOS Prototype				CAOS Label					
		SOVI	EM		Adj.Rand	Davies	Dunn	Silhouette	Adj.Rand	Davies	Dunn	Silhouette	Adj.Rand	Davies	Dunn	Silhouette		
100d-10c	0.80±0.02	0.81±0.00	0.92±0.02	0.45±0.02	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
100d-4c	0.82±0.00	0.91±0.00	1.00±0.00	0.84±0.03	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
100d-10c	0.78±0.00	0.88±0.00	0.94±0.01	0.46±0.11	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
10d-4c	0.79±0.08	0.57±0.00	1.00±0.00	0.96±0.01	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
2d-10c	0.82±0.03	0.53±0.00	0.92±0.02	0.53±0.01	0.94±0.00	0.68±0.00	0.68±0.00	0.93±0.00	0.81±0.10	0.60±0.14	0.94±0.00	0.68±0.07	0.68±0.06	0.94±0.00	0.68±0.07	0.68±0.06	0.68±0.07	
2d-4c	0.81±0.00	0.44±0.00	1.00±0.00	0.95±0.01	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
curves1	1.00±0.00	0.40±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
curves2	0.32±0.01	0.36±0.00	0.45±0.00	0.00±0.00	1.00±0.00	0.46±0.00	0.50±0.00	1.00±0.00	1.00±0.00	0.29±0.08	0.25±0.04	1.00±0.00	0.25±0.04	1.00±0.00	0.37±0.03	0.34±0.01	0.32±0.01	
darboard1	0.31±0.00	0.09±0.00	0.29±0.00	0.00±0.00	1.00±0.00	0.10±0.05	0.13±0.05	1.00±0.00	0.25±0.02	0.33±0.02	0.33±0.02	0.05±0.01	0.38±0.00	1.00±0.00	0.11±0.05	0.19±0.06	0.00±0.00	
darboard2	0.40±0.00	0.14±0.00	0.41±0.00	0.05±0.02	1.00±0.00	0.37±0.02	0.36±0.01	1.00±0.00	0.60±0.00	0.71±0.00	0.36±0.00	0.37±0.00	0.37±0.00	1.00±0.00	0.38±0.03	0.37±0.03	0.37±0.03	
donut1	0.67±0.00	0.11±0.00	0.67±0.00	0.17±0.00	1.00±0.00	0.59±0.02	0.57±0.01	1.00±0.00	0.99±0.00	0.99±0.02	0.54±0.00	0.58±0.00	0.58±0.00	1.00±0.00	0.46±0.07	0.35±0.06	0.41±0.06	
donut2	0.39±0.01	0.06±0.00	0.39±0.00	0.01±0.00	1.00±0.00	0.00±0.00	0.00±0.00	1.00±0.00	0.00±0.00	0.48±0.03	0.48±0.03	0.48±0.02	0.98±0.00	0.58±0.05	0.58±0.05	0.58±0.05	0.58±0.05	
donut3	0.75±0.00	0.24±0.00	0.87±0.00	0.44±0.00	1.00±0.00	0.71±0.05	1.00±0.00	1.00±0.00	1.00±0.00	0.67±0.04	0.67±0.04	0.76±0.01	0.76±0.01	1.00±0.00	0.69±0.05	0.57±0.04	0.73±0.14	
donutcurves	0.53±0.00	0.45±0.00	0.66±0.00	0.50±0.00	1.00±0.00	0.91±0.00	0.50±0.00	1.00±0.00	0.91±0.00	0.50±0.08	0.50±0.00	0.50±0.00	0.50±0.00	1.00±0.00	0.50±0.03	0.50±0.01	0.50±0.01	
long1	1.00±0.00	0.23±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	0.90±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
long2	1.00±0.00	0.20±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
long3	1.00±0.00	0.12±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
longsquare	0.78±0.04	0.50±0.00	0.98±0.00	0.27±0.00	1.00±0.00	0.27±0.00	0.27±0.00	1.00±0.00	0.98±0.00	0.98±0.05	0.27±0.02	0.27±0.02	0.27±0.02	1.00±0.00	0.91±0.03	0.91±0.03	0.91±0.03	
size1	0.96±0.00	0.40±0.00	0.95±0.00	0.55±0.02	1.00±0.00	0.96±0.00	0.00±0.00	1.00±0.00	0.96±0.00	0.96±0.00	0.27±0.02	0.96±0.00	0.95±0.00	0.95±0.00	0.91±0.03	0.95±0.02	0.95±0.00	0.94±0.00
size2	0.97±0.00	0.26±0.00	0.97±0.00	0.60±0.00	1.00±0.00	0.00±0.00	0.00±0.00	1.00±0.00	0.97±0.00	0.97±0.00	0.97±0.00	0.97±0.00	0.97±0.00	1.00±0.00	0.95±0.02	0.95±0.02	0.95±0.02	
size3	0.94±0.00	0.18±0.00	0.98±0.00	0.71±0.00	1.00±0.00	0.97±0.00	0.00±0.00	1.00±0.00	0.97±0.00	0.97±0.00	0.97±0.00	0.97±0.00	0.97±0.00	1.00±0.00	0.95±0.01	0.95±0.01	0.95±0.01	
size4	0.90±0.00	0.13±0.00	0.98±0.00	0.77±0.00	1.00±0.00	0.95±0.00	0.00±0.00	1.00±0.00	0.96±0.00	0.96±0.00	0.96±0.00	0.96±0.00	0.96±0.00	1.00±0.00	0.94±0.01	0.94±0.01	0.94±0.01	
size5	0.91±0.00	0.11±0.00	0.98±0.00	0.83±0.00	1.00±0.00	0.01±0.00	0.01±0.00	1.00±0.00	0.98±0.00	0.98±0.00	0.97±0.00	0.78±0.10	0.78±0.10	1.00±0.00	0.01±0.00	0.77±0.11	0.77±0.11	
smile1	0.69±0.00	0.47±0.00	0.96±0.00	0.60±0.00	1.00±0.00	0.73±0.00	0.71±0.00	1.00±0.00	0.99±0.00	0.99±0.01	0.62±0.02	0.60±0.00	0.60±0.00	1.00±0.00	0.62±0.01	0.63±0.01	0.63±0.01	
smile2	0.49±0.02	0.26±0.00	0.78±0.00	0.37±0.01	1.00±0.00	0.33±0.00	0.53±0.00	1.00±0.00	0.70±0.03	0.44±0.02	0.54±0.00	0.54±0.00	0.54±0.00	1.00±0.00	0.55±0.02	0.55±0.01	0.55±0.01	
smile3	0.31±0.00	0.23±0.00	0.49±0.00	0.21±0.00	1.00±0.00	0.19±0.02	0.11±0.00	1.00±0.00	0.73±0.00	0.84±0.01	0.28±0.00	0.28±0.04	0.31±0.02	1.00±0.00	0.29±0.02	0.31±0.05	0.30±0.03	
spiral	0.08±0.00	0.03±0.00	0.10±0.00	0.07±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	0.84±0.01	0.84±0.01	0.05±0.03	0.05±0.03	0.02±0.01	1.00±0.00	0.06±0.04	0.06±0.04	0.06±0.04	
spiral-square	0.41±0.00	0.23±0.00	0.49±0.00	0.41±0.00	1.00±0.00	0.42±0.00	0.42±0.00	1.00±0.00	0.51±0.01	0.51±0.01	0.42±0.00	0.42±0.00	0.42±0.00	1.00±0.00	0.42±0.00	0.42±0.00	0.42±0.00	
square1	0.95±0.00	0.54±0.00	0.95±0.00	0.95±0.00	1.00±0.00	0.95±0.00	0.00±0.00	1.00±0.00	0.95±0.00	0.95±0.00	0.95±0.00	0.95±0.00	0.95±0.00	1.00±0.00	0.95±0.00	0.95±0.00	0.95±0.00	
square2	0.92±0.00	0.50±0.00	0.92±0.00	0.46±0.00	1.00±0.00	0.93±0.00	0.00±0.00	1.00±0.00	0.91±0.00	0.91±0.00	0.91±0.00	0.91±0.00	0.91±0.00	1.00±0.00	0.82±0.02	0.78±0.07	0.78±0.07	
square3	0.86±0.00	0.46±0.00	0.85±0.00	0.45±0.00	1.00±0.00	0.83±0.00	0.00±0.00	1.00±0.00	0.86±0.01	0.86±0.01	0.86±0.00	0.86±0.01	0.86±0.01	1.00±0.00	0.75±0.02	0.72±0.08	0.66±0.10	
square4	0.79±0.00	0.40±0.00	0.76±0.00	0.42±0.00	1.00±0.00	0.00±0.00	0.00±0.00	1.00±0.00	0.79±0.00	0.79±0.00	0.79±0.00	0.79±0.00	0.79±0.00	1.00±0.00	0.65±0.03	0.62±0.06	0.57±0.00	
square5	0.68±0.00	0.33±0.00	0.64±0.00	0.38±0.00	1.00±0.00	0.00±0.00	0.00±0.00	1.00±0.00	0.67±0.01	0.67±0.01	0.66±0.01	0.66±0.01	0.66±0.01	1.00±0.00	0.60±0.03	0.60±0.00	0.60±0.00	
triangle	0.91±0.00	0.76±0.00	1.00±0.00	0.62±0.06	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	
triangle2	0.89±0.00	0.40±0.00	0.99±0.00	0.37±0.00	1.00±0.00	1.70±0.00	1.00±0.00	1.00±0.00	0.99±0.00	0.91±0.00	0.91±0.00	0.91±0.00	0.91±0.00	1.00±0.00	0.98±0.01	0.98±0.01	0.97±0.00	

Table B.1: Accuracy and standard deviation of conventional algorithms and CAOS solutions with the artificial data sets. The accuracy was calculated with the Adjusted Rand index.

Data set	Single-objective clustering				CAOS Graph			CAOS Prototype			CAOS Label					
	k-means	SOM	EM	x-means	Adj.Rand	Davies	Dunn	Silhouette	Adj.Rand	Davies	Dunn	Silhouette	Adj.Rand	Davies	Dunn	Silhouette
bal	0.16±0.00	0.08±0.00	0.14±0.00	0.09±0.01	0.01±0.00	0.00±0.00	0.06±0.00	0.00±0.00	0.25±0.00	0.08±0.01	0.08±0.00	0.01±0.27	0.18±0.00	0.00±0.00	0.00±0.00	0.00±0.00
bioptn	0.21±0.00	0.11±0.00	0.20±0.00	0.16±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.28±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.01	0.00±0.00	0.00±0.00
bpa	0.01±0.00	0.00±0.00	0.01±0.00	0.00±0.00	0.19±0.04	0.00±0.00	0.00±0.00	0.00±0.00	0.07±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.03±0.02	0.00±0.00	0.00±0.00	0.00±0.00
dermatology	0.68±0.02	0.69±0.00	0.77±0.00	0.57±0.00	0.86±0.01	0.43±0.41	0.21±0.00	0.87±0.00	0.87±0.02	0.21±0.00	0.21±0.00	0.21±0.17	0.66±0.02	0.36±0.27	0.07±0.12	0.73±0.00
ecoli	0.66±0.01	0.42±0.00	0.72±0.00	0.64±0.02	0.76±0.03	0.00±0.00	0.00±0.00	0.00±0.00	0.75±0.02	0.00±0.02	0.00±0.02	0.00±0.02	0.73±0.03	0.00±0.00	0.00±0.00	0.00±0.00
glass	0.19±0.02	0.19±0.00	0.24±0.01	0.15±0.01	0.27±0.02	0.04±0.08	0.00±0.05	0.18±0.00	0.29±0.02	0.00±0.00	0.00±0.00	0.00±0.00	0.33±0.02	0.07±0.08	0.04±0.05	0.18±0.00
heart-staffg	0.26±0.04	0.14±0.00	0.38±0.01	0.31±0.04	0.29±0.05	0.11±0.00	0.00±0.00	0.11±0.00	0.28±0.08	0.05±0.02	0.09±0.04	0.05±0.02	0.20±0.03	0.11±0.00	0.00±0.00	0.11±0.00
iono	0.23±0.02	0.13±0.00	0.38±0.01	0.28±0.05	0.56±0.04	0.00±0.00	0.00±0.00	0.25±0.03	0.55±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.68±0.03	0.00±0.00	0.00±0.00	0.00±0.00
iris	0.63±0.02	0.57±0.00	0.76±0.00	0.57±0.00	0.79±0.06	0.57±0.00	0.57±0.00	0.57±0.00	0.74±0.01	0.57±0.00	0.57±0.00	0.57±0.00	0.70±0.08	0.57±0.00	0.57±0.00	0.57±0.00
liver-disorders	0.00±0.00	0.00±0.00	0.01±0.00	0.00±0.00	0.17±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.11±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.03±0.02	0.00±0.00	0.00±0.00	0.00±0.00
pendigits	0.63±0.04	0.35±0.00	0.65±0.00	0.32±0.00	0.63±0.02	0.00±0.00	0.00±0.00	0.00±0.00	0.53±0.02	0.43±0.01	0.46±0.01	0.46±0.01	0.50±0.02	0.12±0.00	0.09±0.00	0.16±0.00
pin	0.12±0.00	0.05±0.00	0.11±0.00	0.10±0.00	0.04±0.00	0.00±0.00	0.00±0.00	0.01±0.01	0.01±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.04±0.00	0.00±0.00	0.00±0.00	0.01±0.00
segment	0.51±0.02	0.50±0.00	0.50±0.00	0.36±0.00	0.64±0.04	0.00±0.00	0.00±0.00	0.24±0.00	0.58±0.01	0.10±0.00	0.10±0.00	0.10±0.00	0.56±0.03	0.09±0.13	0.08±0.08	0.09±0.08
segment2c1	0.04±0.00	0.04±0.00	0.04±0.00	0.00±0.01	0.55±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.32±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.51±0.07	0.00±0.21	0.00±0.13	0.00±0.16
segment2c2	0.45±0.01	0.11±0.00	0.37±0.00	0.28±0.02	1.00±0.00	0.00±0.00	0.00±0.00	0.57±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.08	0.13±0.16	0.19±0.07	0.13±0.10
segment2c3	0.04±0.01	0.03±0.00	0.03±0.00	0.00±0.01	0.32±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.50±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.43±0.07	0.00±0.21	0.00±0.13	0.00±0.16
segment2c4	0.07±0.01	0.04±0.00	0.12±0.00	0.11±0.02	0.08±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.35±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.32±0.02	0.01±0.18	0.00±0.15	0.01±0.15
segment2c5	0.05±0.00	0.02±0.00	0.03±0.00	0.00±0.01	0.42±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.22±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.32±0.07	0.00±0.20	0.00±0.13	0.00±0.16
segment2c6	0.12±0.00	0.06±0.00	0.19±0.00	0.13±0.03	0.24±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.54±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.41±0.08	0.00±0.16	0.00±0.13	0.00±0.14
segment2c7	0.29±0.06	0.09±0.00	0.24±0.00	0.08±0.01	1.00±0.00	0.00±0.00	0.00±0.00	0.56±0.00	0.99±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.99±0.08	0.13±0.18	0.19±0.08	0.13±0.12
sonar	0.04±0.01	0.03±0.00	0.05±0.02	0.02±0.00	0.37±0.11	0.00±0.00	0.00±0.00	0.03±0.02	0.21±0.01	0.00±0.01	0.00±0.00	0.00±0.00	0.18±0.03	0.00±0.40	0.00±0.40	0.01±0.00
tue	0.29±0.03	0.15±0.00	0.24±0.00	0.37±0.03	0.43±0.00	0.26±0.00	0.02±0.00	0.01±0.00	0.44±0.05	0.14±0.03	0.19±0.11	0.23±0.06	0.37±0.03	0.26±0.07	0.25±0.03	0.00±0.00
thy	0.63±0.00	0.20±0.00	0.89±0.00	0.63±0.00	0.92±0.01	0.22±0.14	0.10±0.05	0.69±0.22	0.86±0.03	0.06±0.12	0.06±0.12	0.02±0.00	0.88±0.03	0.02±0.00	0.02±0.00	0.53±0.00
transfusion	0.01±0.00	0.02±0.00	0.03±0.00	0.00±0.00	0.06±0.00	0.02±0.00	0.00±0.00	0.02±0.00	0.04±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.03±0.00	0.01±0.01	0.01±0.01	0.00±0.00
vehicle	0.11±0.00	0.07±0.00	0.16±0.00	0.06±0.00	0.19±0.01	0.06±0.01	0.00±0.00	0.00±0.00	0.14±0.00	0.08±0.00	0.08±0.00	0.08±0.00	0.10±0.01	0.06±0.01	0.06±0.01	0.00±0.00
vehicle2c1	0.06±0.00	0.01±0.00	0.07±0.00	0.00±0.00	0.37±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.21±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.18±0.00	0.01±0.00	0.01±0.00	0.01±0.23
vehicle2c2	0.11±0.00	0.00±0.00	0.12±0.00	0.06±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.07±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.01±0.22
vehicle2c3	0.10±0.00	0.00±0.00	0.10±0.00	0.04±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.04±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.25
vehicle2c4	0.04±0.00	0.00±0.00	0.08±0.00	0.00±0.00	0.09±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.12±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.20±0.00	0.00±0.00	0.00±0.00	0.01±0.24
wdbc	0.73±0.00	0.23±0.00	0.68±0.00	0.35±0.00	0.78±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.76±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.74±0.02	0.00±0.00	0.00±0.00	0.35±0.09
wine	0.84±0.00	0.54±0.00	0.92±0.00	0.78±0.00	0.91±0.02	0.82±0.00	0.00±0.00	0.82±0.00	0.88±0.01	0.00±0.00	0.00±0.00	0.17±0.43	0.88±0.07	0.66±0.00	0.00±0.00	0.82±0.00
wine	0.84±0.00	0.30±0.00	0.78±0.00	0.49±0.00	0.86±0.00	0.00±0.00	0.00±0.00	0.30±0.09	0.88±0.01	0.68±0.17	0.68±0.17	0.68±0.17	0.86±0.00	0.01±0.01	0.03±0.01	0.17±0.11
wybc	0.04±0.00	0.01±0.00	0.02±0.00	0.00±0.00	0.21±0.15	0.00±0.00	0.00±0.00	0.03±0.04	0.08±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.11±0.00	0.00±0.00	0.00±0.00	0.03±0.00
yeast	0.18±0.00	0.07±0.00	0.15±0.00	0.12±0.00	0.25±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.28±0.02	0.00±0.00	0.00±0.00	0.00±0.00	0.17±0.01	0.00±0.00	0.00±0.00	0.01±0.00
zoo	0.70±0.06	0.78±0.00	0.80±0.01	0.57±0.12	0.95±0.00	0.68±0.00	0.68±0.00	0.68±0.00	0.95±0.00	0.34±0.05	0.34±0.05	0.41±0.02	0.92±0.00	0.30±0.03	0.52±0.03	0.37±0.02

Table B.2: Accuracy and standard deviation of conventional algorithms and CAOS solutions with the real-world data sets. The accuracy was calculated with the Adjusted Rand index.

Data set	kmeans		Single-objective clustering		CAOS Graph		CAOS Prototype		CAOS Label			
		SCVI	EM		AdjRand	Davies	Dunn	Silhouette	AdjRand	Davies	Dunn	Silhouette
100d-10c	12±0	21±0	9±0	4±0	10±0	10±0	10±0	10±1	10±1	10±0	10±0	10±0
100d-4c	5±0	8±0	4±0	3±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0
100f-10c	12±0	21±0	11±0	4±1	10±0	10±0	10±0	10±0	10±0	10±0	10±0	10±0
100f-4c	4±0	9±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0
2d-10c	11±0	2±0	10±0	4±0	10±0	6±0	6±0	8±1	7±1	7±1	7±1	10±0
2d-4c	3±0	9±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0	6±0
curves1	2±0	6±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	4±0
curves2	6±0	8±0	4±0	2±0	2±0	4±0	4±0	2±0	170±41	7±41	8±1	2±0
darboand1	7±0	8±0	5±0	2±0	4±0	5±1	6±1	2±0	168±9	7±9	8±1	3±1
darboand2	4±0	9±0	4±0	2±0	4±0	7±1	6±1	3±1	10±1	8±1	7±1	3±0
donut1	4±0	9±0	4±0	2±0	2±0	7±1	6±1	2±0	14±2	7±2	7±0	2±0
donut2	5±0	9±0	5±0	2±0	2±0	2±0	2±0	2±0	7±1	7±1	7±1	2±0
donut3	7±0	9±0	4±0	2±0	3±0	6±1	3±0	3±1	14±1	8±1	12±1	3±1
donutcurves	9±0	8±0	9±0	2±0	5±0	2±0	2±0	4±2	19±5	3±5	2±0	4±0
long1	2±0	9±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0
long2	2±0	9±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0
long3	2±0	9±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0	2±0
longsquare	8±0	16±0	7±0	2±0	6±0	2±0	2±0	6±1	2±0	2±0	2±0	2±0
sizes1	4±0	9±0	4±0	2±0	4±0	4±0	2±0	4±0	4±0	4±0	4±0	4±0
sizes2	4±0	9±0	4±0	2±0	4±0	2±0	2±0	4±0	4±0	4±0	4±0	4±0
sizes3	4±0	9±0	4±0	2±0	4±0	2±0	2±0	4±0	4±0	4±0	4±0	4±0
sizes4	3±0	9±0	4±0	2±0	4±0	4±0	3±0	4±0	4±0	4±0	3±1	4±1
sizes5	3±0	9±0	4±0	2±0	4±0	2±0	2±0	4±0	4±0	4±1	4±1	4±1
smile1	9±0	9±0	4±0	3±0	4±0	7±0	3±0	4±3	13±2	4±2	3±0	3±1
smile2	3±0	9±0	4±0	3±0	4±0	3±0	3±0	4±1	18±3	3±3	3±0	3±1
smile3	9±0	9±0	8±0	2±0	4±0	3±2	2±0	4±2	20±1	6±1	16±2	4±1
spiral	2±0	8±0	2±0	2±0	2±0	2±0	2±0	2±0	214±22	4±22	150±59	3±1
spiral-square	2±0	9±0	3±0	2±0	6±0	2±0	2±0	5±0	5±0	4±0	2±0	2±0
square1	4±0	9±0	4±0	2±0	4±0	4±0	2±0	4±0	4±0	4±0	4±0	4±0
square2	4±0	9±0	4±0	2±0	4±0	4±0	2±0	4±0	4±0	4±0	4±0	4±0
square3	4±0	9±0	4±0	2±0	4±0	4±0	2±0	4±0	4±0	4±0	4±0	4±0
square4	4±0	9±0	4±0	2±0	4±1	2±0	2±0	4±0	4±0	4±0	4±0	4±0
square5	4±0	9±0	4±0	3±1	4±1	2±0	2±0	4±0	4±0	4±0	4±0	4±0
triangle1	5±0	9±0	4±0	3±1	4±0	4±0	4±0	4±0	4±0	4±0	4±0	4±0
triangle2	4±0	9±0	4±0	2±0	4±0	2±0	2±0	4±0	4±0	4±0	4±1	4±0

Table B.3: The number of clusters and its standard deviation of conventional algorithms and CAOS solutions with the real-world data sets.

Data set	k-means		Single-objective clustering		CAOS Graph		CAOS Prototype		CAOS Label			
		x-means	SOM	EM	Adj.Rand	Davies	Dunn	Silhouette	Adj.Rand	Davies	Dunn	Silhouette
bal	3±0	9±0	5±0	2±0	5±5	2±0	2±0	2±0	2±0	2±0	2±0	2±0
bopn	2±0	9±0	2±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	2±0
bpa	3±0	25±0	9±0	2±0	6±6	3±0	2±0	2±0	4±1	2±0	2±0	2±0
dermatology	6±0	8±0	4±0	3±0	5±5	3±2	2±0	2±0	6±1	2±0	2±0	4±0
ecoli	4±0	9±0	4±0	3±0	7±7	2±0	2±0	2±1	6±1	2±0	2±0	2±0
glass	12±0	16±0	4±0	3±0	4±4	3±1	3±1	2±0	4±1	2±0	3±1	5±0
heart-stalog	2±0	8±0	2±0	2±0	2±2	8±0	2±0	36±18	2±1	8±0	2±0	8±0
iono	3±0	9±0	4±0	3±0	2±2	2±0	2±0	2±1	2±1	2±0	2±0	2±0
iris	3±0	8±0	3±0	2±0	3±3	3±0	2±0	2±0	3±0	3±0	2±0	3±0
liver-disorders	13±0	16±0	9±0	2±0	6±6	3±0	2±1	2±0	4±8	2±1	2±0	2±0
penigits	13±0	16±0	13±0	4±0	9±2	2±0	2±0	9±0	17±4	7±0	6±0	8±0
pin	3±0	9±0	4±0	2±0	7±7	2±0	2±0	2±0	2±0	2±0	2±0	2±0
segment	6±0	9±0	8±0	4±0	7±7	2±0	2±0	2±0	6±0	2±0	2±1	3±1
segment2c1	6±0	9±0	5±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c2	2±0	9±0	3±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c3	6±0	9±0	13±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c4	2±0	9±0	3±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c5	6±0	9±0	8±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c6	2±0	9±0	3±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±1	3±1
segment2c7	3±0	9±0	4±0	4±0	2±2	2±0	2±0	2±0	2±1	2±0	2±1	3±1
sonar	5±0	24±0	5±0	4±0	3±3	2±1	2±0	15±30	2±8	2±0	4±1	2±1
tue	2±0	9±0	4±0	2±0	2±2	5±0	3±0	6±2	2±0	4±1	3±1	2±0
thy	3±0	9±0	3±0	3±0	3±3	4±1	3±0	2±0	3±3	2±0	2±0	5±0
transfusion	10±0	9±0	4±0	2±0	4±4	2±0	2±0	2±0	2±0	2±1	2±0	2±0
vehicle	10±0	16±0	7±0	4±0	5±5	2±1	2±0	3±0	4±0	3±1	3±0	2±0
vehicle2c1	9±0	16±0	6±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	3±3
vehicle2c2	2±0	16±0	2±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	3±3
vehicle2c3	2±0	16±0	2±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	3±3
vehicle2c4	9±0	16±0	7±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	3±3
wdbc	2±0	9±0	2±0	4±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	2±1
wine	3±0	7±0	3±0	3±0	3±3	3±0	2±0	2±0	3±1	3±0	2±0	3±0
wisc	2±0	9±0	2±0	3±0	2±2	2±0	2±0	2±0	2±0	2±0	2±0	2±0
wybc	2±0	9±0	2±0	3±0	3±3	2±1	2±0	2±0	2±1	2±0	3±0	2±1
yeast	6±0	9±0	9±0	2±0	10±10	3±0	2±0	2±0	7±1	2±0	2±0	2±1
zoo	4±0	9±0	4±0	3±1	6±6	7±0	7±0	4±0	6±1	8±1	8±1	5±1

Table B.4: The number of clusters and its standard deviation of conventional algorithms and CAOS solutions with the real-world data sets.

Appendix C

Full Results of the Experimentation of Chapter 6

This appendix details the results obtained by each one of the data subsets strategies of CAOS in the experimentation described in Chapter 6. Table C.1 to Table C.8 show the accuracy results for each one of the analyzed approaches for the 75 artificial data sets using the 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of the instances of the original data sets, respectively. Table C.9 to Table C.16 show the accuracy results for each one of the analyzed approaches for the 25 real-world data sets using the 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of the instances of the original data sets, respectively. Table C.17 to Table C.24 show the time in seconds used to find the results for each one of the analyzed approaches for the 75 artificial data sets using the 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of the instances of the original data sets, respectively. Table C.25 to Table C.32 show the time in seconds used to find the results for each one of the analyzed approaches for the 25 real-world data sets using the 50%, 34%, 25%, 20%, 10%, 7%, 5% and 4% of the instances of the original data sets, respectively.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	1	1	1
100d-10c-175m	1	1	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	0.964082	0.999966	0.997512
100d-20c-175m	1	0.957819	1	0.997547
100d-20c-75m	1	1	0.988239	1
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.999888	0.99432	0.994818
100d-30c-175m	0.98434	0.97644	0.97232	0.973232
100d-30c-75m	0.99324	0.98686	0.97863	0.97793
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	1	1	1
10d-10c-175m	1	1	1	1
10d-10c-75m	1	1	1	1
10d-20c-125m	1	0.976028	0.995823	0.977062
10d-20c-175m	1	0.978004	0.983742	0.967484
10d-20c-75m	1	0.96663	0.991286	1
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.87629	0.87329	0.87231
10d-30c-175m	0.974734	0.883903	0.88254	0.88132
10d-30c-75m	0.995329	0.886829	0.884231	0.881354
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	1	1
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	1	0.986997	1
20d-20c-175m	1	0.987712	0.995568	0.99309
20d-20c-75m	1	0.982695	1	0.991478
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.908929	0.912349	0.923449
20d-30c-175m	0.98898	0.869966	0.924354	0.942334
20d-30c-75m	0.969109	0.917607	0.943247	0.939875
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
24d-10c-125m	0.936128	0.935537	0.936294	0.930335
24d-10c-175m	0.999289	0.952871	0.909235	0.909586
24d-10c-75m	0.915825	0.886051	0.892486	0.842066
24d-20c-125m	0.682375	0.655466	0.681505	0.686394
24d-20c-175m	0.86862	0.870938	0.854747	0.862997
24d-20c-75m	0.604147	0.613223	0.603254	0.586622
24d-2c-125m	1	1	1	1
24d-2c-175m	1	1	1	1
24d-2c-75m	1	1	1	1
24d-30c-125m	0.577244	0.584292	0.58323	0.582312
24d-30c-175m	0.871619	0.769555	0.83345	0.869324
24d-30c-75m	0.332496	0.323229	0.323423	0.32423
24d-5c-125m	0.848217	0.815432	0.840675	0.787148
24d-5c-175m	0.903762	0.99918	0.99959	0.950643
24d-5c-75m	0.945857	0.95084	0.938022	0.952346
54d-10c-125m	1	0.979026	0.980423	0.983423
54d-10c-175m	1	1	1	1
54d-10c-75m	1	1	1	1
54d-20c-125m	0.957281	0.880714	0.933442	0.924455
54d-20c-175m	0.964107	0.890271	0.900956	0.879424
54d-20c-75m	0.951824	0.880274	0.923422	0.868602
54d-2c-125m	1	1	1	1
54d-2c-175m	1	1	1	1
54d-2c-75m	1	1	1	1
54d-30c-125m	0.931717	0.847506	0.872343	0.863406
54d-30c-175m	0.931432	0.849254	0.89232	0.88123
54d-30c-75m	0.800023	0.792269	0.793423	0.786349
54d-5c-125m	1	1	0.999632	1
54d-5c-175m	1	1	1	1
54d-5c-75m	1	1	1	1

Table C.1: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	1	1	1
100d-10c-175m	1	1	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	1	0.978361	0.987034
100d-20c-175m	1	1	0.999829	0.987654
100d-20c-75m	1	1	0.990857	0.996083
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.96466	0.96534	0.971681
100d-30c-175m	0.98434	0.943838	0.938553	0.965616
100d-30c-75m	0.99324	0.95839	0.957338	0.949209
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	1	1	1
10d-10c-175m	1	1	1	1
10d-10c-75m	1	1	1	1
10d-20c-125m	1	0.986904	0.9558	0.960913
10d-20c-175m	1	0.967484	0.977786	0.957728
10d-20c-75m	1	1	0.979711	0.975343
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.931432	0.909505	0.922046
10d-30c-175m	0.974734	0.917896	0.856175	0.848888
10d-30c-75m	0.995329	0.863799	0.904098	0.887952
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	1	1
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.922011	0.993367	0.987143
20d-20c-175m	1	0.994686	0.992926	0.98618
20d-20c-75m	1	0.982955	0.991478	0.974341
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.874237	0.917312	0.930783
20d-30c-175m	0.98898	0.96081	0.91932	0.867865
20d-30c-75m	0.969109	0.863041	0.921409	0.935058
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
2d-10c-125m	0.936128	0.876673	0.904496	0.953101
2d-10c-175m	0.999289	0.907074	0.954585	0.926287
2d-10c-75m	0.915825	0.867577	0.924656	0.897473
2d-20c-125m	0.682375	0.667478	0.679066	0.658951
2d-20c-175m	0.86862	0.852356	0.804814	0.871642
2d-20c-75m	0.604147	0.558628	0.5762	0.61873
2d-2c-125m	1	1	1	1
2d-2c-175m	1	1	1	1
2d-2c-75m	1	1	1	1
2d-30c-125m	0.577244	0.542086	0.533829	0.533586
2d-30c-175m	0.871619	0.823822	0.804462	0.810625
2d-30c-75m	0.332496	0.315367	0.32864	0.306747
2d-5c-125m	0.848217	0.792419	0.802923	0.774598
2d-5c-175m	0.903762	0.99918	1	0.99918
2d-5c-75m	0.945857	0.869004	0.903311	0.924407
5d-10c-125m	1	0.979026	0.979026	0.979026
5d-10c-175m	1	1	1	1
5d-10c-75m	1	1	1	1
5d-20c-125m	0.957281	0.927712	0.85739	0.917438
5d-20c-175m	0.964107	0.813437	0.910532	0.843721
5d-20c-75m	0.951824	0.88228	0.860716	0.857582
5d-2c-125m	1	1	1	1
5d-2c-175m	1	1	1	1
5d-2c-75m	1	1	1	1
5d-30c-125m	0.931717	0.886995	0.875471	0.890192
5d-30c-175m	0.931432	0.829724	0.864355	0.858398
5d-30c-75m	0.800023	0.782652	0.782671	0.769942
5d-5c-125m	1	0.999265	0.999632	0.999632
5d-5c-175m	1	1	1	1
5d-5c-75m	1	1	1	1

Table C.2: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	1	1	1
100d-10c-175m	1	1	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	0.981673	0.933395	0.965125
100d-20c-175m	1	0.988581	0.987654	0.967661
100d-20c-75m	1	0.991363	0.983961	0.974654
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.966116	0.964233	0.966535
100d-30c-175m	0.98434	0.89631	0.899462	0.921774
100d-30c-75m	0.99324	0.91034	0.924953	0.92348
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	1	1	1
10d-10c-175m	1	1	1	1
10d-10c-75m	1	1	1	1
10d-20c-125m	1	0.940517	0.957413	0.937658
10d-20c-175m	1	0.953741	0.943955	0.947097
10d-20c-75m	1	0.909442	0.955286	0.949299
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.879615	0.820402	0.86348
10d-30c-175m	0.974734	0.892807	0.852967	0.862416
10d-30c-75m	0.995329	0.912848	0.839607	0.885733
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	1	1
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.967388	0.963404	0.901899
20d-20c-175m	1	0.976293	0.96727	0.96093
20d-20c-75m	1	0.93523	0.944438	0.976899
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.925813	0.829919	0.844276
20d-30c-175m	0.98898	0.857649	0.906304	0.844882
20d-30c-75m	0.969109	0.870429	0.913076	0.916563
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
24d-10c-125m	0.936128	0.877792	0.887611	0.917057
24d-10c-175m	0.999289	0.923839	0.895935	0.958617
24d-10c-75m	0.915825	0.835722	0.800565	0.829386
24d-20c-125m	0.682375	0.638289	0.632972	0.665809
24d-20c-175m	0.86862	0.861808	0.881613	0.843053
24d-20c-75m	0.604147	0.557596	0.576488	0.556036
24d-2c-125m	1	1	1	1
24d-2c-175m	1	1	1	1
24d-2c-75m	1	1	1	1
24d-30c-125m	0.577244	0.529114	0.51117	0.543853
24d-30c-175m	0.871619	0.828461	0.85218	0.778463
24d-30c-75m	0.332496	0.296304	0.310551	0.314192
24d-5c-125m	0.848217	0.735032	0.717558	0.704442
24d-5c-175m	0.903762	0.935407	1	0.939634
24d-5c-75m	0.945857	0.937287	0.920235	0.887927
54d-10c-125m	1	0.979026	0.979026	0.979026
54d-10c-175m	1	1	1	1
54d-10c-75m	1	1	1	1
54d-20c-125m	0.957281	0.877155	0.833226	0.863259
54d-20c-175m	0.964107	0.854789	0.900784	0.850398
54d-20c-75m	0.951824	0.82553	0.86532	0.898731
54d-2c-125m	1	1	1	1
54d-2c-175m	1	1	1	1
54d-2c-75m	1	1	1	1
54d-30c-125m	0.931717	0.841353	0.87752	0.908545
54d-30c-175m	0.931432	0.838837	0.826915	0.840195
54d-30c-75m	0.800023	0.75679	0.743988	0.76349
54d-5c-125m	1	0.999265	1	1
54d-5c-175m	1	1	1	1
54d-5c-75m	1	1	1	1

Table C.3: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	1	1	1
100d-10c-175m	1	1	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	0.882893	0.941513	0.920049
100d-20c-175m	1	0.964009	0.973231	0.97612
100d-20c-75m	1	0.977455	0.980859	0.992585
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.943086	0.954323	0.971655
100d-30c-175m	0.98434	0.889428	0.850482	0.922416
100d-30c-75m	0.99324	0.90239	0.91835	0.90594
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	1	1	1
10d-10c-175m	1	1	1	1
10d-10c-75m	1	0.97877	0.98761	1
10d-20c-125m	1	0.902308	0.945933	0.901231
10d-20c-175m	1	0.903628	0.858764	0.934503
10d-20c-75m	1	0.888046	0.88962	0.873455
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.860028	0.877219	0.83068
10d-30c-175m	0.974734	0.878559	0.835206	0.891876
10d-30c-75m	0.995329	0.831538	0.85364	0.818358
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	1	1
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.95067	0.937254	0.937502
20d-20c-175m	1	0.941818	0.952584	0.946597
20d-20c-75m	1	0.939555	0.905153	0.95294
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.91158	0.879764	0.870402
20d-30c-175m	0.98898	0.840251	0.882148	0.797356
20d-30c-75m	0.969109	0.821566	0.842918	0.885141
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
2d-10c-125m	0.936128	0.870774	0.881827	0.900716
2d-10c-175m	0.999289	0.927984	0.893264	0.925665
2d-10c-75m	0.915825	0.795287	0.819358	0.761802
2d-20c-125m	0.682375	0.615868	0.62109	0.623244
2d-20c-175m	0.86862	0.875028	0.825412	0.79635
2d-20c-75m	0.604147	0.55903	0.526874	0.582315
2d-2c-125m	1	1	1	1
2d-2c-175m	1	1	1	1
2d-2c-75m	1	1	1	1
2d-30c-125m	0.577244	0.52862	0.511124	0.524377
2d-30c-175m	0.871619	0.752277	0.789205	0.809468
2d-30c-75m	0.332496	0.306487	0.302461	0.304901
2d-5c-125m	0.848217	0.779457	0.742248	0.77941
2d-5c-175m	0.903762	0.99959	0.99959	1
2d-5c-75m	0.945857	0.892976	0.897833	0.934184
5d-10c-125m	1	0.989513	0.979026	0.979026
5d-10c-175m	1	1	1	1
5d-10c-75m	1	1	1	1
5d-20c-125m	0.957281	0.835411	0.874045	0.863337
5d-20c-175m	0.964107	0.79065	0.834896	0.872679
5d-20c-75m	0.951824	0.801694	0.807488	0.818105
5d-2c-125m	1	1	1	1
5d-2c-175m	1	1	1	1
5d-2c-75m	1	1	1	1
5d-30c-125m	0.931717	0.820496	0.823303	0.845936
5d-30c-175m	0.931432	0.832778	0.835382	0.819607
5d-30c-75m	0.800023	0.766647	0.752958	0.765117
5d-5c-125m	1	0.999265	0.999632	1
5d-5c-175m	1	1	1	1
5d-5c-75m	1	1	1	1

Table C.4: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	0.959869	1	1
100d-10c-175m	1	1	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	0.928775	0.906431	0.901065
100d-20c-175m	1	0.829675	0.925368	0.947971
100d-20c-75m	1	0.940493	0.873284	0.858695
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.859355	0.903213	0.913619
100d-30c-175m	0.98434	0.864511	0.769558	0.875287
100d-30c-75m	0.99324	0.873652	0.854383	0.88547
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	0.953648	0.966506	1
10d-10c-175m	1	1	1	1
10d-10c-75m	1	0.974745	0.924325	0.987231
10d-20c-125m	1	0.861519	0.900418	0.873144
10d-20c-175m	1	0.823823	0.90262	0.888794
10d-20c-75m	1	0.831678	0.901734	0.839989
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.841481	0.807304	0.846795
10d-30c-175m	0.974734	0.851642	0.852478	0.821466
10d-30c-75m	0.995329	0.88057	0.838645	0.850997
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	0.972557	1	0.972557
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.875851	0.917763	0.855614
20d-20c-175m	1	0.877753	0.877341	0.921212
20d-20c-75m	1	0.904848	0.856409	0.832643
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.89187	0.867437	0.914416
20d-30c-175m	0.98898	0.805488	0.859573	0.822127
20d-30c-75m	0.969109	0.818848	0.861198	0.840118
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
24d-10c-125m	0.936128	0.800008	0.79872	0.796618
24d-10c-175m	0.999289	0.998389	0.954358	0.919414
24d-10c-75m	0.915825	0.785405	0.702078	0.740202
24d-20c-125m	0.682375	0.600429	0.616383	0.631549
24d-20c-175m	0.86862	0.78219	0.767678	0.846402
24d-20c-75m	0.604147	0.50056	0.497366	0.303391
24d-2c-125m	1	1	1	1
24d-2c-175m	1	1	1	1
24d-2c-75m	1	1	1	1
24d-30c-125m	0.577244	0.507232	0.504635	0.481325
24d-30c-175m	0.871619	0.696796	0.723537	0.738976
24d-30c-75m	0.332496	0.289713	0.290525	0.23471
24d-5c-125m	0.848217	0.562411	0.5668	0.612397
24d-5c-175m	0.903762	1	0.99959	0.996316
24d-5c-75m	0.945857	0.8698	0.82602	0.869322
54d-10c-125m	1	0.979026	0.979026	0.979026
54d-10c-175m	1	0.973776	1	1
54d-10c-75m	1	0.953186	0.999856	0.999856
54d-20c-125m	0.957281	0.800473	0.788558	0.810076
54d-20c-175m	0.964107	0.773651	0.863037	0.78872
54d-20c-75m	0.951824	0.75763	0.696037	0.762487
54d-2c-125m	1	1	1	1
54d-2c-175m	1	1	1	1
54d-2c-75m	1	1	1	1
54d-30c-125m	0.931717	0.792319	0.854557	0.822887
54d-30c-175m	0.931432	0.795656	0.857524	0.850145
54d-30c-75m	0.800023	0.697908	0.695722	0.655811
54d-5c-125m	1	1	1	1
54d-5c-175m	1	1	1	1
54d-5c-75m	1	1	1	1

Table C.5: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	0.951432	0.947542	0.967008
100d-10c-175m	1	0.958175	1	1
100d-10c-75m	1	1	1	1
100d-20c-125m	1	0.857074	0.830399	0.845191
100d-20c-175m	1	0.875215	0.88828	0.864638
100d-20c-75m	1	0.809215	0.87241	0.873323
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.878274	0.854232	0.825484
100d-30c-175m	0.98434	0.820753	0.829688	0.826137
100d-30c-75m	0.99324	0.857398	0.874923	0.853849
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	0.900388	0.922738	0.885942
10d-10c-175m	1	0.969947	1	1
10d-10c-75m	1	0.973801	0.947603	0.947603
10d-20c-125m	1	0.851924	0.724623	0.897852
10d-20c-175m	1	0.84228	0.868311	0.866125
10d-20c-75m	1	0.770166	0.861098	0.818068
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.817217	0.803329	0.804486
10d-30c-175m	0.974734	0.826137	0.791822	0.804194
10d-30c-75m	0.995329	0.806523	0.780593	0.796553
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	0.99126	1
20d-10c-175m	1	1	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.887064	0.87613	0.883395
20d-20c-175m	1	0.779252	0.827585	0.864232
20d-20c-75m	1	0.905768	0.865978	0.837847
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.817465	0.790811	0.847647
20d-30c-175m	0.98898	0.786671	0.786773	0.847905
20d-30c-75m	0.969109	0.7649	0.847606	0.814137
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
2d-10c-125m	0.936128	0.800857	0.756051	0.999663
2d-10c-175m	0.999289	0.789921	0.909427	0.909235
2d-10c-75m	0.915825	0.623962	0.667426	0.816879
2d-20c-125m	0.682375	0.331458	0.573832	0.468896
2d-20c-175m	0.86862	0.720956	0.76242	0.704266
2d-20c-75m	0.604147	0.458561	0.255112	0.388145
2d-2c-125m	1	1	1	1
2d-2c-175m	1	1	1	1
2d-2c-75m	1	1	1	1
2d-30c-125m	0.577244	0.360215	0.467805	0.344669
2d-30c-175m	0.871619	0.746536	0.698924	0.758415
2d-30c-75m	0.332496	0.10984	0.26594	0.110352
2d-5c-125m	0.848217	0.470948	0.470948	0.607653
2d-5c-175m	0.903762	0.924387	0.99959	0.99959
2d-5c-75m	0.945857	0.937827	0.831084	0.785628
5d-10c-125m	1	0.919277	0.979026	0.979026
5d-10c-175m	1	0.899625	1	0.999785
5d-10c-75m	1	1	1	1
5d-20c-125m	0.957281	0.746024	0.788473	0.772002
5d-20c-175m	0.964107	0.813179	0.751586	0.791327
5d-20c-75m	0.951824	0.74614	0.753169	0.776872
5d-2c-125m	1	1	1	1
5d-2c-175m	1	1	1	1
5d-2c-75m	1	1	1	1
5d-30c-125m	0.931717	0.722101	0.784979	0.749243
5d-30c-175m	0.931432	0.699489	0.75897	0.769796
5d-30c-75m	0.800023	0.663812	0.621922	0.691766
5d-5c-125m	1	1	0.999632	0.999632
5d-5c-175m	1	1	1	1
5d-5c-75m	1	1	1	1

Table C.6: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	0.973771	0.973771	0.911565
100d-10c-175m	1	1	1	1
100d-10c-75m	1	0.955079	1	1
100d-20c-125m	1	0.808269	0.864117	0.819311
100d-20c-175m	1	0.744573	0.810391	0.894141
100d-20c-75m	1	0.90515	0.86005	0.890362
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.884117	0.849839	0.8043
100d-30c-175m	0.98434	0.786123	0.788126	0.834762
100d-30c-75m	0.99324	0.848573	0.85943	0.848695
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	0.793816	0.91826	0.842893
10d-10c-175m	1	1	1	1
10d-10c-75m	1	1	0.973801	0.927268
10d-20c-125m	1	0.870618	0.83285	0.814445
10d-20c-175m	1	0.859004	0.813124	0.877884
10d-20c-75m	1	0.73608	0.797079	0.770016
10d-2c-125m	1	1	1	1
10d-2c-175m	1	1	0.950474	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.72752	0.769756	0.768803
10d-30c-175m	0.974734	0.792436	0.7788	0.761752
10d-30c-75m	0.995329	0.794276	0.781461	0.814719
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	0.968042	0.972557	0.96303
20d-10c-175m	1	0.893394	1	1
20d-10c-75m	1	1	1	1
20d-20c-125m	1	0.739728	0.899922	0.808557
20d-20c-175m	1	0.88408	0.79367	0.789704
20d-20c-75m	1	0.788623	0.87668	0.788784
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.764842	0.803608	0.838036
20d-30c-175m	0.98898	0.797856	0.840388	0.797142
20d-30c-75m	0.969109	0.834868	0.82145	0.810854
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
24d-10c-125m	0.936128	0.829883	0.652578	0.595191
24d-10c-175m	0.999289	0.708658	0.954411	0.953637
24d-10c-75m	0.915825	0.528818	0.718008	0.676388
24d-20c-125m	0.682375	0.336036	0.471236	0.335662
24d-20c-175m	0.86862	0.712203	0.725294	0.701639
24d-20c-75m	0.604147	0.312118	0.270394	0.23192
24d-2c-125m	1	1	1	1
24d-2c-175m	1	1	1	1
24d-2c-75m	1	1	0.951641	1
24d-30c-125m	0.577244	0.28981	0.297364	0.158326
24d-30c-175m	0.871619	0.482784	0.664666	0.753932
24d-30c-75m	0.332496	0.150432	0.153745	0.110457
24d-5c-125m	0.848217	0.470948	0.715099	0.547362
24d-5c-175m	0.903762	1	0.99918	0.99918
24d-5c-75m	0.945857	0.72959	0.79998	0.679566
54d-10c-125m	1	0.989513	0.947199	0.979026
54d-10c-175m	1	0.90559	1	1
54d-10c-75m	1	1	1	0.999283
54d-20c-125m	0.957281	0.751405	0.812118	0.807155
54d-20c-175m	0.964107	0.637307	0.775012	0.752374
54d-20c-75m	0.951824	0.714042	0.789072	0.644705
54d-2c-125m	1	1	1	1
54d-2c-175m	1	1	1	1
54d-2c-75m	1	1	1	1
54d-30c-125m	0.931717	0.69295	0.760975	0.677954
54d-30c-175m	0.931432	0.780114	0.720846	0.764217
54d-30c-75m	0.800023	0.520839	0.583101	0.597443
54d-5c-125m	1	0.999266	1	0.999632
54d-5c-175m	1	1	1	1
54d-5c-75m	1	1	1	1

Table C.7: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1	0.973771	0.951008	0.947542
100d-10c-175m	1	0.814726	1	1
100d-10c-75m	1	0.96382	1	1
100d-20c-125m	1	0.839863	0.842878	0.785633
100d-20c-175m	1	0.888867	0.807113	0.843093
100d-20c-75m	1	0.906216	0.867072	0.916818
100d-2c-125m	1	1	1	1
100d-2c-175m	1	1	1	1
100d-2c-75m	1	1	1	1
100d-30c-125m	1	0.819122	0.823432	0.838605
100d-30c-175m	0.98434	0.826226	0.847527	0.859163
100d-30c-75m	0.99324	0.829855	0.838975	0.828504
100d-5c-125m	1	1	1	1
100d-5c-175m	1	1	1	1
100d-5c-75m	1	1	1	1
10d-10c-125m	1	0.854642	0.973306	0.660871
10d-10c-175m	1	0.868592	1	1
10d-10c-75m	1	0.974745	0.947603	0.924325
10d-20c-125m	1	0.781469	0.574164	0.81239
10d-20c-175m	1	0.663816	0.756925	0.83017
10d-20c-75m	1	0.735528	0.820625	0.834225
10d-2c-125m	1	1	1	1
10d-2c-175m	1	0.958044	1	1
10d-2c-75m	1	1	1	1
10d-30c-125m	0.978499	0.772943	0.801114	0.733399
10d-30c-175m	0.974734	0.730697	0.81328	0.760564
10d-30c-75m	0.995329	0.719764	0.687868	0.751572
10d-5c-125m	1	1	1	1
10d-5c-175m	1	1	1	1
10d-5c-75m	1	1	1	1
20d-10c-125m	1	1	1	0.997742
20d-10c-175m	1	0.96804	1	1
20d-10c-75m	1	0.972035	1	1
20d-20c-125m	1	0.7653	0.762889	0.870891
20d-20c-175m	1	0.778274	0.855162	0.802727
20d-20c-75m	1	0.863712	0.837033	0.793203
20d-2c-125m	1	1	1	1
20d-2c-175m	1	1	1	1
20d-2c-75m	1	1	1	1
20d-30c-125m	1	0.801076	0.819519	0.735705
20d-30c-175m	0.98898	0.747147	0.808692	0.901488
20d-30c-75m	0.969109	0.701377	0.82659	0.810264
20d-5c-125m	1	1	1	1
20d-5c-175m	1	1	1	1
20d-5c-75m	1	1	1	1
2d-10c-125m	0.936128	0.34327	0.659002	0.968064
2d-10c-175m	0.999289	0.722358	0.975288	0.997835
2d-10c-75m	0.915825	0.654511	0.701008	0.652774
2d-20c-125m	0.682375	0.240801	0.308434	0.433052
2d-20c-175m	0.86862	0.652216	0.695548	0.464205
2d-20c-75m	0.604147	0.164986	0.170651	0.268146
2d-2c-125m	1	1	1	1
2d-2c-175m	1	1	1	1
2d-2c-75m	1	1	0.954071	1
2d-30c-125m	0.577244	0.088834	0.151972	0.066178
2d-30c-175m	0.871619	0.711369	0.288964	0.678213
2d-30c-75m	0.332496	0.165479	0.157323	0.110682
2d-5c-125m	0.848217	0.827933	0.470948	0.639763
2d-5c-175m	0.903762	1	1	1
2d-5c-75m	0.945857	0.863767	0.734312	0.675427
5d-10c-125m	1	0.979026	0.979026	0.979026
5d-10c-175m	1	1	1	1
5d-10c-75m	1	0.94563	0.998712	0.999856
5d-20c-125m	0.957281	0.56134	0.68508	0.736406
5d-20c-175m	0.964107	0.81228	0.817623	0.755156
5d-20c-75m	0.951824	0.518766	0.703498	0.553438
5d-2c-125m	1	1	1	1
5d-2c-175m	1	1	1	1
5d-2c-75m	1	1	1	1
5d-30c-125m	0.931717	0.682177	0.56904	0.781469
5d-30c-175m	0.931432	0.794281	0.720387	0.77926
5d-30c-75m	0.800023	0.381251	0.568165	0.526059
5d-5c-125m	1	0.999632	1	0.999632
5d-5c-175m	1	1	1	1
5d-5c-75m	1	1	1	1

Table C.8: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the artificial data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.170184	0.191761	0.147913
biopn	0.228691	0.225856	0.246393	0.238311
bpa	0.00886	0.008151	0.005557	0.010093
dermatology	0.868666	0.861777	0.877633	0.867854
ecoli	0.754064	0.733693	0.733413	0.740159
glass	0.253647	0.247328	0.230173	0.229499
heart-statlog	0.187839	0.258794	0.218429	0.187524
iono	0.319639	0.306377	0.291283	0.302819
iris	0.739415	0.712582	0.739875	0.714126
letter-rec	0.153947	0.151512	0.149409	0.14179
liver-disorders	0.010528	0.00742	0.006713	0.00624
magic	0.242654	0.227498	0.229073	0.234709
pendigits	0.646557	0.601473	0.64039	0.644326
pim	0.143558	0.134446	0.146012	0.145649
segment	0.558108	0.543899	0.53961	0.542184
shuttle	0.9532	0.9478	0.9405	0.9552
sonar	0.042394	0.042405	0.045971	0.050484
thyroids	0.86236	0.85324	0.85543	0.84643
transfusion	0.035632	0.035985	0.041976	0.036339
vehicle	0.151451	0.14764	0.129961	0.141306
waveform	0.29229	0.291241	0.281596	0.282499
wdbc	0.747802	0.751763	0.770398	0.737767
wisc	0.862961	0.86137	0.841788	0.836444
wpbc	0.063228	0.038296	0.077662	0.071418
yeast	0.230616	0.19616	0.210303	0.229813

Table C.9: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.173564	0.2061	0.163074
biopn	0.228691	0.232427	0.248723	0.235378
bpa	0.00886	0.011529	0.006872	0.007274
dermatology	0.868666	0.852928	0.849655	0.848042
ecoli	0.754064	0.723611	0.733586	0.721277
glass	0.253647	0.245891	0.237654	0.225512
heart-statlog	0.187839	0.224345	0.211188	0.202915
iono	0.319639	0.274385	0.293711	0.265053
iris	0.739415	0.77205	0.692838	0.642804
letter-rec	0.153947	0.143493	0.138433	0.143343
liver-disorders	0.010528	0.007935	0.009469	0.008328
magic	0.242654	0.153703	0.21039	0.21876
pendigits	0.646557	0.57296	0.607463	0.610735
pim	0.143558	0.130584	0.14724	0.154953
segment	0.558108	0.524637	0.544095	0.528213
shuttle	0.9532	0.9348	0.93905	0.9465
sonar	0.042394	0.043503	0.033243	0.045092
thyroids	0.86236	0.84654	0.85013	0.84639
transfusion	0.035632	0.029011	0.034213	0.035059
vehicle	0.151451	0.140457	0.131127	0.128919
waveform	0.29229	0.281827	0.272509	0.286367
wdbc	0.747802	0.744309	0.747156	0.755615
wisc	0.862961	0.85388	0.848428	0.848514
wpbc	0.063228	0.069468	0.05665	0.053381
yeast	0.230616	0.210746	0.202226	0.214392

Table C.10: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.198387	0.19152	0.201456
biopn	0.228691	0.21182	0.216568	0.238036
bpa	0.00886	0.009984	0.006675	0.009954
dermatology	0.868666	0.861322	0.846462	0.858958
ecoli	0.754064	0.714993	0.717121	0.708736
glass	0.253647	0.22917	0.250542	0.226896
heart-statlog	0.187839	0.213877	0.238694	0.24688
iono	0.319639	0.245936	0.246404	0.274515
iris	0.739415	0.581408	0.596254	0.619627
letter-rec	0.153947	0.146506	0.143031	0.138365
liver-disorders	0.010528	0.008927	0.011359	0.010127
magic	0.242654	0.175796	0.204143	0.219585
pendigits	0.646557	0.593317	0.574352	0.584477
pim	0.143558	0.127045	0.138804	0.146952
segment	0.558108	0.546472	0.522968	0.526832
shuttle	0.9532	0.9334	0.93643	0.9418
sonar	0.042394	0.047564	0.039228	0.047014
thyroids	0.86236	0.84542	0.84932	0.84304
transfusion	0.035632	0.036736	0.031494	0.034981
vehicle	0.151451	0.124623	0.121382	0.116318
waveform	0.29229	0.286281	0.268734	0.268114
wdbc	0.747802	0.748826	0.747223	0.754555
wisc	0.862961	0.852518	0.849764	0.864681
wpbc	0.063228	0.078822	0.059537	0.051031
yeast	0.230616	0.212745	0.210746	0.201102

Table C.11: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.230929	0.182584	0.169737
biopn	0.228691	0.24328	0.238485	0.226422
bpa	0.00886	0.005166	0.012797	0.008819
dermatology	0.868666	0.85129	0.833745	0.861698
ecoli	0.754064	0.680879	0.748392	0.72151
glass	0.253647	0.230405	0.22622	0.23083
heart-statlog	0.187839	0.232241	0.256431	0.238513
iono	0.319639	0.248593	0.2462	0.226341
iris	0.739415	0.613851	0.560577	0.651074
letter-rec	0.153947	0.133718	0.140734	0.138013
liver-disorders	0.010528	0.009409	0.006155	0.006822
magic	0.242654	0.219238	0.219868	0.176055
pendigits	0.646557	0.570577	0.557386	0.586625
pim	0.143558	0.142196	0.151112	0.12041
segment	0.558108	0.532232	0.528135	0.503506
shuttle	0.9532	0.9343	0.93125	0.9388
sonar	0.042394	0.055839	0.049859	0.025033
thyroids	0.86236	0.84324	0.84022	0.85065
transfusion	0.035632	0.031366	0.039579	0.030752
vehicle	0.151451	0.118892	0.116352	0.125853
waveform	0.29229	0.2636	0.27367	0.273869
wdbc	0.747802	0.745988	0.749676	0.747034
wisc	0.862961	0.855202	0.84451	0.842621
wdbc	0.063228	0.048264	0.076235	0.056465
yeast	0.230616	0.204262	0.205886	0.182761

Table C.12: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.21657	0.155011	0.163518
biopn	0.228691	0.204295	0.241116	0.222368
bpa	0.00886	0.005387	0.014446	0.008497
dermatology	0.868666	0.777998	0.851146	0.818189
ecoli	0.754064	0.610386	0.588144	0.711106
glass	0.253647	0.196908	0.203916	0.223896
heart-statlog	0.187839	0.261562	0.227061	0.249852
iono	0.319639	0.193485	0.197698	0.230329
iris	0.739415	0.652813	0.722269	0.689663
letter-rec	0.153947	0.132775	0.147375	0.118269
liver-disorders	0.010528	0.004201	0.007381	0.008253
magic	0.242654	0.209066	0.149577	0.208855
pendigits	0.646557	0.539073	0.525887	0.51202
pim	0.143558	0.128428	0.113906	0.154114
segment	0.558108	0.465803	0.50797	0.519026
shuttle	0.9532	0.848675	0.834112	0.934936
sonar	0.042394	0.032043	0.010332	0.023785
thyroids	0.86236	0.83724	0.83052	0.84125
transfusion	0.035632	0.026026	0.029918	0.030742
vehicle	0.151451	0.10367	0.114436	0.117382
waveform	0.29229	0.261993	0.256652	0.264992
wdbc	0.747802	0.757769	0.719833	0.732497
wisc	0.862961	0.849851	0.840453	0.840955
wdbc	0.063228	0.060037	0.041683	0.047806
yeast	0.230616	0.162768	0.183443	0.186938

Table C.13: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.175567	0.194547	0.158817
biopn	0.228691	0.1988	0.230809	0.198184
bpa	0.00886	0.004922	0.007299	0.012049
dermatology	0.868666	0.653115	0.753512	0.749018
ecoli	0.754064	0.605827	0.610043	0.680998
glass	0.253647	0.216515	0.193827	0.216097
heart-statlog	0.187839	0.33215	0.184487	0.230779
iono	0.319639	0.214389	0.20497	0.195827
iris	0.739415	0.639524	0.753638	0.657326
letter-rec	0.153947	0.003834	0.145676	0.124976
liver-disorders	0.010528	0.00461	0.008176	0.007522
magic	0.242654	0.224192	0.191248	0.184522
pendigits	0.646557	0.428299	0.489226	0.519403
pim	0.143558	0.145378	0.122708	0.128502
segment	0.558108	0.44972	0.478776	0.399778
shuttle	0.9532	0.898803	0.739589	0.850929
sonar	0.042394	0.029049	0.023422	0.014883
thyroids	0.86236	0.83324	0.82242	0.83455
transfusion	0.035632	0.014623	0.021427	0.018143
vehicle	0.151451	0.108886	0.099003	0.110458
waveform	0.29229	0.285669	0.268596	0.275808
wdbc	0.747802	0.689579	0.745695	0.729296
wisc	0.862961	0.843146	0.836508	0.832504
wdbc	0.063228	0.029833	0.022989	0.04984
yeast	0.230616	0.160226	0.174285	0.14473

Table C.14: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.154863	0.231951	0.147189
biopn	0.228691	0.202329	0.199993	0.184874
bpa	0.00886	0.007693	0.012113	0.00614
dermatology	0.868666	0.590938	0.72289	0.594833
ecoli	0.754064	0.610877	0.590481	0.654161
glass	0.253647	0.158607	0.212602	0.213282
heart-statlog	0.187839	0.265984	0.324906	0.261358
iono	0.319639	0.102211	0.137872	0.168257
iris	0.739415	0.714652	0.786418	0.673837
letter-rec	0.153947	0.140349	0.128098	0.124728
liver-disorders	0.010528	0.005562	0.01622	0.008929
magic	0.242654	0.157435	0.198982	0.060581
pendigits	0.646557	0.408564	0.50261	0.29192
pim	0.143558	0.137902	0.135579	0.105845
segment	0.558108	0.447449	0.454806	0.365357
shuttle	0.9532	0.829486	0.831158	0.845841
sonar	0.042394	0.023046	0.040229	0.028843
thyroids	0.86236	0.82424	0.82542	0.82942
transfusion	0.035632	0.028032	0.038525	0.018249
vehicle	0.151451	0.088872	0.105795	0.110046
waveform	0.29229	0.271414	0.216106	0.269825
wdbc	0.747802	0.717615	0.681995	0.687261
wise	0.862961	0.805533	0.819212	0.833882
wpbc	0.063228	0.059904	0.026871	0.058086
yeast	0.230616	0.126783	0.18392	0.143259

Table C.15: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the real-world data sets.

Data set	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.169766	0.174458	0.17318	0.157827
biopn	0.228691	0.142486	0.204046	0.219347
bpa	0.00886	0.007514	0.012995	0.007088
dermatology	0.868666	0.620453	0.694847	0.636374
ecoli	0.754064	0.662275	0.634537	0.640415
glass	0.253647	0.223801	0.200609	0.187938
heart-statlog	0.187839	0.357584	0.294774	0.331949
iono	0.319639	0.14611	0.109221	0.16226
iris	0.739415	0.75132	0.802067	0.799433
letter-rec	0.153947	0.131896	0.142669	0.035788
liver-disorders	0.010528	0.004442	0.008581	0.010826
magic	0.242654	0.143388	0.203973	0.105898
pendigits	0.646557	0.332899	0.363517	0.319356
pim	0.143558	0.135755	0.140383	0.140721
segment	0.558108	0.227719	0.329282	0.217131
shuttle	0.9532	0.840712	0.827118	0.80408
sonar	0.042394	0.021933	0.027219	0.06105
thyroids	0.86236	0.82043	0.82012	0.82763
transfusion	0.035632	0.028032	0.038525	0.018249
vehicle	0.151451	0.116024	0.094734	0.103232
waveform	0.29229	0.174416	0.223408	0.249784
wdbc	0.747802	0.706732	0.669422	0.737754
wise	0.862961	0.8153	0.821387	0.813828
wpbc	0.063228	0.043786	0.025294	0.068049
yeast	0.230616	0.09553	0.137486	0.140955

Table C.16: Accuracy results of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the real-world data sets.

Data set	Precalculating time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters
1004-10c-125m	1559.47	419.13	420.23	68257.9	15811.3	17711.3	69817.37	18131.53	16863.3
1004-10c-175m	1518.51	401.95	401.43	68939.9	15600.2	18713.9	70458.41	16002.15	23008.46
1004-10c-75m	1588.06	455.05	454.68	67134.5	14713.6	14866.2	68722.86	15108.65	21281.13
1004-20c-125m	1460.35	410.63	414.43	197312.65	59297.5	77418.1	198772.8	57081.3	54253.44
1004-20c-175m	1118.83	389.12	389.67	182929.23	58410.7	65342	184088.06	58799.82	61041.85
1004-20c-75m	1661.64	467.85	467.32	219990.83	61582.4	38138.6	221652.47	62050.25	58605.97
1004-2c-125m	34.94	10.46	10.46	2558.29	600.12	505.4	2993.23	610.58	42631.22
1004-2c-175m	38.23	11.41	11.42	2695.53	549.61	541.61	2733.76	561.02	468.13
1004-2c-75m	84.08	25.91	25.91	4099.47	1111.74	1062.55	4993.55	1137.65	553.03
1004-30c-125m	5760.12	1701.99	1702.43	236056.8	90142.24	57034.23	241816.92	1137.65	1088.46
1004-30c-175m	5240.54	2542.78	2542.78	251745.12	131564.64	139273.48	256985.66	134107.09	63984.31
1004-30c-75m	5430.69	2686.75	2686.75	252835.54	85600.12	81528.94	258206.23	83923.83	121213.27
1004-5c-125m	240.67	70.64	70.58	16063.8	3683	2878.09	16304.47	3753.64	84514.15
1004-5c-175m	416.16	131.65	131.29	18648.5	5104.41	3961.45	19064.66	5236.06	10378.17
1004-5c-75m	318.86	90.51	91.06	18722.1	4598.45	4375.58	19040.96	4466.64	9150.66
104-10c-125m	319.88	86.54	86.31	11171.9	2697.3	2425.74	11491.78	2512.02	3659.98
104-10c-175m	277.83	73.11	72.99	10057.2	2538.96	2268.58	10335.03	2612.07	3169.69
104-10c-75m	220.74	58.88	58.74	9078.58	1966.8	2114.37	9299.32	2025.68	2428.96
104-20c-125m	1253.22	368.53	367.84	32143.53	8025.92	8508.79	33396.75	8394.45	1714.95
104-20c-175m	1287.38	380.43	378.53	42235.92	12647.2	8917.51	43523.3	13027.63	8876.63
104-20c-75m	1495.49	437.05	436.24	41602.64	12658.1	13034.4	43098.13	13095.15	10378.17
104-2c-125m	7.89	2.05	2.05	462.35	90.64	90.21	470.24	92.26	24646.82
104-2c-175m	5.42	1.44	1.44	349.87	66.07	66.95	335.29	67.51	2646.82
104-2c-75m	2.82	0.83	0.84	174.58	36	40.15	177.4	36.83	61.01
104-30c-125m	5320.54	1439.43	1439.43	107058.22	26769.51	24636.86	112378.76	28209.63	30068.87
104-30c-175m	5034.86	1345.32	1346.46	24693.73	24693.73	27603.54	109718.28	26309.05	28915.81
104-30c-75m	5128.76	1403.54	1402.54	103793.54	25948.38	25820.42	108922.3	27351.92	30468.78
104-5c-125m	46.06	13.72	13.69	1662.31	381.96	449.12	1708.37	395.68	706.84
104-5c-175m	53.51	15.03	15.01	1964.03	662.64	429.9	2071.54	677.67	444.91
104-5c-75m	61.16	17.14	17.14	2572.01	845.47	757.62	2633.7	562.61	530.42
204-10c-125m	2733.37	773.77	771	12453.1	3668.01	2575.03	12726.47	3745.2	599.82
204-10c-175m	405.14	123.98	122.51	15488.5	7098.95	4026.06	15893.64	2652.13	2500.99
204-10c-75m	344.74	100.03	99.95	12892.2	3653.7	4237.3	13236.94	3753.73	4748.42
204-20c-125m	2012.43	671.93	670.08	84354	24059.6	18522.6	86366.43	24731.53	3692.83
204-20c-175m	2069.92	631.45	628.93	68349	23921.7	19621	77776.92	23553.15	24646.82
204-20c-75m	1956.12	574.84	573.07	61646.6	16419.3	2238.2	63602.72	16994.14	22967.15
204-2c-125m	19.71	5.46	5.53	1137.58	235.96	203.68	1157.29	241.42	227.5
204-2c-175m	18.68	5.48	5.48	1073.98	303.99	282.28	1092.66	209.1	288.43
204-2c-75m	20.38	6.82	6.82	1177.48	240.07	218.5	1197.86	246.89	280.67
204-30c-125m	5370.46	1472.56	1472.98	164464.42	26810.83	25425.13	169834.88	26898.11	28022.26
204-30c-175m	5434.12	1495.97	1496.12	125117.47	30548.62	27534.43	13051.59	29030.55	26380.98
204-30c-75m	5228.25	1398.08	1397.87	146856.93	26478.67	25035.18	152085.18	31876.75	31129.72
204-5c-125m	130.38	41.27	41.23	4930.26	1419.34	2287.56	5060.64	1460.61	1666.44
204-5c-175m	71.12	19.69	19.6	4713.83	909.71	944.05	4790.95	929.4	997.86
204-5c-75m	91.44	29.63	29.77	4612.69	1246.62	1440	4704.13	1276.25	836.74
2d-10c-125m	154.95	41.52	41.41	2991.08	659.42	485.71	3146.63	700.94	559.05
2d-10c-175m	127.46	35.08	34.87	2795.27	584.58	528.02	3146.63	619.66	566.51
2d-10c-75m	140.54	37.03	36.9	2885.6	510.85	539.59	3026.14	547.88	580.47
2d-20c-125m	840.015	386.01	383.77	16254.4	2884.74	3386.88	17094.415	3270.75	3322.86
2d-20c-175m	1113.04	228.2	228.2	11162.4	1960.58	2122.84	12275.44	2188.78	2689.67
2d-20c-75m	916.765	317.12	316.01	16273.45	2565.16	3458.24	17190.215	2882.28	3139.54
2d-2c-125m	3.49	0.91	0.91	159.25	23.95	28.96	162.74	24.86	26.63
2d-2c-175m	2.36	0.64	0.65	120.01	20.6	21.84	122.37	21.24	23.25
2d-2c-75m	3.79	1.01	1.01	186.59	33.44	32.35	190.38	22.49	23.25
2d-30c-125m	5124.43	1298.54	1297.98	26810.85	27423.42	27423.42	34276.94	28833.12	28830.19
2d-30c-175m	4987.26	1249.07	1249.78	27043.15	26548.68	26422.24	32030.41	27797.75	26774.94
2d-30c-75m	5043.14	1264.21	1263.94	25036.34	30478.64	29843.21	30079.48	31742.85	31643.28
2d-5c-125m	27.86	7.08	7.05	785.2	156.42	158.33	813.06	163.58	158.42
2d-5c-175m	27.82	7.09	7.09	747.54	160.27	138.74	775.36	167.36	146.9
2d-5c-75m	44.08	11.14	11.14	1272.25	247.15	258.68	1316.33	258.34	232.89
5d-10c-125m	204.3	51.23	51.29	5864.02	1402.39	1038.2	6068.32	1453.62	1353.41
5d-10c-175m	196.31	51.66	51.41	6126.79	1396.54	1058.1	6323.1	1448.2	1114.18
5d-10c-75m	265.21	70.58	70.45	6718.24	1400	1604.93	6983.45	1470.58	1675.38
5d-20c-125m	1086.36	328.99	327	30277	6397.02	4821.84	21563.36	6726.01	5161.21
5d-20c-175m	1591.11	409.95	408.56	27562.3	6104.55	5654.82	29153.4	6514.5	6063.38
5d-20c-75m	1892.81	256.05	255.4	3474.29	4877.34	4474.85	3730.34	3730.34	5147.16
5d-2c-125m	6.6	1.77	1.77	324.86	68.51	65.9	331.46	70.28	68.09
5d-2c-175m	2.58	0.69	0.68	138.95	21.96	28.86	141.53	22.65	30.25
5d-2c-75m	4.97	1.19	1.2	253.73	43.74	47.85	258.7	44.93	37.6
5d-30c-125m	5146.73	1292.05	1292.46	68901.43	10227.31	8643.42	74048.16	11519.36	9935.88
5d-30c-175m	5032.54	1265.45	1262.84	71943.24	5326.55	7533.63	76975.78	6590	11077.32
5d-30c-75m	5138.96	1290.55	1290.78	69875.32	9069.61	8325.12	75014.28	10600.16	9615.9
5d-5c-125m	28.3	7.66	7.64	1109.03	273.35	243.09	1137.33	281.01	1022.75
5d-5c-175m	41.08	10.41	10.42	1479.01	329.09	338.32	1520.09	339.5	280.61
5d-5c-75m	30.56	7.89	7.87	1185.01	219.19	253.5	1215.57	227.08	241.74

Table C.17: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the artificial data sets.

Data set	Precalculation time				Clustering time				Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1589.47	195.01	194.19	230.99	6857.9	1209.9	593.47	1269.4	6917.37	1204.91	6137.66	1290.39
100d-10c-175m	1518.51	203.02	202.66	239.99	6899.9	639.33	787.68	5618.96	7048.41	659.35	588.95	6769.99
100d-10c-75m	1588.36	219.91	219.91	260.12	6714.5	5861.26	9205.41	6509.87	6872.86	6081.21	9425.32	588.95
100d-20c-125m	1460.05	79.93	778.51	929.59	19313.65	24130.6	17810.4	27010.8	19877.2	24909.93	27940.39	31172.7
100d-20c-175m	1118.83	802.27	798.6	944.18	182929.23	29355.5	30374.1	22428.4	184048.06	31172.7	23375.88	31172.7
100d-20c-75m	1661.64	823.69	825.47	971.99	219990.83	34678.8	32329.7	26764.6	221652.47	33592.49	27756.19	31551.17
100d-3c-125m	34.94	5.1	5.1	6.04	2588.29	232.27	191.96	231.25	22593.23	227.37	297.06	297.06
100d-3c-175m	38.23	5.01	5	5.91	2695.53	222.63	199.76	211.56	2733.76	227.44	204.76	217.47
100d-2c-75m	84.08	11.94	11.97	14.21	4909.47	411.08	458.46	471.37	4993.53	423.02	470.43	483.58
100d-30c-125m	7760.12	706.79	774.81	927.33	236056.8	30720.7	38156.14	35872.2	241816.62	31497.74	38909.93	36799.53
100d-30c-175m	5240.54	2020.63	2020.98	2453.89	251745.12	134466	73888.8	65240.5	226893.96	136482.63	78979.78	78979.78
100d-30c-75m	5430.69	2003.87	2006.07	2385.64	252383.54	66036.9	67109.9	134418.54	252652.23	66040.77	69115.97	118040.18
100d-5c-125m	240.67	33.39	33.45	40.97	16063.8	966.88	1288.55	1069.48	16304.47	1000.27	1000.27	1000.27
100d-5c-175m	416.16	57.64	57.68	68.53	18684.5	1741.96	2628.83	2205.8	19064.66	1799.6	1799.6	2686.53
100d-5c-75m	318.86	43.96	43.94	52.95	18722.1	1507.17	1646.15	1731.34	19040.96	1591.13	1591.13	1690.09
100d-10c-125m	319.88	39.99	39.92	48.76	11171.9	892.21	872.48	799.48	11491.78	932.2	912.4	912.4
100d-10c-175m	277.83	34.37	34.27	41.45	10067.2	1016.77	834.2	1798.21	10335.03	1051.14	1051.14	888.47
100d-10c-75m	220.74	28.06	28.0	34.09	9078.58	742.25	830.93	635.96	9299.32	770.31	770.31	888.47
100d-20c-125m	1253.22	175.48	175.47	199.47	32143.53	4088.26	4260.99	3111.17	33306.75	4233.74	4233.74	4436.46
100d-20c-175m	1287.38	172.74	171.96	197.7	42233.92	3440.64	4363.46	4328.89	43323.3	3613.38	4535.42	4536.59
100d-20c-75m	1495.49	198.64	198.52	228.2	41602.64	3239.78	3221.09	3978.12	43098.13	3438.42	3438.42	3419.61
100d-2c-125m	7.89	0.98	0.99	1.36	462.38	26.83	28.54	31.58	470.24	27.83	27.83	29.53
100d-2c-175m	5.42	0.63	0.63	1.36	349.87	17.85	26.28	24.21	355.29	18.48	26.91	25.06
100d-2c-75m	2.82	0.38	0.38	0.52	174.38	12.47	12.07	13.02	177.4	12.85	12.45	13.54
100d-30c-125m	5320.54	543.81	540.84	604.32	107058.22	17542.5	11743.8	14310.8	11278.76	18086.31	12384.64	32306.06
100d-30c-175m	5024.86	686.01	683.86	755.64	106683.42	11698.5	31622.2	15082.3	109718.28	12384.51	14915.12	15873.94
100d-30c-75m	5128.76	679.52	676.84	748.6	103793.54	22609.2	10780.4	13448.6	108718.28	22388.72	11457.24	15873.94
100d-5c-125m	46.06	6.6	6.61	8.34	1662.31	181.75	215.77	204.46	1708.37	188.33	222.8	212.8
100d-5c-175m	53.51	7.97	7.98	10.36	7.97	175.35	257.05	225.52	2017.54	183.32	235.88	265.03
100d-5c-75m	61.16	8.17	8.18	10.14	2572.01	277.88	218.69	210.68	2633.17	286.05	226.87	220.82
200d-10c-125m	273.37	36.99	37.16	47.37	12453.1	2104.47	1691.18	998.9	12726.67	2141.46	1728.34	1046.27
200d-10c-175m	405.14	51.55	51.39	65.58	15488.5	2464.59	2419.71	2545.97	15893.64	2515.14	2471.1	2611.55
200d-10c-75m	344.74	48.23	48.2	61.41	12892.2	1825.84	1311.89	1761.93	13236.94	1837.07	1860.09	1832.34
200d-20c-125m	2012.43	317.8	317.62	370.07	84354	7323.47	15350.4	10878.3	86366.43	7680.27	15848.02	11248.37
200d-20c-175m	2069.92	285.75	284.76	339.37	75707	7303.37	15772.2	13016.5	77776.96	7589.12	16568.96	13355.87
200d-20c-75m	1956.12	271.31	270.99	327.58	61646.6	5722.55	9803.38	7215.41	65602.72	7953.86	10074.37	10074.37
200d-2c-125m	19.71	2.83	2.83	3.6	1137.58	111.51	105.8	90.79	1157.29	114.34	114.34	108.63
200d-2c-175m	18.68	2.44	2.44	3.12	1073.98	103.74	103.74	95.91	1097.66	102.18	98.35	98.35
200d-2c-75m	20.38	2.64	2.64	3.58	1177.48	90.39	93.68	110.67	1197.86	106.33	96.32	96.32
200d-30c-125m	5370.46	657.56	655.61	751.81	164464.42	19302.3	21078.8	17388.3	169834.88	20159.86	21734.41	18401.11
200d-30c-175m	5248.12	712.11	711.23	824.08	125117.47	24380.4	28600.1	23584.5	130651.59	23584.51	21312.53	24208.58
200d-30c-75m	5228.25	747.44	744.62	871.37	148686.93	28584.4	28584.4	25751.2	132085.18	29351.84	29351.84	26622.57
200d-5c-125m	130.38	18.05	17.91	23.24	4930.26	489.22	984.38	592.29	5060.64	592.29	506.64	515.53
200d-5c-175m	77.12	9.83	9.84	12.77	4713.83	286.36	372.22	442.59	4790.95	296.19	296.19	382.06
200d-5c-75m	91.44	13.67	13.66	17.34	4612.69	772.23	469.89	520.49	4704.13	470.13	783.9	483.55
2d-10c-125m	154.95	19.46	19.42	23.66	2991.68	184.78	212.19	185.06	3146.63	204.74	200.72	200.72
2d-10c-175m	127.46	15.63	15.63	19.19	2795.27	185.11	254.73	181.04	2922.73	200.78	200.78	200.72
2d-10c-75m	140.54	17.55	17.46	21.44	2885.6	230.34	299.45	202.65	2247.79	247.79	216.91	216.91
2d-20c-125m	840.015	175.63	175.19	196.84	16254.4	1608.14	1098.36	1168.79	17094.415	1783.77	1273.55	1365.63
2d-20c-175m	1113.04	105.65	105.76	120.11	11162.4	810	785.73	744.8	12275.44	915.65	864.91	864.91
2d-20c-75m	916.765	143.38	143.43	162.19	16273.45	1443.02	1597.53	1102.99	17190.215	1586.6	1740.96	1265.18
2d-2c-125m	3.49	0.43	0.43	0.63	159.25	8.04	7.75	9.59	162.74	8.47	8.18	10.22
2d-2c-175m	2.36	0.3	0.3	0.44	120.01	8.09	7.05	5.97	162.74	8.47	8.18	10.22
2d-2c-75m	3.79	0.49	0.49	0.71	186.59	9.62	11.19	11.44	190.38	10.11	11.69	11.69
2d-30c-125m	5121.43	408.08	408.08	467.15	29152.51	23621.1	3014.48	3287.33	34276.94	2770.19	3420.82	3735.24
2d-30c-175m	4987.26	429.05	427.15	468.21	27043.15	2596.16	2917.61	3025.21	32030.41	3025.21	3344.76	3395.46
2d-30c-75m	5003.14	333.32	331.56	369.46	25036.34	1790.14	2138	1922.59	30079.48	2123.46	2469.56	2292.05
2d-3c-125m	27.86	3.24	3.23	4.32	785.2	51.51	52.59	39.73	813.06	54.75	55.82	44.05
2d-3c-175m	27.82	3.41	3.41	4.49	747.54	53.61	45.68	48.62	773.36	57.02	49.08	53.11
2d-3c-75m	44.08	5.19	5.18	6.72	1272.25	79.62	89.83	76.87	1316.33	84.81	83.59	95.03
2d-10c-125m	204.3	23.82	23.86	28.68	5864.02	493.51	592.43	377.13	6068.32	517.33	616.29	405.81
2d-10c-175m	196.31	23.82	23.74	28.54	6126.79	666.03	738.06	382.09	6323.1	693.85	761.8	610.63
2d-10c-75m	265.21	31.64	31.54	37.63	6718.24	511.3	562.41	368.35	6983.45	542.94	655.75	605.98
2d-20c-125m	1086.36	145.49	144.88	164.79	20277	2499.98	1720.7	1902.1	21363.36	2645.47	1865.58	2066.99
2d-20c-175m	1391.1	185.81	184.94	209.09	27126.24	2003.9	2623.02	3754.74	29153.4	3089.71	2807.96	3690.83
2d-20c-75m	1892.81	122.18	122.1	139.09	21212.64	2172.16	2212.03	2501.54	23103.45	2334.13	2334.13	2334.13
2d-2c-125m	6.6	0.86	0.86	1.29	324.86	25.28	21.8	29.76	331.46	26.14	22.66	31.05
2d-2c-175m	2.88	0.33	0.33	0.47	138.95	12.53	7.41	9.91	141.53	7.74	7.74	7.74
2d-2c-75m	4.97	0.54	0.54	0.73	253.73	13.94	17.95	17.52	258.7	14.48	14.48	18.25
2d-30c-125m	5146.73	486.77	484.67	531.18	68901.43	8745.96	6033.67	10323.1	7408.16	9232.73	6518.34	10854.28
2d-30c-175m	5052.54	418.73	419.43	459.48	71943.24	6259.32	7104.21	5154.99	76975.7			

Data set	Precipitation time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters
1004-10c-125m	1559.47	123.64	123.7	68257.9	3308.3	4828.02	69817.37	3432	4988.55
1004-10c-175m	1518.51	119.16	119.2	68939.9	4020.51	4248.32	70458.41	4139.71	4405.01
1004-10c-75m	1588.36	136.59	136.93	61734.5	4360.82	5922.39	68722.86	4497.75	5969.08
1004-20c-125m	1460.05	468.56	467.61	197312.65	35752.9	25696.5	198772.7	21181.51	26315.47
1004-20c-175m	1118.83	440.57	441.04	182929.23	24967.7	16390.9	184088.06	25408.74	16975.79
1004-20c-75m	1661.64	470.26	468.87	219990.83	30351.5	23188.2	221652.47	16120.97	23805.94
1004-2c-125m	34.94	3.13	3.14	2558.29	100.74	98.7	2932.23	101.84	114.38
1004-2c-175m	38.23	2.9	2.9	2695.53	96.71	119.51	2733.76	85.99	123.32
1004-2c-75m	84.08	7	7	4909.47	192.8	247.33	4993.55	199.8	266.55
1004-30c-125m	5760.12	462.07	462.15	226056.8	11957.87	14956.55	241816.92	19404.58	15569.06
1004-30c-175m	5240.54	1323.45	1323.45	251745.12	37861.24	68988.4	61115.43	39184.69	70730.78
1004-30c-75m	5430.69	1027.46	1027.46	252835.54	34333.82	29032.6	238206.23	34297.64	30400.58
1004-5c-125m	240.67	18.81	18.84	16063.8	6367.77	545.29	16304.47	655.61	571.63
1004-5c-175m	416.16	33.77	33.83	18648.5	990.77	851.03	19064.66	1161.57	895.83
1004-5c-75m	318.86	25.51	25.5	18722.1	846.67	818.22	19040.96	872.17	852.76
104-10c-125m	319.88	25.79	25.77	11171.9	837.32	707.29	11491.78	731.06	624.9
104-10c-175m	277.83	20.13	20.08	10057.2	498.33	721.69	10335.03	518.46	730.96
104-10c-75m	220.74	16.48	16.48	9078.58	396.24	331.95	9299.32	412.72	354.53
104-20c-125m	1253.22	99.09	99.09	32143.53	2201.38	2500.5	33396.75	2300.43	2623.76
104-20c-175m	1287.38	102.86	102.86	42235.92	3155.61	2367.03	43523.3	2962.64	2495.03
104-20c-75m	1495.49	121.04	120.91	41602.64	2128.76	5102.18	43098.13	3702.33	5253.05
104-2c-125m	7.89	0.6	0.59	462.35	12.75	13.77	470.24	14.98	14.73
104-2c-175m	5.42	0.37	0.37	349.87	9.3	9.74	355.29	9.67	10.34
104-2c-75m	2.82	0.23	0.23	174.58	8.61	7.63	177.4	5.13	8.84
104-30c-125m	5320.54	327.86	327.86	107058.22	11174	11059.2	112378.76	7064.53	11500.85
104-30c-175m	5034.86	402.26	401.98	104683.42	6366.38	6639.59	109718.28	8161.74	7114.02
104-30c-75m	5128.76	372.58	371.04	103793.54	5840.56	6191.28	108922.3	13300.74	6635.2
104-5c-125m	46.06	3.95	3.94	1662.31	107.13	113.59	1708.37	68.5	119.27
104-5c-175m	53.51	4.46	4.48	1964.03	121.68	101.04	2017.54	105.52	120.83
104-5c-75m	61.16	4.67	4.68	2572.01	134.3	94.37	2633.17	138.97	109.88
204-10c-125m	273.37	21.52	21.5	12453.1	581.27	902.01	12726.47	602.79	1214.75
204-10c-175m	405.14	32.11	32.03	15488.5	1631.52	840.5	15893.64	1092.03	886.76
204-10c-75m	344.74	28.42	28.36	12892.2	1110.93	593.95	13236.94	1139.35	635.47
204-20c-125m	2012.43	177.92	177.79	93454	6046.71	9364.56	86366.43	9542.35	6700.98
204-20c-175m	2069.92	168.33	167.83	75707	4454.33	6469.68	77776.92	6224.66	8195.21
204-20c-75m	1956.12	161.26	161.26	61646.6	5052.28	4023.9	63602.72	5213.66	5024.82
204-2c-125m	19.71	1.57	1.57	1137.58	45.39	50.21	1157.29	46.96	52.73
204-2c-175m	18.68	1.44	1.47	1073.98	43.53	43.68	1092.66	44.97	45.82
204-2c-75m	20.38	1.62	1.59	1177.48	46.61	51.79	1197.86	48.23	55.38
204-30c-125m	5370.46	361.54	361.54	164464.42	20519.5	13624.3	169834.88	20881.26	14082.55
204-30c-175m	5434.12	390.83	390.83	125117.47	13676.2	13357	130351.59	17134.15	13859.9
204-30c-75m	5228.25	441.31	440.69	146856.93	26653.1	14138.8	152085.18	27094.41	14707.43
204-5c-125m	130.38	10.93	10.93	4930.26	291.37	333.59	5060.64	302.3	297.5
204-5c-175m	171.12	16.02	16.02	4713.83	184.47	202.32	4790.95	190.49	211.26
204-5c-75m	91.44	8.59	8.51	4612.69	277.1	207.65	4704.13	285.69	219.83
2d-10c-125m	154.95	11.17	11.17	2991.08	93.68	94.18	3146.63	104.89	109.54
2d-10c-175m	127.46	9.3	9.3	2795.27	126.88	97	2922.73	145.31	136.18
2d-10c-75m	140.54	10.1	10.06	2885.6	103.58	95.7	3026.14	113.68	109.7
2d-20c-125m	840.015	101.64	101.68	16254.4	709.36	669.11	17094.415	135.4	135.4
2d-20c-175m	1113.04	59.61	59.85	11162.4	732.61	379.88	12275.44	573.1	792.27
2d-20c-75m	916.765	82	82.1	16273.45	612.62	799.63	17190.215	694.62	454.13
2d-2c-125m	3.49	0.27	0.27	159.25	4.19	5.39	162.74	4.46	5.86
2d-2c-175m	2.36	0.18	0.18	120.01	4.29	3.15	122.37	3.25	3.47
2d-2c-75m	3.79	0.28	0.28	186.59	6.76	5.31	190.38	7.04	5.81
2d-30c-125m	5124.43	234.11	233.08	29152.51	1956.04	1122.34	34276.94	2190.15	1552.22
2d-30c-175m	4987.26	244.73	244.25	27043.15	1758.38	1480.93	32030.41	2003.11	1765.94
2d-30c-75m	5043.14	193.48	192.73	25036.34	882.06	998.32	30079.48	1075.54	868.63
2d-5c-125m	27.86	2	2.01	785.2	26.53	24.35	813.06	28.53	27.44
2d-5c-175m	27.82	1.99	1.99	747.54	26.34	21.09	775.56	26.28	24.17
2d-5c-75m	44.08	2.99	2.99	1272.25	41.17	37.62	1316.33	44.16	42.16
5d-10c-125m	204.3	14.19	14.25	5864.02	365.54	427.47	6068.32	379.73	220.12
5d-10c-175m	196.31	13.65	13.63	6126.79	275.91	309.31	6321.4	244.49	289.54
5d-10c-75m	265.21	19.21	19.21	6718.24	285.64	300.47	6983.45	341.85	325.76
5d-20c-125m	1086.36	86.15	85.71	20277	1437.16	1295.87	21563.36	1323.31	1125.86
5d-20c-175m	1591.11	110.87	110.19	27562.3	1313.77	1489.9	29153.4	1424.64	1624.68
5d-20c-75m	1892.81	70.59	70.6	21212.64	1133.03	882.41	23105.45	1203.62	970.14
5d-2c-125m	6.6	0.51	0.51	324.86	8.81	8.24	331.46	9.32	9.18
5d-2c-175m	2.58	0.2	0.2	138.95	6.28	3.6	141.53	6.48	3.94
5d-2c-75m	4.97	0.34	0.35	253.73	10.99	6.1	258.7	8.23	8.58
5d-30c-125m	5146.73	279.28	278.28	68901.43	4624.59	5800.25	74048.16	4903.87	6125.35
5d-30c-175m	5032.54	239.4	241.05	71943.24	3198.25	2033.58	76975.78	3457.65	2315.19
5d-30c-75m	5138.96	319.93	319.85	69875.32	5429.9	3488.68	75014.28	3257.44	3860.01
5d-5c-125m	28.3	2	2	1109.03	66.15	56.9	1137.33	68.15	42.15
5d-5c-175m	41.08	2.79	2.8	1479.01	82.75	92.79	1520.09	85.54	59.89
5d-5c-75m	30.56	2.2	2.2	1185.01	39.29	55.05	1215.57	41.49	58.26

Table C.19: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the artificial data sets.

Data set	CAOS-CD	Pre-calculation time			CAOS-CD	Clustering time			CAOS-CD	Overall time		
		CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters		CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters		CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
100d-10c-125m	1589.47	78.53	78.4	115.02	6857.9	184.94	2478.84	3653.31	69817.57	1927.47	2557.24	3768.33
100d-10c-175m	1518.51	80.65	80.42	117.8	6899.9	347.2	1947.67	2218.38	70458.41	3507.85	2028.09	2361.18
100d-10c-75m	1588.36	80.88	81.12	121.2	6713.4	1969.7	2802	2512.67	68782.86	2633.87	2883.12	2633.87
100d-20c-125m	1460.05	888	286.02	438.53	19731.65	14201.3	9672.09	9137.11	198772.7	10489.58	4080.38	1489.71
100d-20c-175m	1118.83	309.61	309.15	454.5	182929.23	16475.7	8695.65	6887.82	184048.06	16785.31	9904.8	7342.32
100d-20c-75m	1661.64	292.41	293.06	442.01	219990.83	12773.4	11987.1	7479.71	221652.47	11585.81	12280.16	7921.22
100d-2c-125m	34.94	2.1	2.1	3.03	2588.29	71.79	74.74	79.35	2593.23	73.89	76.84	82.38
100d-2c-175m	38.23	2.2	2.21	3.12	2695.53	39.4	62.3	60.75	2733.76	41.6	64.51	63.87
100d-2c-75m	84.08	4.92	4.9	7.12	4909.47	114.83	143.22	134.66	4993.55	119.75	148.12	141.78
100d-30c-125m	5760.12	287.13	288.67	437.27	236056.8	7744.89	75346.91	7283.62	241816.66	8033.02	7565.58	7720.89
100d-30c-175m	5240.54	933.98	933.21	1352.34	251745.12	29049.3	31314.6	5576.91	256983.28	32247.31	32247.31	57121.44
100d-30c-75m	5430.69	681.46	680.3	1022.46	252835.84	31281.8	19747.5	17274.6	258266.23	31899.64	20427.8	18297.66
100d-5c-125m	240.67	13.07	13.08	20.58	16063.8	401.03	394.18	300.01	16304.47	414.1	407.26	3320.59
100d-5c-175m	416.16	23.58	23.6	34.47	18684.5	733.82	931.29	1178.56	19064.66	757.4	757.4	954.89
100d-5c-75m	318.86	18.64	18.66	27.61	18722.1	733.43	603.18	546.5	19040.96	752.07	752.07	814.84
100d-10c-125m	319.88	15.52	15.52	24.35	11171.9	358.32	498.85	325.51	11491.78	514.37	514.37	621.87
100d-10c-175m	277.83	13.29	13.27	20.44	10067.2	356.58	622.28	798.87	10353.03	639.87	635.55	635.55
100d-10c-75m	220.74	11.49	11.37	17.47	9078.58	239.46	328.5	248.78	9299.32	250.9	250.9	266.25
100d-20c-125m	1253.22	68.34	65.34	89.72	32143.53	1227.35	975.97	1879.64	33306.75	1292.69	1041.31	1041.31
100d-20c-175m	1287.38	68.03	67.76	92.91	42235.92	1551.3	1789.89	1528.76	43523.3	1629.63	1857.65	1621.67
100d-20c-75m	1495.49	82.26	82.28	112.21	41602.64	1103.82	1582.41	2246.13	45098.13	1880.08	1604.69	2358.34
100d-2c-125m	7.89	0.4	0.4	0.76	462.35	9.34	7.44	6.35	470.24	9.74	9.74	7.11
100d-2c-175m	5.42	0.28	0.28	0.5	349.87	5.32	3.71	7.24	355.29	5.6	5.6	4.29
100d-2c-75m	2.82	0.16	0.16	0.3	174.58	4.24	2.41	3.99	177.4	4.4	4.4	3.99
100d-30c-125m	5320.54	207.52	206.71	269.38	107058.22	6902.07	6590.07	4544.2	11278.76	7109.59	6715.78	4813.58
100d-30c-175m	5034.86	255.19	255.03	327.93	106683.42	4518.62	4421.88	3808.49	109718.28	4773.81	4676.91	4136.42
100d-30c-75m	5128.76	247.26	246.92	319.21	103793.54	5788.84	3863.33	4732.72	108922.3	6036.1	4160.25	5031.93
100d-5c-125m	46.06	2.48	2.49	4.21	1662.31	65.33	65.09	42.75	1708.37	1708.37	678.3	617.88
100d-5c-175m	53.51	2.98	2.98	5.36	1964.03	67.48	80.19	44.48	2017.54	70.46	83.17	49.84
100d-5c-75m	61.16	3.12	3.12	5.08	2572.01	73.57	56.93	70.82	2633.17	76.69	60.05	75.9
200d-10c-125m	273.37	14.71	14.66	25.12	1245.1	404.75	300.79	326.66	12726.47	147.96	315.45	315.45
200d-10c-175m	405.14	20.5	20.45	34.72	15488.5	779.54	419.55	458.18	15893.64	800.04	440	492.9
200d-10c-75m	344.74	18.14	18.07	31.24	12897.2	412.25	408.05	1498.22	1326.94	430.39	426.12	1529.46
200d-20c-125m	2012.43	116.12	115.58	168.49	84354	6721.32	5165.14	3276.63	86366.43	6837.44	5280.72	4261.2
200d-20c-175m	2069.92	113.86	113.87	168.09	75707	2958.89	4840.53	3655.51	86666.43	77776.92	4954.4	3823.6
200d-20c-75m	1956.12	107.11	106.91	163.22	61646.6	3097.93	3366.02	4524.19	65602.72	3205.74	3472.93	4687.41
200d-2c-125m	19.71	1.17	1.18	1.95	1137.58	33.8	36.58	23.33	1157.29	34.97	37.76	25.28
200d-2c-175m	18.68	0.99	0.99	1.68	1073.98	17.46	24.78	32.73	1092.66	18.45	25.77	34.41
200d-2c-75m	20.38	1.16	1.16	2.09	1177.48	28.01	20.25	18.68	1197.86	29.17	21.41	20.77
200d-30c-125m	5370.46	243.89	243.64	340.41	16464.42	8471.29	7482.87	7735.45	169834.88	8713.18	7726.51	8073.86
200d-30c-175m	5434.12	266.46	266.33	378.56	125117.47	7403.59	11260	4762.22	130651.59	7470.05	11526.33	5140.78
200d-30c-75m	5228.25	279.1	278.87	407.02	146856.93	5139.77	12970.1	12970.1	132488.97	132488.97	14273.72	14273.72
200d-5c-125m	130.38	7.92	7.91	13.22	4930.26	230.09	155.1	166.89	5060.64	238.01	163.01	151.14
200d-5c-175m	77.12	3.76	3.76	6.66	4713.83	100.93	118.72	144.48	4790.95	104.69	122.48	151.14
200d-5c-75m	91.44	5.26	5.26	8.92	4612.69	119.72	203.82	215.96	4704.13	124.88	209.08	224.88
2d-10c-125m	154.95	7.27	7.25	11.43	2991.68	57.48	70.72	53.53	3146.63	64.75	77.97	64.96
2d-10c-175m	127.46	6.69	6.66	9.6	2795.27	53.32	75.7	70.03	3146.63	59.38	81.75	79.63
2d-10c-75m	140.54	6.69	6.59	10.58	2888.6	54.24	51.24	50.9	3026.14	60.93	57.9	61.48
2d-20c-125m	840.015	65.3	65.39	86.75	16254.4	469.83	424.71	424.71	17094.415	535.13	439.76	511.46
2d-20c-175m	1113.04	39.08	39.28	53.65	11162.4	259.18	264.04	279.52	12275.44	298.26	333.17	303.32
2d-20c-75m	916.765	53.05	53.1	71.83	16273.45	557.34	305.91	294.14	17190.215	610.39	359.01	365.97
2d-2c-125m	3.49	0.19	0.19	0.39	159.25	2.43	2.19	2.88	162.74	2.62	2.38	3.27
2d-2c-175m	2.36	0.12	0.12	0.25	120.01	1.79	3.02	1.18	122.37	1.9	2.2	1.43
2d-2c-75m	3.79	0.21	0.21	0.43	186.59	2.64	2.08	1.45	190.38	2.85	3.23	3.23
2d-30c-125m	5124.43	151.66	151.54	192.04	29152.51	765.02	820.89	1035.79	34276.94	9126.67	972.43	1227.53
2d-30c-175m	4987.26	159.34	159.34	192.04	27043.15	1121.33	1372.79	688.3	33203.41	1280.67	1532.13	888.25
2d-30c-75m	5003.14	125.37	125.12	161.36	25036.34	694.57	763.88	581.22	30079.48	819.94	819.94	742.58
2d-3c-125m	27.86	1.28	1.3	2.38	785.2	9.94	16.22	14.1	813.06	11.22	17.52	16.48
2d-3c-175m	27.82	1.29	1.29	2.38	747.54	14.83	13.75	11.03	775.36	16.12	15.04	13.41
2d-3c-75m	44.08	2.17	2.16	3.72	1127.25	26.33	21.82	20.32	1316.33	28.5	23.98	23.98
5d-10c-125m	204.3	9.41	9.47	14.36	5864.02	148.36	225.38	227.7	6068.32	157.97	234.85	242.06
5d-10c-175m	196.31	9.51	9.5	14.3	6126.79	147.67	220.48	244.4	6923.1	157.18	229.98	229.98
5d-10c-75m	265.21	12.43	12.42	18.49	6718.24	146.73	158.98	201.28	6983.45	159.16	171.4	219.77
5d-20c-125m	1086.36	55.48	55.15	75.01	20277	878.14	642.49	598.84	21363.36	933.62	697.64	673.85
5d-20c-175m	1391.1	71.53	71.53	95.81	27562.3	908.49	1276.58	896.27	29153.4	1040.37	1348.11	1348.11
5d-20c-75m	1892.81	47.52	47.65	64.74	21212.64	782.44	542.47	7.5	23105.45	829.96	590.12	743.25
5d-2c-125m	6.6	0.36	0.35	0.77	324.86	6.65	5.68	7.01	331.46	7.01	6.03	8.27
5d-2c-175m	2.88	0.13	0.13	0.27	138.95	3.38	2.64	2.1	141.53	3.51	2.77	2.37
5d-2c-75m	4.97	0.24	0.24	0.42	253.73	4.76	3.83	4.12	258.7	5	4.07	4.54
5d-30c-125m	5146.73	180.04	179.87	226.33	68901.43	2827.76	1791.62	2008.51	74048.16	3007.8	1971.49	2234.84
5d-30c-175m	5052.54	160.7	162.23	203.59	71943.24	2461.44	1493.35	1233.56	76995.78	2622.14	1653.58	1437.15
5d-30c-75m	5138.96	206.79	206.69	258.14	69875.32	1959.91	2452.13	2272.27	75014.28	2658.82	2658.82	2530.41
5d-5c-125m	28.3	1.4	1.4									

Data set	Precipitation time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters
1004-10c-125m	1559.47	23.38	59.92	68257.9	401.19	382.76	69817.37	424.57	442.68
1004-10c-175m	1518.51	24.06	61.44	68939.9	544.49	380.01	70458.41	568.55	441.45
1004-10c-75m	1588.36	26.03	66.34	67134.5	511.31	972.21	68722.86	1060.86	1038.55
1004-20c-125m	1460.05	90.11	241.48	197312.65	1807.95	2824.85	198772.87	3340.94	3066.33
1004-20c-175m	1118.83	89.72	233.56	1919.5	2333.55	2804.63	221652.47	2009.42	2423.27
1004-20c-75m	1661.64	95.88	245.37	219290.83	3131.71	2105.94	221652.47	1831.76	2351.31
1004-2c-125m	34.94	0.65	1.58	2558.29	13.94	12.37	2933.23	1074	13.95
1004-2c-175m	38.23	0.66	1.58	2695.53	9.21	10.3	2733.76	1117	11.88
1004-2c-75m	84.08	1.54	3.74	4909.47	22.6	20.17	4993.55	197	23.91
1004-30c-125m	5760.12	94.58	245.43	252056.8	2843.81	4111.11	241816.92	2937.48	4356.54
1004-30c-175m	5240.54	258.1	676.84	251745.12	8166.62	6316.52	256985.66	7511.18	6993.36
1004-30c-75m	5430.69	206.48	546.69	252835.54	4203.93	5293.53	238206.23	3265.56	5840.22
1004-5c-125m	240.67	4.25	11.76	16063.8	68.67	71.76	16304.47	9019	83.52
1004-5c-175m	416.16	6.55	17.5	18648.5	135.72	132.3	19064.66	16177	149.8
1004-5c-75m	318.86	5.53	14.44	18722.1	103.29	76.97	19040.96	8767	91.41
104-10c-125m	319.88	4.41	13.22	11711.9	58.84	72.62	11491.78	77.03	63.23
104-10c-175m	277.83	3.77	10.92	10057.2	34.79	62.06	10335.03	80.27	85.84
104-10c-75m	220.74	3.16	9.26	9078.58	44.47	41.33	9299.32	50.38	72.98
104-20c-125m	1253.22	18.77	43.33	32143.53	296.25	207.91	33396.75	315.02	448.85
104-20c-175m	1287.38	19.88	45.04	42235.92	364.24	378.25	43523.3	380.02	251.24
104-20c-75m	1495.49	21.8	51.64	41602.64	451.84	426.51	43098.13	473.64	480.29
104-2c-125m	7.89	0.12	0.48	462.35	1.84	1.32	470.24	1.6	1.8
104-2c-175m	5.42	0.08	0.31	349.87	0.56	0.81	355.29	0.64	1.12
104-2c-75m	2.82	0.05	0.2	174.58	0.5	0.64	177.4	0.55	0.69
104-30c-125m	5320.54	59.8	122.15	107058.22	577.01	899.55	112378.76	637.09	1021.7
104-30c-175m	5034.86	73.42	145.65	104683.42	1829.14	869.13	109718.28	1825.96	1014.78
104-30c-75m	5128.76	70.5	142.59	103793.54	511.36	1340.29	108922.3	581.86	1482.88
104-5c-125m	46.06	0.76	2.49	1662.31	9.41	14.11	1708.37	10.17	14.87
104-5c-175m	53.51	0.9	3.28	1964.03	11.06	23.88	2017.54	11.96	27.16
104-5c-75m	61.16	0.92	2.87	2572.01	8.55	22.26	2633.17	17.69	9.48
204-10c-125m	273.37	4.48	14.94	12453.1	86.2	75.25	12726.47	146.67	90.63
204-10c-175m	405.14	6.38	20.59	15488.5	121.78	188.15	15893.64	128.16	208.74
204-10c-75m	344.74	5.67	18.82	12892.2	154.95	173.79	13246.94	97.47	95.52
204-20c-125m	2012.43	35.03	88.26	84354	749.49	767.82	86366.43	784.62	786.95
204-20c-175m	2069.92	33.24	1145.58	1028.39	751.92	1254.43	7776.92	1178.82	1061.4
204-20c-75m	1956.12	29.46	85.89	61646.6	548.44	722.93	63602.72	578.05	985.46
204-2c-125m	19.71	0.35	1.13	1137.58	3.22	4.2	1157.29	3.56	5.33
204-2c-175m	18.68	0.33	1.02	1073.98	3.27	3.52	1092.66	3.6	4.54
204-2c-75m	20.38	0.37	1.3	1177.48	5.59	4.12	1197.86	5.96	5.42
204-30c-125m	5370.46	69.78	166.95	164464.42	787.17	1419.98	169834.88	856.95	1586.93
204-30c-175m	5434.12	74.53	186.92	125117.47	2105.97	1505.23	130551.59	2180.5	3506.69
204-30c-75m	5238.25	82.74	209.92	146856.93	959.84	2696.28	152085.18	1533.5	2062.2
204-5c-125m	130.38	2.36	7.7	4930.26	34.96	26.69	5060.64	37.32	28.6
204-5c-175m	77.12	1.22	4.13	4713.83	23.22	25.3	4790.95	23.84	29.43
204-5c-75m	91.44	1.7	5.37	4612.69	16.51	47.88	4704.13	18.21	53.25
2d-10c-125m	154.95	2.11	6.26	2991.08	8.34	8.01	3146.63	10.46	14.27
2d-10c-175m	127.46	1.74	5.3	2795.27	14.53	11.76	2922.73	13.34	17.06
2d-10c-75m	140.54	1.94	5.83	2885.6	11.15	10.84	3026.14	12.76	16.64
2d-20c-125m	840.015	18.08	39.59	16254.4	79.39	77.65	17094.415	97.47	159.57
2d-20c-175m	1113.04	10.92	25.58	11162.4	61.73	34.2	12275.44	72.73	117.24
2d-20c-75m	916.765	14.86	33.69	16273.45	85.92	49.39	17190.215	100.78	83.08
2d-2c-125m	3.49	0.06	0.26	159.25	0.27	0.28	162.74	0.33	0.54
2d-2c-175m	2.36	0.04	0.17	120.01	0.37	0.37	122.37	0.41	0.54
2d-2c-75m	3.79	0.06	0.28	186.59	0.45	0.37	190.38	0.41	0.65
2d-30c-125m	5124.43	41.04	81.62	12079	165.28	120.86	34276.94	161.83	202.48
2d-30c-175m	4987.26	42.86	83.52	27043.15	146.51	83.47	32030.41	189.37	166.99
2d-30c-75m	5043.14	33.5	69.55	25036.34	153.52	100.14	30079.48	187.02	169.69
2d-5c-125m	27.86	0.38	1.47	785.2	1.61	1.62	813.06	1.99	3.09
2d-5c-175m	27.82	0.37	1.46	747.54	1.38	2.42	775.36	1.75	3.88
2d-5c-75m	44.08	0.6	2.15	1272.25	1.68	3	1316.33	2.29	3.68
5d-10c-125m	204.3	2.73	7.62	5864.02	37.1	52.67	6068.32	39.83	60.29
5d-10c-175m	196.31	2.57	7.35	6126.79	22.95	51.92	6323.1	25.52	59.27
5d-10c-75m	265.21	3.53	9.59	6718.24	35.45	30.76	6983.45	39.13	40.35
5d-20c-125m	1086.36	15.66	35.42	20277	135.93	144.05	21563.36	151.62	179.47
5d-20c-175m	1591.1	20.04	44.43	27562.3	208.62	127.54	29153.4	238.83	228.61
5d-20c-75m	1892.81	13.23	30.36	21212.64	96.94	130.39	23105.45	110.17	171.11
5d-2c-125m	6.6	0.11	0.53	324.86	0.44	1.16	331.46	0.94	1.69
5d-2c-175m	2.58	0.04	0.18	138.95	0.51	0.29	141.53	0.55	0.44
5d-2c-75m	4.97	0.07	0.25	253.73	1.28	1.03	258.7	1.35	1.45
5d-30c-125m	5146.73	48.23	94.98	68901.43	463.83	342.32	74048.16	450.17	437.3
5d-30c-175m	5032.54	44.16	86.55	71943.24	284.38	426.54	76975.78	328.54	513.09
5d-30c-75m	5138.96	56.22	108.27	69875.32	383.47	502.38	75014.28	410.35	610.65
5d-5c-125m	28.3	0.42	1.4	1109.03	4.05	4.28	1370.33	4.47	5.68
5d-5c-175m	41.08	0.56	1.91	1479.01	8.52	5.28	1520.09	9.08	7.19
5d-5c-75m	30.56	0.42	1.44	1185.01	2.55	2.72	1215.57	2.97	4.16

Table C.21: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the artificial data sets.

Data set	CAOS-CD				Preclassification time				Clustering time				Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	
100d-10c-125m	1589.47	13.46	12.45	48.91	6857.9	211.6	262.49	140.73	69817.57	224.06	274.94	224.06	224.06	274.94	224.06	
100d-10c-175m	1518.51	13.05	13.02	50.41	68999.9	122.90	219.35	150.87	70488.41	135.96	232.37	135.96	135.96	232.37	135.96	
100d-10c-125m	1588.36	13.67	13.71	53.93	67134.5	228.05	311.05	111.41	68722.86	241.72	324.76	241.72	241.72	324.76	241.72	
100d-20c-125m	1460.05	44.21	44.05	195.12	197312.65	703.84	378.31	786.53	198772.7	748.05	422.36	748.05	748.05	422.36	748.05	
100d-20c-175m	1118.83	46.74	46.74	191.88	182929.23	1006.13	1491.14	852.15	184048.06	1057.88	1577.88	1057.88	1057.88	1577.88	1057.88	
100d-20c-125m	1661.64	47.13	46.96	196.5	219990.83	718.79	804.35	789.64	221652.47	852.15	851.31	852.15	852.15	851.31	852.15	
100d-2c-175m	34.94	0.43	0.43	1.36	2588.29	3.87	3.34	6.46	2293.23	4.3	3.77	4.3	4.3	3.77	4.3	
100d-2c-125m	38.23	0.4	0.4	1.32	2695.53	4.36	3.29	4.55	2733.76	4.76	3.69	4.76	4.76	3.69	4.76	
100d-2c-75m	84.08	0.84	0.84	3.04	4909.47	10.1	8.58	9.45	4993.55	10.94	9.42	10.94	10.94	9.42	10.94	
100d-30c-125m	5760.12	45.93	46.12	196.71	234605.68	437.08	783.03	1341.15	241816.92	483.01	829.15	483.01	483.01	829.15	483.01	
100d-30c-175m	5240.54	128.68	128.68	548.18	2638.35	2638.35	3918.46	2193.36	256986.66	2791.29	4007.14	2791.29	2791.29	4007.14	2791.29	
100d-30c-125m	5430.69	108.7	108.47	450.18	253835.54	2083.92	1508.38	1348.21	258266.23	1348.21	1616.85	1348.21	1348.21	1616.85	1348.21	
100d-5c-125m	240.67	2.28	2.27	9.78	16063.8	24.36	35.52	22.54	16304.47	21.62	37.29	21.62	21.62	37.29	21.62	
100d-5c-175m	416.16	3.75	3.75	14.69	18684.5	47.74	96.2	67.18	19064.66	51.49	99.95	51.49	51.49	99.95	51.49	
100d-5c-75m	318.86	3.08	3.09	12.04	18722.1	36.73	36.98	26.18	19040.96	39.81	40.07	39.81	39.81	40.07	39.81	
100d-10c-125m	319.88	2.24	2.24	11.1	11171.9	22.53	23.39	22.12	11491.78	24.77	33.22	24.77	24.77	33.22	24.77	
100d-10c-175m	277.83	2.03	2.02	9.2	10067.2	31.21	17.49	23.73	10353.03	33.24	19.51	33.24	33.24	19.51	33.24	
100d-10c-75m	220.74	1.65	1.64	7.75	9078.58	24.26	19.56	12.19	9299.32	25.91	21.2	25.91	25.91	21.2	25.91	
100d-20c-125m	1253.22	9.34	9.34	33.91	32143.53	44.9	117.84	116.86	33396.75	54.24	127.18	54.24	54.24	127.18	54.24	
100d-20c-175m	1287.38	10.83	10.85	40.61	42233.92	147.24	73.44	70.18	43523.3	156.64	82.78	156.64	156.64	82.78	156.64	
100d-20c-75m	1495.49	10.83	10.85	40.61	41602.64	108.27	189.81	84.01	43098.13	119.1	191.1	119.1	119.1	191.1	119.1	
100d-2c-125m	7.89	0.07	0.07	0.44	462.35	0.5	0.37	0.36	470.24	0.57	0.57	0.57	0.57	0.57	0.57	
100d-2c-175m	5.42	0.05	0.04	0.28	349.87	0.3	0.35	0.3	355.29	0.35	0.39	0.35	0.35	0.39	0.35	
100d-2c-75m	2.82	0.03	0.03	0.17	174.58	0.18	0.22	0.25	177.4	0.21	0.25	0.21	0.21	0.25	0.21	
100d-30c-125m	5320.54	28.49	28.32	90.09	107058.22	207.04	292.31	353.61	11278.76	235.53	320.63	235.53	235.53	320.63	235.53	
100d-30c-175m	5024.86	35.02	35.06	107.94	104683.42	260.37	404.57	317.91	109718.28	295.39	439.63	295.39	295.39	439.63	295.39	
100d-30c-75m	5128.76	33.98	33.95	106.27	103793.54	146.57	386.06	234.49	108922.3	180.55	420.01	180.55	180.55	420.01	180.55	
100d-5c-125m	46.06	0.41	0.4	2.14	1662.31	4.13	6.61	3.05	1708.37	4.54	7.01	4.54	4.54	7.01	4.54	
100d-5c-175m	53.51	0.48	0.48	2.86	1964.03	2.41	3.23	4.09	2017.54	2.89	3.71	2.89	2.89	3.71	2.89	
100d-5c-75m	61.16	0.5	0.49	2.46	2572.01	5.87	8.35	4.66	2633.17	6.07	8.84	6.07	6.07	8.84	6.07	
200d-10c-125m	273.37	2.33	2.32	12.84	12483.5	35.25	60.4	33.83	12726.47	37.58	62.72	37.58	37.58	62.72	37.58	
200d-10c-175m	405.14	3.22	3.21	17.42	15488.5	42.4	74.91	26.58	15893.64	45.62	78.12	45.62	45.62	78.12	45.62	
200d-10c-75m	344.74	2.89	2.87	16.03	12892.2	40.41	21.14	20.22	13236.94	45.3	78.12	45.3	45.3	78.12	45.3	
200d-20c-125m	2012.43	18	18.01	71.07	84834	314.46	423.11	303.02	86666.43	332.46	441.12	332.46	332.46	441.12	332.46	
200d-20c-175m	2069.92	16.46	16.3	70.8	75707	218.92	256.65	503.29	77776.92	335.38	474.09	335.38	335.38	474.09	335.38	
200d-20c-75m	1956.12	14.6	14.58	70.99	61646.6	301.22	211.05	222.14	65602.72	315.82	245.63	315.82	315.82	245.63	315.82	
200d-2c-125m	19.71	0.21	0.2	0.98	1137.58	1.7	2.65	3.02	1157.29	1.91	1.91	1.91	1.91	1.91	1.91	
200d-2c-175m	18.68	0.18	0.18	0.86	1073.98	0.85	2.18	0.89	1092.66	1.03	1.03	1.03	1.03	1.03	1.03	
200d-2c-75m	20.38	0.21	0.2	1.14	1177.48	1.92	1.63	1.53	1197.86	2.13	1.83	2.13	2.13	1.83	2.13	
200d-30c-125m	3570.46	35.52	35.76	132.98	164464.42	306.56	291.35	720.02	169834.88	342.08	527.11	342.08	342.08	527.11	342.08	
200d-30c-175m	5434.12	41.84	41.84	148.84	125117.47	369.84	298.44	449.4	13051.59	406.41	335.3	406.41	406.41	335.3	406.41	
200d-30c-75m	5228.25	41.84	41.84	168.95	148856.93	968.84	1112.07	479.25	10100.64	596.24	1153.76	596.24	596.24	1153.76	596.24	
200d-5c-125m	130.38	1.21	1.22	6.52	4930.26	32.2	21.03	11.8	5060.64	33.41	14.02	33.41	33.41	14.02	33.41	
200d-5c-175m	77.12	0.64	0.64	3.54	4713.83	6.11	7.73	10.48	4790.95	6.75	8.37	6.75	6.75	8.37	6.75	
200d-5c-75m	91.44	0.87	0.88	4.56	4612.69	5.87	8.83	14.77	4704.13	6.74	9.71	6.74	6.74	9.71	6.74	
2d-10c-125m	154.95	1.05	1.04	5.23	2991.68	3.29	3.77	3.99	3146.63	4.34	4.81	4.34	4.34	4.81	4.34	
2d-10c-175m	127.46	0.9	0.89	4.44	2795.27	3.12	3.12	6	2922.73	4.1	4.01	4.1	4.1	4.01	4.1	
2d-10c-75m	140.54	0.93	0.93	4.85	2888.6	2.82	2.03	3.31	3026.14	3.75	2.96	3.75	3.75	2.96	3.75	
2d-20c-125m	840.015	8.61	8.56	30.11	16254.4	21.65	26.27	24.82	17094.415	30.26	34.83	30.26	30.26	34.83	30.26	
2d-20c-175m	1113.04	5.18	5.21	19.81	11162.4	17.54	13.32	17.08	12275.44	22.72	18.53	22.72	22.72	18.53	22.72	
2d-20c-75m	916.765	7.15	7.13	26	16273.45	21.84	13.65	24.33	17190.215	28.99	20.78	28.99	28.99	20.78	28.99	
2d-2c-125m	3.49	0.03	0.03	0.23	159.25	0.2	0.23	0.14	162.74	0.23	0.26	0.23	0.23	0.26	0.23	
2d-2c-175m	2.36	0.02	0.02	0.16	120.01	0.19	0.16	0.15	122.37	0.21	0.21	0.21	0.21	0.21	0.21	
2d-2c-75m	3.79	0.03	0.03	0.25	186.59	0.2	0.25	0.19	190.38	0.23	0.28	0.23	0.23	0.28	0.23	
2d-30c-125m	5124.43	19.37	19.34	59.82	29152.51	46.97	37.91	32.53	34276.94	66.34	57.25	66.34	66.34	57.25	66.34	
2d-30c-175m	4987.26	20.34	20.32	60.98	27043.15	54.25	56.32	40.37	32020.41	74.59	76.64	74.59	74.59	76.64	74.59	
2d-30c-75m	5093.14	16.16	16.08	52.24	25036.34	39.82	31.65	25.97	30079.48	55.98	47.73	55.98	55.98	47.73	55.98	
2d-3c-125m	27.86	0.19	0.19	1.28	785.2	0.74	0.61	0.56	813.06	0.93	0.8	0.93	0.93	0.8	0.93	
2d-3c-175m	27.82	0.2	0.2	1.28	747.54	1.29	0.61	0.56	775.36	1.49	0.8	1.49	1.49	0.8	1.49	
2d-3c-75m	44.08	0.31	0.31	1.86	1272.25	1.27	0.59	0.67	1316.33	1.58	1.58	1.58	1.58	1.58	1.58	
5d-10c-125m	204.3	1.35	1.35	6.23	5864.02	11.38	15.03	9.23	6068.32	12.72	16.38	12.72	12.72	16.38	12.72	
5d-10c-175m	196.31	1.37	1.36	6.16	6126.79	9.62	9.12	7.22	6933.1	10.99	10.48	10.99	10.99	10.48	10.99	
5d-10c-75m	265.21</															

Data set	Precipitation time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Random	CAOS-DS-Clusters
1004-10c-125m	1559.47	7.94	44.4	68257.9	150.89	113.06	69817.37	158.83	157.46
1004-10c-175m	1518.51	8.37	45.75	68939.9	187.23	183.81	70458.41	153.95	229.56
1004-10c-75m	1588.36	9.02	49.26	67134.5	147.5	159.97	68722.86	133.69	205.23
1004-20c-125m	1460.05	31.03	30.96	197312.65	537.2	718.02	198772.7	568.23	900.22
1004-20c-175m	1118.83	30.54	30.47	182929.23	481.27	473.94	184048.06	511.81	896.72
1004-20c-75m	1661.64	32.27	181.51	219990.83	839.96	473.84	221652.47	872.23	655.35
1004-2c-125m	34.94	0.29	1.23	2558.29	2.72	2.04	2933.23	3.11	3.27
1004-2c-175m	38.23	0.28	1.2	2695.53	3.07	2.75	2733.76	3.35	3.95
1004-2c-75m	84.08	0.6	2.81	4909.47	4.36	5.74	4993.55	4.52	8.55
1004-30c-125m	5760.12	31.15	181.76	236056.8	321.06	457.48	241816.92	352.71	639.24
1004-30c-175m	5240.54	91.21	509.79	251745.12	857.73	888.91	256985.66	1635.91	1368.7
1004-30c-75m	5430.69	72.41	412.57	252835.54	1136.45	1102.12	238206.23	1614.95	1514.69
1004-5c-125m	240.67	1.47	8.98	16063.8	12.99	15.14	16304.47	14.46	24.12
1004-5c-175m	416.16	2.46	13.41	18648.5	44.72	44.67	19064.66	47.18	58.08
1004-5c-75m	318.86	2.07	11	18722.1	12.19	49.14	19040.96	14.26	60.14
104-10c-125m	319.88	1.48	10.33	11171.9	10.14	12.53	11491.78	12.94	22.86
104-10c-175m	277.83	1.27	8.44	10057.2	13.83	8.57	10335.03	15.1	17.01
104-10c-75m	220.74	1.05	7.15	9078.58	4.8	4.7	9299.32	8.64	5.84
104-20c-125m	1253.22	5.8	5.82	32143.53	25.23	14.5	33396.75	62.68	44.9
104-20c-175m	1287.38	5.84	5.79	42235.92	56.67	82.76	43523.3	69.51	113.72
104-20c-75m	1495.49	6.56	36.32	41602.64	17.58	64.03	43098.13	24.14	100.35
104-2c-125m	7.89	0.05	0.41	462.35	0.25	0.24	470.24	0.3	0.65
104-2c-175m	5.42	0.03	0.26	349.87	0.22	0.22	355.29	0.33	0.48
104-2c-75m	2.82	0.02	0.16	174.58	0.17	0.2	177.4	0.19	0.38
104-30c-125m	5320.54	17.13	78.63	107058.22	127.12	90.54	112378.76	144.25	169.17
104-30c-175m	5034.86	21.14	94.53	104683.42	104.94	152.64	109718.28	126.08	247.17
104-30c-75m	5128.76	20.38	92.61	103793.54	117.15	133.44	108922.3	137.53	226.05
104-5c-125m	46.06	0.26	2	1662.31	2.73	2.29	1708.37	3.21	4.29
104-5c-175m	61.16	0.34	2.3	2572.01	1.67	2.14	2037.54	1.84	2.89
104-5c-75m	61.16	0.34	2.3	2572.01	3.09	1.96	2633.17	2.01	4.26
204-10c-125m	273.37	1.46	11.92	12453.1	12.04	12.87	12726.47	13.5	11.89
204-10c-175m	405.14	2.04	16.3	15488.5	14.04	21.49	15893.64	16.08	37.79
204-10c-75m	344.74	1.86	15.01	12892.2	27.01	15.91	13246.94	28.87	30.92
204-20c-125m	2012.43	11.13	64.37	84354	98.45	92.62	86366.43	120.42	156.99
204-20c-175m	2069.92	9.73	9.74	75707	108.33	79.15	77776.92	118.06	194.77
204-20c-75m	1956.12	9.53	65.97	61646.6	108.01	151.07	63602.72	117.54	217.04
204-2c-125m	19.71	0.14	0.91	1137.58	0.48	1.66	1157.29	1.06	2.57
204-2c-175m	18.68	0.12	0.81	1073.98	1.12	0.54	1092.66	1.24	1.35
204-2c-75m	20.38	0.14	1.07	1177.48	0.83	0.71	1097.86	0.97	1.78
204-30c-125m	5370.46	21.58	118.92	164464.42	73.07	191.68	169834.88	94.65	310.6
204-30c-175m	5434.12	23.1	135.32	125117.47	104.79	317.08	130531.59	127.89	452.4
204-30c-75m	5238.25	25.16	152.44	146856.93	213.38	239.14	152085.18	238.54	391.58
204-5c-125m	130.38	0.78	6.08	4930.26	5.94	3.93	5060.64	6.28	10.01
204-5c-175m	77.12	0.45	3.35	4713.83	3.58	9.28	4790.95	4.03	5.88
204-5c-75m	91.44	0.58	4.24	4612.69	5.41	4.09	4704.13	5.99	8.33
2d-10c-125m	154.95	0.64	4.82	2991.08	1.36	1.85	3146.63	2	6.67
2d-10c-175m	127.46	0.54	4.09	2795.27	1.48	2.21	2922.73	1.74	6.3
2d-10c-75m	140.54	0.58	4.47	2885.6	1.27	1.34	3026.14	1.85	5.8
2d-20c-125m	840.015	5.12	26.62	16254.4	17.43	6.64	17094.415	19.3	33.26
2d-20c-175m	1113.04	3.13	17.77	11162.4	7.28	8.54	12275.44	10.41	26.31
2d-20c-75m	916.765	4.22	23.05	16273.45	10.01	9.44	17190.215	14.23	32.49
2d-2c-125m	3.49	0.02	0.22	159.25	0.17	0.19	162.74	0.19	0.41
2d-2c-175m	2.36	0.01	0.15	120.01	0.12	0.12	122.37	0.14	0.27
2d-2c-75m	3.79	0.02	0.24	186.59	0.18	0.13	190.38	0.2	0.15
2d-30c-125m	11624.43	11.67	52.15	29152.51	13.49	23	34276.94	25.16	25.68
2d-30c-175m	4987.26	12	52.69	27043.15	25.47	39.32	32030.41	38.59	92.01
2d-30c-75m	5043.14	9.54	45.62	25036.34	15.62	18.4	30079.48	25.16	64.02
2d-5c-125m	27.86	0.12	1.2	785.2	0.35	0.4	813.06	0.47	0.53
2d-5c-175m	27.82	0.12	1.21	747.54	0.35	0.39	775.36	0.67	1.6
2d-5c-75m	44.08	0.19	1.73	1272.25	0.61	0.56	1316.33	0.92	2.29
5d-10c-125m	204.3	0.84	5.78	5864.02	3.75	5.07	6068.32	4.59	10.85
5d-10c-175m	196.31	0.83	5.65	6126.79	3.29	3.12	6323.1	4.12	8.77
5d-10c-75m	265.21	1.1	7.18	6718.24	4.62	4.72	6983.45	5.72	11.9
5d-20c-125m	1086.36	4.61	24.45	20277	17.04	14.2	21563.36	21.65	38.65
5d-20c-175m	1591.1	5.85	30.1	27562.3	34.24	40.9	29153.4	36.82	71
5d-20c-75m	1892.81	3.95	21.09	21212.64	15.78	14.92	23105.45	19.73	26.04
5d-2c-125m	6.6	0.04	0.46	324.86	0.3	0.24	331.46	0.34	0.7
5d-2c-175m	2.58	0.02	0.16	138.95	0.14	0.16	141.53	0.16	0.29
5d-2c-75m	4.97	0.02	0.21	253.73	0.22	0.18	258.7	0.24	0.43
5d-30c-125m	5146.73	14.15	60.97	88901.43	49.34	47.33	74048.16	63.49	108.3
5d-30c-175m	5032.54	12.57	55.65	71943.24	73.36	35.84	76975.78	85.93	91.49
5d-30c-75m	5138.96	15.99	68.06	69875.32	44.81	82.26	75014.28	60.8	35.27
5d-5c-125m	28.3	0.14	1.12	1109.03	1.4	1.09	1137.33	1.54	0.89
5d-5c-175m	41.08	0.18	1.53	1479.01	1.11	1.08	1520.09	1.29	2.61
5d-5c-75m	30.56	0.15	1.16	1185.01	1.19	0.73	1215.57	1.34	1.8

Table C.23: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the artificial data sets.

Data set	CAOS-CD				Precalculation time				CAOS-CD				Clustering time				Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters				
100d-10c-125m	1589.47	5.89	5.78	42.33	6857.9	76.97	68.96	38.55	6981.737	82.86	74.85	74.85	6981.737	82.86	74.85	74.85				
100d-10c-175m	1518.51	5.78	5.78	43.14	6899.9	61.43	38.81	51.5	7048.41	58.37	44.59	44.59	7048.41	58.37	44.59	44.59				
100d-10c-75m	1588.36	6.65	6.65	46.88	6713.5	34.22	54.22	62.58	6872.86	68.08	60.87	60.87	6872.86	68.08	60.87	60.87				
100d-20c-125m	1460.05	21.82	21.81	172.05	197312.65	346.86	391.09	285.05	198772.7	368.68	412.9	412.9	198772.7	368.68	412.9	412.9				
100d-20c-175m	1118.83	20.49	20.45	164.73	182929.23	244.42	274.6	181.5	184048.06	264.91	295.05	295.05	184048.06	264.91	295.05	295.05				
100d-20c-75m	1661.64	22.2	22.21	171.64	219990.83	173	304.29	217.64	221652.47	326.5	398.2	398.2	221652.47	326.5	398.2	398.2				
100d-3c-125m	34.94	0.24	0.23	1.16	2588.29	1.92	1.33	2.19	2593.23	1.26	1.76	1.76	2593.23	1.26	1.76	1.76				
100d-3c-175m	38.23	0.23	0.24	1.16	2695.53	1.63	1.88	1.39	2733.76	1.86	2.55	2.55	2733.76	1.86	2.55	2.55				
100d-2c-75m	84.08	0.48	0.48	2.69	4909.47	3.62	3.27	5.72	4993.53	4.1	3.75	3.75	4993.53	4.1	3.75	3.75				
100d-30c-125m	5760.12	22.43	22.43	172.48	236056.8	254.93	243.61	296.01	241816.62	276.96	266.04	266.04	241816.62	276.96	266.04	266.04				
100d-30c-175m	5240.54	61.09	61.07	481.1	251745.12	1067.75	1061.88	1181.61	256685.96	1181.61	1122.95	1122.95	256685.96	1181.61	1122.95	1122.95				
100d-30c-75m	5430.69	49.82	49.82	390.37	225383.54	574.27	727.22	360.53	228266.23	624.09	776.99	776.99	228266.23	624.09	776.99	776.99				
100d-5c-125m	240.67	1.14	1.15	8.66	16063.8	7.45	10.12	21.28	16304.47	8.59	11.27	11.27	16304.47	8.59	11.27	11.27				
100d-5c-175m	416.16	1.89	1.89	12.84	18684.5	16.99	13.17	28.86	19064.66	18.88	15.06	15.06	19064.66	18.88	15.06	15.06				
100d-5c-75m	318.86	1.48	1.48	10.42	18722.1	31.36	6.53	15.71	19040.96	32.84	8.01	8.01	19040.96	32.84	8.01	8.01				
10d-10c-125m	319.88	1.05	1.04	9.86	11171.9	5.91	7.36	4.61	11491.78	4.61	6.96	6.96	11491.78	4.61	6.96	6.96				
10d-10c-175m	277.83	0.87	0.88	8.04	10067.2	3.58	4.57	5.7	10353.03	4.45	7.25	7.25	10353.03	4.45	7.25	7.25				
10d-10c-75m	220.74	0.75	0.74	6.84	9078.58	6.87	4.57	5.73	9299.32	7.62	5.31	5.31	9299.32	7.62	5.31	5.31				
10d-20c-125m	1253.22	4.04	4.04	28.83	32143.53	21.12	21.75	20.05	33396.75	25.16	25.79	25.79	33396.75	25.16	25.79	25.79				
10d-20c-175m	1287.38	4.07	4.07	29.19	42233.92	14.92	53.51	17	43323.3	18.99	57.53	57.53	43323.3	18.99	57.53	57.53				
10d-20c-75m	1495.49	4.67	4.68	34.42	41602.64	28.47	23.46	24.07	43098.13	33.14	30.14	30.14	43098.13	33.14	30.14	30.14				
10d-2c-125m	7.89	0.04	0.04	0.4	462.38	0.22	0.2	0.23	470.24	0.24	0.24	0.24	470.24	0.24	0.24	0.24				
10d-2c-175m	5.42	0.03	0.03	0.26	349.87	0.22	0.19	0.18	355.29	0.25	0.22	0.22	355.29	0.25	0.22	0.22				
10d-2c-75m	2.82	0.02	0.02	0.16	174.58	0.2	0.2	0.17	177.4	0.22	0.22	0.22	177.4	0.22	0.22	0.22				
10d-30c-125m	5320.54	11.93	11.93	87.43	107058.22	170.19	107.92	64.55	109718.28	182.12	119.75	119.75	109718.28	182.12	119.75	119.75				
10d-30c-175m	5094.86	14.56	14.35	73.43	106683.42	64.8	64.55	56.06	108971.82	79.36	79.1	79.1	108971.82	79.36	79.1	79.1				
10d-30c-75m	5128.76	14.05	14.13	86.84	103793.54	102.06	42.9	100.29	108922.3	116.11	57.03	57.03	108922.3	116.11	57.03	57.03				
10d-5c-125m	46.06	0.2	0.2	1.94	1662.31	0.7	1.23	1.68	1708.37	0.9	1.43	1.43	1708.37	0.9	1.43	1.43				
10d-5c-175m	53.51	0.22	0.22	2.6	1964.03	0.85	1.71	1.4	2017.54	1.07	1.93	1.93	2017.54	1.07	1.93	1.93				
10d-5c-75m	61.16	0.24	0.24	2.2	2572.01	1.36	1.98	1.48	2633.17	1.6	2.22	2.22	2633.17	1.6	2.22	2.22				
20d-10c-125m	273.37	1.06	1.06	11.6	1245.1	3.36	7.55	5.54	1276.47	4.42	4.42	4.42	1276.47	4.42	4.42	4.42				
20d-10c-175m	405.14	1.46	1.45	15.68	15488.5	6.3	16.05	11.2	15893.64	7.76	7.76	7.76	15893.64	7.76	7.76	7.76				
20d-10c-75m	344.74	1.35	1.34	14.5	12897.2	8.66	10.35	11.07	13236.94	10.01	11.69	11.69	13236.94	10.01	11.69	11.69				
20d-20c-125m	2012.43	7.75	7.75	60.85	84354	76.42	32.28	78.85	86366.43	84.17	40.04	40.04	86366.43	84.17	40.04	40.04				
20d-20c-175m	2069.92	7.22	7.22	61.77	75707	68.81	62.82	44.08	77776.92	76.08	70.04	70.04	77776.92	76.08	70.04	70.04				
20d-20c-75m	1956.12	6.56	6.54	62.97	61646.6	44.33	98.3	115.37	63602.72	50.89	104.84	104.84	63602.72	50.89	104.84	104.84				
20d-2c-125m	19.71	0.1	0.1	0.88	1137.58	0.46	0.8	0.95	1157.29	0.56	0.9	0.9	1157.29	0.56	0.9	0.9				
20d-2c-175m	18.68	0.1	0.1	0.78	1073.98	0.51	0.48	0.48	1092.66	0.61	0.58	0.58	1092.66	0.61	0.58	0.58				
20d-2c-75m	20.38	0.1	0.1	1.03	1177.48	0.56	0.45	0.45	1197.86	0.66	0.55	0.55	1197.86	0.66	0.55	0.55				
20d-30c-125m	5370.46	15.33	15.42	112.66	164464.42	114.99	201.92	104.99	169834.88	129.92	217.34	217.34	169834.88	129.92	217.34	217.34				
20d-30c-175m	5434.12	16.67	16.67	128.9	125117.47	84.62	114.21	101.17	130515.9	101.17	130.88	130.88	130515.9	101.17	130.88	130.88				
20d-30c-75m	5228.25	17.5	17.5	144.77	148684.93	60.31	328.63	69.56	152083.18	77.81	346.11	346.11	152083.18	77.81	346.11	346.11				
20d-5c-125m	130.38	0.56	0.56	5.87	4930.26	1.7	6.3	4.51	5060.64	2.26	6.86	6.86	5060.64	2.26	6.86	6.86				
20d-5c-175m	77.12	0.32	0.31	3.24	4713.83	4.1	1.45	3.07	4790.95	4.42	1.76	1.76	4790.95	4.42	1.76	1.76				
20d-5c-75m	91.44	0.44	0.44	4.12	4612.69	3.16	2.44	3.6	4704.13	3.6	2.88	2.88	4704.13	3.6	2.88	2.88				
2d-10c-125m	154.95	0.45	0.45	4.62	2991.68	1.08	0.83	0.66	3146.63	1.34	1.28	1.28	3146.63	1.34	1.28	1.28				
2d-10c-175m	127.46	0.37	0.37	3.92	2795.27	1.94	0.88	0.86	2922.73	2.32	1.25	1.25	2922.73	2.32	1.25	1.25				
2d-10c-75m	140.54	0.41	0.4	4.3	2885.6	0.69	0.75	0.64	3026.14	1.1	1.15	1.15	3026.14	1.1	1.15	1.15				
2d-20c-125m	840.015	3.49	3.46	24.99	16254.4	11.75	4.51	7.05	17064.415	15.24	15.24	15.24	17064.415	15.24	15.24	15.24				
2d-20c-175m	1113.04	2.17	2.17	16.84	11162.4	6.39	5.83	3.7	12275.44	8.56	8.56	8.56	12275.44	8.56	8.56	8.56				
2d-20c-75m	916.765	2.92	2.92	21.76	16273.45	6.34	7.21	19.59	17190.215	9.16	10.13	10.13	17190.215	9.16	10.13	10.13				
2d-2c-125m	3.49	0.02	0.02	0.21	159.25	0.1	0.17	0.14	162.74	0.12	0.19	0.19	162.74	0.12	0.19	0.19				
2d-2c-175m	2.36	0.01	0.01	0.14	120.01	0.12	0.1	0.1	122.37	0.13	0.11	0.11	122.37	0.13	0.11	0.11				
2d-2c-75m	3.79	0.02	0.02	0.24	186.59	0.1	0.11	0.09	190.38	0.12	0.12	0.12	190.38	0.12	0.12	0.12				
2d-30c-125m	5124.43	7.74	7.74	48.22	29152.51	6.81	14.99	5.6	34276.94	14.55	22.69	22.69	34276.94	14.55	22.69	22.69				
2d-30c-175m	4987.26	8.14	8.09	48.83	27043.15	11.51	11.32	16.7	32050.41	19.65	19.41	19.41	32050.41	19.65	19.41	19.41				
2d-30c-75m	5093.14	6.39	6.33	42.59	25036.34	7.59	6.87	21.12	30079.48	13.98	13.2	13.2	30079.48	13.98	13.2	13.2				
2d-5c-125m	27.86	0.09	0.09	1.18	785.2	0.22	0.24	0.28	813.06	0.31	0.33	0.33	813.06	0.31	0.33	0.33				
2d-5c-175m	27.82	0.09	0.09	1.18	747.54	0.29	0.28	0.34	775.36	0.38	0.37	0.37	775.36	0.38	0.37	0.37				
2d-5c-75m	44.08	0.14</																		

Data set	Precalculation time			Clustering time			Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.61	0.21	0.28	78.13	16.53	13.15	78.74	16.74	17.72	13.43
bioipn	7.29	2.43	2.72	504.59	106.15	165.57	511.88	106.58	135.7	168.29
bpa	0.29	0.08	0.09	27.08	5.34	4.95	27.37	5.42	5.94	5.04
dermatology	0.26	0.41	0.46	144.82	29.17	32.34	146.08	29.58	30.1	32.8
ecoli	0.29	0.08	0.09	29.09	5.53	5.14	29.38	5.61	5.32	5.23
glass	0.14	0.04	0.05	12.63	2.23	2.7	12.77	2.27	2.74	2.72
heart-stalog	0.3	0.09	0.11	29.72	7.01	4.86	30.02	7.11	4.95	6.34
iono	1.25	0.36	0.41	116.91	21.86	29.38	118.16	22.22	26.06	29.79
iris	0.04	0.01	0.01	4.67	0.85	1.36	4.71	0.86	1.37	1.15
letter-rec	1260.31	720.04	754.2	45873.1	5891.59	6817.34	46933.41	6611.63	8904.71	7571.54
liver-disorders	0.32	0.07	0.1	34.87	5.75	4.06	35.19	5.82	6.97	4.16
magic	853.432	515.57	602.35	37539.3	15428.2	13722	38392.732	15943.77	12473.66	14324.35
pendigits	354.69	105.21	118.19	13734.9	4448.8	3468.21	14089.59	4554.12	3573.42	4330.8
pim	1.52	0.48	0.55	124.02	32.66	25	125.54	33.14	31.35	25.55
segment	34.08	8.85	9.95	952.5	308.94	322.07	986.58	317.8	282.72	332.02
shuttle	10588.17	3647.48	4105.54	135211.31	3864.87	36267.24	145769.48	39513.02	37278.12	40372.78
sonar	0.88	0.24	0.27	124.69	25.66	24.34	125.57	25.91	22.54	24.61
thyroids	0.24	0.05	0.05	28.43	4.85	4.78	28.67	4.9	5.24	4.83
transfusion	1.06	0.28	0.34	18.22	4.57	5.71	19.28	4.85	4.45	6.05
vehicle	4.7	1.15	1.3	354.02	65.94	56	358.72	67.1	65.15	57.3
waveform	338.58	91.4	99.32	21737.1	4196.11	4099.04	22075.68	4287.51	4289.13	4198.36
wdbc	3.08	0.9	0.99	261	42.13	51.12	264.08	43.03	40.76	52.11
wisconsin	1.77	0.45	0.54	32.77	13.05	7.06	34.54	13.5	9.83	7.6
wppc	0.42	0.12	0.13	62.77	15.12	13.92	63.19	15.24	11.36	14.05
yeast	7.51	1.96	2.22	392.59	81.54	80.3	400.1	83.5	74.58	82.52

Table C.25: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 50% of the instances applied to the real-world data sets.

Data set	Precalculation time					Clustering time					Overall time				
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters		
ball	0.61	0.1	0.1	0.18	78.13	3.85	3.92	4.84	78.74	3.95	4.02	5.02			
bioipn	7.29	1.06	1.06	0.05	504.59	46.95	61.04	43.49	511.88	48.01	62.1	44.81			
byd	0.29	0.04	0.04	0.05	27.08	1.68	2.34	2.05	27.37	1.72	2.38	2.1			
dermatology	1.26	0.19	0.19	0.24	144.82	9.63	10.62	9.87	146.08	9.82	10.81	10.11			
ecoli	0.29	0.04	0.04	0.05	29.09	1.5	1.68	1.66	29.38	1.54	1.72	1.71			
glass	0.14	0.02	0.02	0.03	12.63	0.9	1.08	1.17	12.77	1.1	1.1	1.2			
heart-voting	0.3	0.05	0.05	0.07	29.72	1.96	2.65	1.75	30.02	2.01	2.7	1.82			
iono	1.25	0.18	0.17	0.22	116.91	6.99	8.91	6.01	118.16	7.17	9.08	6.23			
iris	0.04	0	0	0.01	4.67	0.44	0.37	0.46	4.71	0.44	0.37	0.47			
letter-rec	1260.31	506.67	505.98	571.64	45673.1	4840.28	3744.14	5047.1	46933.41	5346.95	4250.12	5618.74			
liver-disorders	0.32	0.04	0.04	0.05	34.87	2.3	2.58	1.77	35.19	2.34	2.62	1.82			
magic	853.432	385.81	385.64	431.79	37539.3	7257.82	5792.06	5177.15	38392.732	7643.63	6177.7	5608.94			
pendigits	354.69	53.68	53.68	66.57	13724.9	1530.53	1583	1417.87	14089.99	1584.21	1636.68	1484.44			
pin	1.52	0.24	0.24	0.31	124.02	8.99	9.32	8.97	125.54	9.23	9.56	9.28			
segment	34.08	4.36	4.35	5.46	952.3	131.14	120.27	139.48	986.58	135.5	124.62	144.94			
shuttle	10558.17	2365.87	2266.43	2765.98	135211.31	15563.64	17533.53	15493.28	145769.48	17929.51	19899.96	18259.26			
sonar	0.88	0.14	0.13	0.16	124.69	5.77	9.5	7.82	125.57	5.91	9.63	7.98			
thyroids	0.24	0.04	0.04	0.04	28.43	2.55	2.59	2.85	28.67	2.59	2.63	2.89			
unbalanced	1.06	0.12	0.13	0.18	18.22	2.6	2.9	2.9	19.28	2.72	3.03	3.08			
vehicle	4.7	0.56	0.56	0.7	354.02	22.8	18.95	20.33	358.72	23.36	19.51	21.03			
waterform	338.58	41.97	42.17	50.08	21737.1	1566.34	1215.94	1463.72	22075.68	1408.31	1258.11	1513.8			
wdbc	3.08	0.38	0.38	0.46	261	16.99	14.48	17.02	264.08	17.37	14.86	17.48			
wis	1.77	0.2	0.2	0.28	32.77	4.57	4.48	6.03	34.54	4.77	4.68	6.31			
wpc	0.42	0.07	0.07	0.08	62.77	3.64	4.71	5.3	63.19	3.71	4.78	5.38			
yeast	7.51	0.88	0.88	1.14	392.59	30.01	25.51	25.32	400.1	30.89	26.39	26.46			

Table C.26: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 34% of the instances applied to the real-world data sets.

Data set	Precalculation time			Clustering time			Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.61	0.06	0.06	78.13	2.42	2.48	78.74	2.48	2.54	2.5
bioipn	7.29	0.62	0.62	504.59	22.02	26.36	511.88	22.64	26.98	22.3
biya	0.29	0.05	0.02	27.08	0.89	0.9	27.37	0.92	0.92	1.18
dermatology	1.26	0.12	0.12	144.82	4.66	7.18	146.08	4.78	7.3	4.43
ecoli	0.29	0.02	0.04	29.09	1.01	0.99	29.38	1.03	1.01	0.7
glass	0.14	0.01	0.01	12.63	0.58	0.51	12.77	0.59	0.52	0.39
heart-stalog	0.3	0.03	0.03	29.72	1.24	1.69	30.02	1.27	1.72	1.07
iono	1.25	0.11	0.11	116.91	3.77	3.33	118.16	3.88	3.44	4.87
iris	0.04	0	0	4.67	0.37	0.5	4.71	0.37	0.5	0.5
letter-rec	1260.31	296.53	296.46	45873.1	2547.62	2415.58	46933.41	2844.15	2712.04	3716.18
liver-disorders	0.32	0.02	0.02	34.87	0.94	1.02	35.19	0.96	1.04	1.05
magic	853.432	219.12	218.88	37539.3	2418.3	3125.71	38392.732	2637.42	3344.59	2553.88
pendigits	354.69	30.66	30.66	13734.9	884.04	732.45	14089.59	914.7	763.11	873.11
pim	1.52	0.14	0.14	124.02	5.52	5.36	125.54	5.66	5.5	4.52
segment	34.08	2.62	2.62	952.5	48.64	61.78	986.58	51.26	64.4	65.8
shuttle	10588.17	2104.32	2105.04	135211.31	8640.13	8339.64	145769.48	10744.45	10444.68	10666.89
sonar	0.88	0.07	0.08	124.69	4.18	4.7	125.57	4.25	4.78	4.05
thyroids	0.24	0.43	0.43	28.43	11.16	18.56	28.67	11.59	18.99	10.54
transfusion	1.06	0.08	0.08	18.22	1.58	1.55	19.28	1.66	1.63	1.68
vehicle	4.7	0.34	0.34	354.02	8.87	12.23	358.72	9.21	12.57	9.89
waveform	338.58	24.67	24.59	21737.1	641.98	605.79	22075.68	666.65	630.38	704.95
wdbc	3.08	0.25	0.25	261	6.67	9.03	264.08	6.92	9.28	7.29
wisconsin	1.77	0.13	0.12	32.77	2.06	2.03	34.54	2.19	2.15	2.3
wppc	0.42	0.04	0.04	62.77	3.75	2.69	63.19	3.79	2.73	2.24
yeast	7.51	0.55	0.56	392.59	11.06	13.18	400.1	11.61	13.74	12.61

Table C.27: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 25% of the instances applied to the real-world data sets.

Data set	Precalculation time				Clustering time				Overall time			
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Classes	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Classes
ball	0.61	0.04	0.04	0.04	78.13	1.24	1.06	1.15	78.74	1.28	1.1	1.27
bioipn	7.29	0.44	0.43	0.02	504.59	10.55	15.29	10.92	511.88	10.99	15.72	1.62
bpn	0.29	0.01	0.02	0.03	27.08	0.49	0.66	0.63	27.57	0.5	0.68	0.66
dermatology	1.26	0.09	0.09	0.14	144.82	2.02	1.57	1.81	146.08	2.11	1.66	1.95
ecoli	0.29	0.01	0.02	0.03	29.09	0.44	0.5	0.72	29.38	0.45	0.52	0.75
glass	0.14	0.01	0.01	0.02	12.63	0.49	0.31	0.47	12.77	0.5	0.32	0.49
heart-voting	0.3	0.02	0.02	0.04	29.72	0.66	0.51	0.68	30.02	0.68	0.53	0.72
iono	1.25	0.08	0.08	0.12	116.91	2.13	2.95	2.64	118.16	2.21	3.03	2.76
iris	0.04	0	0	0	4.67	0.32	0.3	0.18	4.71	0.32	0.3	0.18
letter-rec	1260.31	199.74	199.84	265.05	45673.1	2090.74	1426.64	1906.99	46933.41	2290.48	1626.48	2172.04
liver-disorders	0.32	0.02	0.02	0.03	34.87	0.47	0.6	0.66	35.19	0.49	0.62	0.69
magic	853.432	143.39	143.26	190.02	37539.3	2041.35	1569.66	2545.07	38392.732	2184.74	1712.92	2725.09
pendigits	354.69	20.25	20.11	33.08	13734.9	383.26	399.48	450.4	14089.99	403.51	419.59	483.48
pin	1.52	0.1	0.1	0.17	124.02	3.38	2.17	2.49	125.54	3.48	2.27	2.66
segment	34.08	1.79	1.76	2.85	952.3	41.57	36.01	44.55	986.58	43.56	37.77	47.4
shuttle	10558.17	1654.42	1653.85	1845.75	135211.31	5643.56	5893.24	5769.93	145769.48	7297.98	7547.09	7615.68
sonar	0.88	0.06	0.06	0.09	124.69	3.2	2.4	3.01	125.57	3.26	2.46	3.1
thyroids	0.24	0.3	0.3	0.6	28.43	6.64	3.73	7.81	28.67	6.94	4.03	8.41
unbalanced	1.06	0.06	0.06	0.12	18.22	0.99	0.72	1.12	19.28	1.05	0.78	1.24
vehicle	4.7	0.23	0.23	0.37	354.02	6.35	5.26	7.09	358.72	6.58	5.49	7.46
waterform	338.58	15.81	15.82	23.9	21737.1	329.17	376.91	351.39	22075.68	344.98	392.73	375.29
wdbc	3.08	0.15	0.15	0.24	261	3.67	5.06	6.61	264.08	3.82	5.21	6.85
wisconsin	1.77	0.09	0.09	0.16	32.77	1.85	1.96	0.94	34.54	1.94	2.05	1.1
wpc	0.42	0.03	0.03	0.04	62.77	1.51	1.8	1.71	63.19	1.54	1.83	1.75
yeast	7.51	0.39	0.38	0.64	392.59	8.23	7.94	8.22	400.1	8.62	8.32	8.86

Table C.28: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 20% of the instances applied to the real-world data sets.

Data set	Precalculation time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
bal	0.61	0.01	0.01	78.13	0.35	0.36	78.74	0.36	0.37
biopn	7.29	0.14	0.14	504.59	2.64	2.07	511.88	2.78	2.21
bpa	0.29	0	0.01	27.08	0.16	0.11	27.37	0.16	0.12
dermatology	1.26	0.03	0.08	144.82	0.59	0.65	146.08	0.62	0.68
ecoli	0.29	0	0.01	29.09	0.17	0.15	29.38	0.17	0.16
glass	0.14	0	0.01	12.63	0.13	0.12	12.77	0.13	0.12
heart-stalog	0.3	0.01	0	29.72	0.22	0.24	30.02	0.23	0.24
iono	1.25	0.04	0.03	116.91	0.47	0.4	118.16	0.51	0.43
iris	0.04	0	0	4.67	0.1	0.1	4.71	0.1	0.1
letter-rec	1260.31	56.62	56.49	45873.1	716.19	614.71	46933.41	772.81	671.2
liver-disorders	0.32	0	0.01	34.87	0.15	0.21	35.19	0.15	0.22
magic	853.432	41.03	41.05	37539.3	217.67	211.42	38392.732	258.7	252.47
pendigits	354.69	5.81	5.81	13734.9	104.66	68.88	14089.59	110.48	74.69
pim	1.52	0.03	0.03	124.02	0.48	0.65	125.54	0.51	0.68
segment	34.08	0.54	0.54	952.5	7.26	7.15	986.58	7.8	7.69
shuttle	10588.17	739.28	738.51	135211.31	366.5	337.86	145769.48	1105.78	1076.37
sonar	0.88	0.02	0.02	124.69	0.68	0.82	125.57	0.7	0.84
thyroids	0.24	0.09	0.09	28.43	0.8	1.3	28.67	0.89	1.39
transfusion	1.06	0.02	0.02	18.22	0.31	0.24	19.28	0.33	0.26
vehicle	4.7	0.08	0.08	354.02	0.96	0.84	358.72	1.04	0.92
waveform	338.58	5.09	5.09	21737.1	126.7	92.62	22075.68	131.79	97.71
wdbc	3.08	0.06	0.06	261	0.6	0.69	264.08	0.66	0.75
wisconsin	1.77	0.03	0.03	32.77	0.42	0.47	34.54	0.45	0.5
wppc	0.42	0.02	0.01	62.77	0.77	0.4	63.19	0.79	0.41
yeast	7.51	0.12	0.12	392.59	1.47	2.04	400.1	1.59	2.16

Table C.29: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 10% of the instances applied to the real-world data sets.

Data set	Precalculation time					Clustering time					Overall time				
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters
ball	0.61	0	0.08	0.01	0.08	78.13	0.15	0.12	0.12	0.11	78.74	0.15	0.13	0.13	0.2
bioipn	7.29	0	0.08	0.08	0.34	504.59	0.67	0.76	1.22	0.11	511.88	0.75	0.84	1.56	0.11
byrd	0.29	0	0	0	0.01	27.08	0.12	0.12	0.1	0.1	27.57	0.12	0.12	0.11	0.11
dermatology	1.26	0.02	0.02	0.02	0.07	144.82	0.35	0.37	0.43	0.39	146.08	0.37	0.39	0.5	0.11
ecoli	0.29	0	0	0	0.02	29.09	0.13	0.12	0.09	0.09	29.38	0.13	0.12	0.11	0.11
glass	0.14	0	0	0	0.01	12.65	0.13	0.11	0.09	0.11	12.77	0.13	0.11	0.1	0.11
heart-voting	0.3	0	0	0	0.02	29.72	0.13	0.14	0.13	0.13	30.02	0.13	0.14	0.15	0.15
iono	1.25	0.02	0.02	0.02	0.06	116.91	0.3	0.35	0.27	0.27	118.16	0.32	0.37	0.33	0.33
iris	0.04	0	0	0	0	4.67	0.11	0.11	0.09	0.11	4.71	0.11	0.11	0.09	0.09
letter-rec	1260.31	28.42	28.38	28.38	94	45673.1	159.53	221.87	369.84	0.14	46933.41	187.95	230.25	463.84	0.15
liver-disorders	0.32	0	0	0	0.01	34.87	0.1	0.1	0.14	0.14	35.19	0.1	0.1	0.15	0.15
magic	853.432	19.86	19.95	19.95	66.13	37539.3	82.17	88.86	203.11	42.63	38392.732	102.03	108.79	269.24	58.73
pendigits	354.69	3.09	3.08	3.08	16.1	13734.9	37.22	39.04	42.63	40.31	14089.99	40.31	42.12	58.73	0.3
pin	1.52	0.02	0.02	0.02	0.08	124.02	0.31	0.28	0.22	0.22	125.54	0.33	0.3	0.3	0.3
segment	34.08	0.3	0.31	0.31	1.41	952.3	2.47	2.33	2.46	2.46	986.58	2.77	2.64	3.87	3.87
shuttle	10558.17	353.95	353.89	353.89	817.32	135211.31	215.78	265.8	285.63	569.73	145769.48	569.73	619.69	1102.95	1102.95
sonar	0.88	0.02	0.01	0.01	0.04	124.69	0.38	0.23	0.29	0.29	125.57	0.4	0.4	0.33	0.33
thyroids	0.24	0.05	0.05	0.05	0.34	28.43	0.38	0.25	0.57	0.57	28.67	0.43	0.3	0.3	0.91
unmashon	1.06	0.01	0.01	0.01	0.06	18.22	0.12	0.13	0.17	0.17	19.28	0.13	0.14	0.23	0.23
vehicle	4.7	0.04	0.04	0.04	0.18	354.02	0.4	0.41	0.47	0.47	358.72	0.44	0.44	0.65	0.65
vehiclem	338.58	2.72	2.72	2.72	107.78	21737.1	64.85	18.95	30.87	67.57	22075.68	67.57	21.67	41.65	41.65
wdbc	3.08	0.04	0.04	0.04	0.12	261	0.44	0.61	0.35	0.35	264.08	0.48	0.65	0.47	0.47
wis	1.77	0.02	0.02	0.02	0.09	32.77	0.11	0.19	0.27	0.27	34.54	0.13	0.21	0.36	0.36
wpc	0.42	0.01	0.01	0.01	0.02	62.77	0.31	0.19	0.2	0.2	63.19	0.32	0.2	0.22	0.22
yeast	7.51	0.06	0.06	0.06	0.33	392.59	0.67	0.77	0.47	0.47	400.1	0.73	0.83	0.8	0.8

Table C.30: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 7% of the instances applied to the real-world data sets.

Data set	Precalculation time			Clustering time			Overall time		
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Random
bal	0.61	0	0.08	78.13	0.12	0.06	78.74	0.12	0.06
biopn	7.29	0.06	0.32	504.59	0.57	0.4	511.88	0.63	0.46
bpa	0.29	0	0.02	27.08	0.08	0.12	27.37	0.08	0.12
dermatology	1.26	0.02	0.06	144.82	0.23	0.27	146.08	0.25	0.29
ecoli	0.29	0	0.01	29.09	0.12	0.1	29.38	0.12	0.1
glass	0.14	0	0.01	12.63	0.08	0.1	12.77	0.08	0.1
heart-stalog	0.3	0	0.02	29.72	0.1	0.12	30.02	0.1	0.12
iono	1.25	0.01	0.06	116.91	0.24	0.26	118.16	0.25	0.28
iris	0.04	0	0	4.67	0.1	0.09	4.71	0.1	0.09
letter-rec	1260.31	17.79	83.29	45673.1	264.46	107.94	46933.41	282.25	125.69
liver-disorders	0.32	0	0.01	34.87	0.09	0.12	35.19	0.09	0.12
magic	853.432	12.12	58.37	37539.3	39.79	54.81	38392.732	51.91	66.92
pendigits	354.69	1.92	14.93	13734.9	27.57	16.08	14089.59	29.49	18
pim	1.52	0.01	0.08	124.02	0.18	0.19	125.54	0.19	0.2
segment	34.08	0.21	1.32	952.5	1.34	2.11	986.58	1.55	2.32
shuttle	10588.17	206.95	667.99	135211.31	135.81	175.76	145769.48	342.76	380.65
sonar	0.88	0.02	0.04	124.69	0.3	0.28	125.57	0.32	0.3
thyroids	0.24	0.03	0.33	28.43	0.19	0.26	28.67	0.22	0.29
transfusion	1.06	0	0.06	18.22	0.12	0.1	19.28	0.12	0.11
vehicle	4.7	0.03	0.17	354.02	0.3	0.32	358.72	0.33	0.36
waveform	338.58	1.71	9.77	21737.1	17.6	16.17	22075.68	19.31	17.88
wdbc	3.08	0.03	0.11	261	0.15	0.32	264.08	0.18	0.34
wisconsin	1.77	0.01	0.08	32.77	0.14	0.15	34.54	0.15	0.16
wppbc	0.42	0.01	0.02	62.77	0.25	0.33	63.19	0.26	0.33
yeast	7.51	0.04	0.31	392.59	0.16	0.54	400.1	0.2	0.58

Table C.31: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 5% of the instances applied to the real-world data sets.

Data set	Precalculation time						Clustering time						Overall time					
	CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters		CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters		CAOS-CD	CAOS-DS-Classes	CAOS-DS-Classes	CAOS-DS-Random	CAOS-DS-Clusters	
ball	0.61	0	0	0	0.04		78.13	0.05	0.06	0.07		78.74	0.05	0.06	0.06		0.15	
bojap	7.29	0	0	0	0		504.59	0.43	0.54	0.4		511.88	0.48	0.58	0.7		0.7	
byd	0.29	0	0	0	0.01		27.08	0.08	0.07	0.08		27.37	0.08	0.07	0.09		0.09	
dermatology	1.26	0.02	0.02	0.01	0.06		144.82	0.25	0.28	0.21		146.08	0.27	0.29	0.27		0.27	
ecoli	0.29	0	0	0	0		29.09	0.07	0.09	0.08		29.38	0.07	0.09	0.1		0.1	
glass	0.14	0	0	0	0.14		12.63	0.07	0.12	0.07		12.77	0.07	0.12	0.07		0.07	
heart-voting	0.3	0	0	0	0.2		29.72	0.11	0.11	0.1		30.02	0.11	0.11	0.11		0.12	
iono	1.25	0.02	0.02	0.01	0.06		116.91	0.2	0.2	0.28		118.16	0.22	0.21	0.21		0.34	
irs	0.04	0	0	0	0		4.67	0.11	0.07	0.11		4.71	0.11	0.07	0.11		0.11	
letter-rec	1260.31	12.23	12.23	12.22	78.21		45673.1	74.5	86.56	114.67		46933.41	86.73	98.78	192.88		192.88	
liver-disorders	0.32	0	0	0	0		34.87	0.09	0.08	0.06		35.19	0.09	0.08	0.07		0.07	
magic	853.432	8.41	8.41	8.39	54.68		37539.3	22.56	39.62	41.08		38392.732	30.97	48.01	95.76		95.76	
pendigits	354.69	1.33	1.34	1.34	14.29		13734.9	2.97	3.59	6.66		14089.99	4.3	4.93	20.95		20.95	
pin	1.52	0.01	0.01	0.01	0.08		124.02	0.12	0.16	0.14		125.54	0.13	0.17	0.22		0.22	
segment	34.08	0.16	0.16	0.16	1.26		952.3	0.37	0.73	1.12		986.58	0.53	0.89	2.38		2.38	
shuttle	10558.17	19.3	19.3	19.2	600.51		135211.31	204.63	66.07	389.27		145769.48	343.93	205.27	989.78		989.78	
sonar	0.88	0.02	0.02	0.01	0.04		124.69	0.33	0.37	0.3		125.57	0.35	0.38	0.34		0.34	
thyroids	0.24	0.02	0.02	0.02	0.32		28.43	0.32	0.27	0.29		28.67	0.34	0.29	0.61		0.61	
unifastion	1.06	0	0	0	0.06		18.22	0.09	0.11	0.1		19.28	0.09	0.11	0.16		0.16	
vehicle	4.7	0.02	0.02	0.02	0.16		354.02	0.25	0.23	0.22		358.72	0.27	0.25	0.38		0.38	
vehiclem	338.58	1.23	1.22	1.22	9.27		21737.1	4.54	9.28	12.63		22075.68	5.77	10.5	21.9		21.9	
waterform	3.08	0.02	0.02	0.02	0.11		261	0.16	0.23	0.19		264.08	0.18	0.25	0.3		0.3	
wdbc	1.77	0.01	0.01	0.01	0.08		32.77	0.11	0.11	0.11		34.54	0.12	0.12	0.19		0.19	
wdbc	0.42	0.01	0.01	0.01	0.02		62.77	0.18	0.25	0.13		63.19	0.19	0.26	0.15		0.15	
yeast	7.51	0.03	0.03	0.03	0.3		392.59	0.2	0.29	0.29		400.1	0.23	0.32	0.59		0.59	

Table C.32: Time results in seconds of CAOS with the complete data set and with the three data subset strategies using the 4% of the instances applied to the real-world data sets.

Appendix D

Full Results of the Experimentation of Chapter 7

This appendix details the results obtained by each one of the retrieval strategies of **CAOS** in the experimentation described in Chapter 7. Table [D.1](#) and Table [D.2](#) show the accuracy results for each one of the analyzed approaches for the 35 artificial data sets and the 35 real-world data sets, respectively.

Data set	Adj. Rand	Overall Pareto set					Sweet spot				
		Davies	Dunn	Silhouette	Calinski-Harabasz	Adjacent Angles	Davies	Dunn	Silhouette	Calinski-Harabasz	Adjacent Angles
100d-10c	1	1	1	1	1	1	1	1	1	1	
100d-4c	1	1	1	1	1	1	1	1	1	1	
10d-10c	1	1	1	1	1	1	1	1	1	1	
10d-4c	1	1	1	1	1	1	1	1	1	1	
2d-10c	0.93553	0.853202	0.840017	0.853202	0.846208	0.740175	0.840017	0.927175	0.927175	0.927175	
2d-4c	0.999419	0.999419	0.998258	0.999419	0.981736	0.998258	0.998258	0.998258	0.998258	0.998258	
curves1	1	1	1	1	1	1	1	1	1	1	
curves2	1	0.024606	0.251122	0.262627	0.000978	0.001555	0.229715	0.750865	0.107296	0.175355	
darbound1	0.344464	0.040497	0.330948	0.043556	0.03634	0.140704	0.330948	0.092973	0.101986	0.101986	
darbound2	0.407792	0.357639	0.360203	0.36067	0.154124	0.357639	0.360203	0.36067	0.129409	0.229638	
donut1	0.653961	0.534774	0.566907	0.588594	0.012412	0.26228	0.534774	0.566907	0.3161	0.371261	
donut2	0.598838	0.461002	0.4845	0.489267	0.01391	0.250708	0.461002	0.489267	0.139259	0.330784	
donut3	0.785241	0.746324	0.688628	0.612901	0.031999	0.746324	0.688628	0.612901	0.254472	0.406818	
donutcurves	0.61407	0.499363	0.499303	0.499248	0.038878	0.358548	0.499363	0.499303	0.314062	0.637596	
long1	1	1	1	1	1	1	1	1	1	1	
long2	1	1	1	1	1	1	1	1	1	1	
long3	1	1	1	1	1	1	1	1	1	1	
longsquare	0.969396	0.274749	0.274311	0.274749	0.18631	0.408693	0.693037	0.672189	0.418631	0.330931	
size1	0.952916	0.952916	0.952916	0.952916	0.872479	0.281794	0.801221	0.801221	0.872479	0.672963	
size2	0.973506	0.971554	0.971554	0.973506	0.952916	0.373808	0.952916	0.952916	0.952916	0.400996	
size3	0.977439	0.977439	0.977439	0.977439	0.696887	0.323168	0.971554	0.973506	0.638087	0.443709	
size4	0.979043	0.976834	0.979043	0.976834	0.53724	0.270729	0.977439	0.977439	0.53724	0.73121	
size5	0.974163	0.970727	0.971924	0.971924	0.306185	0.29238	0.976834	0.976834	0.306185	0.664296	
smile1	0.733962	0.695492	0.641617	0.599784	0.621578	0.417574	0.970841	0.97173	0.306185	0.77062	
smile2	0.627518	0.456242	0.561695	0.543392	0.044114	0.313056	0.659492	0.659492	0.650572	0.561533	
smile3	0.394334	0.276763	0.263969	0.263969	0.023313	0.240037	0.276763	0.263969	0.347439	0.435832	
spiral	0.105399	0.013206	0.051835	0.016061	0.013206	0.043892	0.034679	0.051835	0.04918	0.264597	
spiral-square	0.393365	0.339215	0.416203	0.416203	0.329237	0.339237	0.339215	0.339215	0.308875	0.659145	
square1	0.945398	0.945398	0.945398	0.945398	0.945398	0.40511	0.945398	0.945398	0.308875	0.439411	
square2	0.914691	0.914691	0.914691	0.914691	0.914691	0.569777	0.914691	0.914691	0.914691	0.439038	
square3	0.846232	0.846232	0.846232	0.846232	0.846232	0.351693	0.846232	0.846232	0.846232	0.434911	
square4	0.78319	0.780684	0.780684	0.780684	0.780684	0.448376	0.780684	0.780684	0.780684	0.447747	
square5	0.667193	0.447658	0.545734	0.221643	0.667193	0.313439	0.66709	0.66709	0.667193	0.606572	
triangle	0.99911	0.98065	0.98065	0.98065	0.98065	0.98065	0.98065	0.98065	0.98065	0.454574	
triangle2	0.989234	0.920176	0.920176	0.960785	0.784198	0.84013	0.920176	0.960785	0.784198	0.469528	
										0.947325	

Table D.1: Accuracy results of CAOS with the artificial data sets retrieving the most suitable solution from the overall Pareto set and from the sweet spot. The solutions retrieved by each clustering validation index and by the strategy based on adjacent angles are shown.

Data set	Overall Pareto set				Sweet spot					
	Adj. Rand	Dunn	Silhouette	Calinski-Harabasz	Adjacent Angles	Davies	Dunn	Silhouette	Calinski-Harabasz	Adjacent Angles
appendicitis	0.446028	0.301228	0.013668	0.278081	0.220743	0.109661	0.301228	0.036704	0.240481	0.134534
bal	0.177398	0.071411	0.000847	0.112635	0.069784	0.067813	0.071411	0.006966	0.112635	0.054389
balance	0.182572	0.000129	0.000129	0.224569	0.072459	0.015396	0.000129	0.015396	0.224569	0.099541
biopn	0.229567	0	0	0	0.000741	0	0	0	0.002979	0
bpn	0.008363	0.064388	0.000083	0.111465	0.106251	0.081206	0.081474	0.005801	0.111465	0.124677
contraceptive	0.024846	0.006111	0.000772	0.000327	0.008984	0.009383	0.008275	0.005086	0.021922	0.008108
crx	0.313192	0.000718	0.000718	0.021524	0.111308	0.093757	0.000718	0.013862	0.18954	0.186253
dermatology	0.870581	0.20502	0.20502	0.580574	0.596602	0.867683	0.206727	0.779967	0.738967	0.499758
echocardiogram	0.856982	0.432031	0.180695	0.856982	0.374247	0.432031	0.433308	0.180695	0.44728	0.477703
ecoli	0.768069	0.004373	0.004373	0.490661	0.370001	0.004373	0.004373	0.516605	0.706383	0.332946
glass	0.249486	0.007052	0.011694	0.176919	0.173842	0.23794	0.007052	0.212962	0.204151	0.167899
haberman	0.031754	0.004631	0.004631	0	0.009748	0.013397	0.004631	0.012129	0.00698	0.013825
heart-stalog	0.18857	0.054093	0.048022	0.113625	0.083586	0.054093	0.073219	0.048022	0.120103	0.093393
hepatitis	0.342289	0.169343	0.204336	0.207596	0.103493	0.169343	0.204336	0.026131	0.207596	0.119635
housevotes	0.606565	0.549293	0.593507	0.57418	0.272257	0.549293	0.531381	0.593507	0.57418	0.2253
iono	0.305868	0.004473	0.003453	0.172785	0.11587	0.069371	0.004473	0.184958	0.300692	0.126389
iris	0.74041	0.568116	0.568116	0.568116	0.524928	0.572163	0.568116	0.658849	0.74041	0.561211
liver-disorders	0.010667	0	0	0	0.003368	0	0	0	0.001256	0.000465
mnammographic	0.392694	0.375507	0.106253	0.375507	0.13475	0.379068	0.375507	0.106253	0.377035	0.123893
pendigits	0.639068	0.471841	0.487949	0.082609	0.450028	0.471841	0.440192	0.487949	0.431262	0.337076
pin	0.135545	0.002253	0.002253	0.094386	0.052413	0.058567	0.002253	0.088747	0.098949	0.059454
segment	0.553243	0.102325	0.102325	0.476451	0.374012	0.474897	0.102325	0.478719	0.503693	0.462635
sonar	0.039081	0	0	0	0.029116	0.006638	0	0.007612	0	0.021145
tae	0.040653	0.009275	0.015479	0.012684	0.025881	0.010895	0.013357	0.015479	0.015512	0.018852
thy	0.794855	0.020893	0.020893	0.581302	0.527256	0.581302	0.106743	0.556058	0.547703	0.336752
transfusion	0.034251	0.022682	0.021404	0	0.003473	0	0.022682	0	0.00689	0.010675
vehicle	0.147907	0.028223	0.028223	0.094614	0.094614	0.083841	0.055706	0.083841	0.090317	0.097204
verbal	0.275754	0	0	0.039021	0.138373	0.081641	0	0.071614	0.200631	0.173824
vowel	0.113539	0	0	0	0.046036	0.070277	0	0.055697	0.029632	0.029305
wdbc	0.753391	0.001609	0.001609	0.736666	0.367613	0.083027	0.013137	0.505468	0.617822	0.420805
wine	0.887875	0	0.264234	0.44192	0.564088	0.674095	0	0.694053	0.876616	0.534993
wisconsin	0.865161	0.276902	0.276902	0.846168	0.766498	0.839043	0.276902	0.839043	0.770198	0.593522
wybc	0.059637	0	0	0.057721	0.009014	0.000986	0	0	0.013877	0.012665
yeast	0.247476	0	0	0.15595	0.118072	0.084572	0	0.084572	0.15595	0.180571
zoo	0.950793	0.274808	0.274808	0.447982	0.491571	0.274808	0.923305	0.274808	0.66473	0.520911

Table D.2: Accuracy results of CAOS with the real-world data sets retrieving the most suitable solution from the overall Pareto set and from the sweet spot. The solutions retrieved by each clustering validation index and by the strategy based on adjacent angles are shown.

Appendix E

Full Results of the Experimentation of Chapter 10

This appendix details the results obtained by each one of the CAOS representations and by the single-objective clustering methods in the experimentation described in Chapter 10. Table E.1 to Table E.4 show the accuracy results for each one of the analyzed approaches for the 35 artificial data sets using 1, 2, 3 and 5 clusters in the retrieve phase of the CBR system, respectively. Table E.5 to Table E.8 show the number of operations done to find the results for each one of the analyzed approaches for the 35 artificial data sets using 1, 2, 3 and 5 clusters in the retrieve phase of the CBR system, respectively.

Data set	Flat	SOM 3 × 3	SOM 4 × 4	SOM 5 × 5	CAOS Davies	CAOS Dunn	CAOS Silhouette
A							
glass2c1	1.86 %	5.12 %	2.77 %	5.52 %	4.19 %	4.19 %	4.19 %
glass2c2	7.01 %	8.53 %	6.02 %	7.63 %	6.54 %	7.01 %	6.07 %
glass2c4	3.74 %	5.14 %	4.32 %	6.90 %	4.67 %	5.14 %	5.14 %
iris	4.67 %	31.33 %	8.71 %	30.45 %	5.41 %	6.08 %	6.08 %
iris2c1	4.67 %	16.00 %	5.74 %	34.10 %	6.71 %	6.71 %	6.71 %
iris2c2	0.00 %	0.00 %	0.35 %	16.95 %	0.00 %	0.00 %	0.00 %
iris2c3	5.33 %	12.33 %	6.97 %	35.05 %	5.37 %	5.37 %	5.37 %
segment	3.81 %	8.83 %	14.87 %	18.09 %	8.61 %	9.05 %	9.05 %
segment2c1	0.43 %	1.95 %	4.86 %	6.89 %	1.43 %	1.86 %	2.16 %
segment2c2	0.00 %	1.43 %	0.34 %	0.62 %	0.00 %	0.00 %	0.00 %
segment2c6	0.22 %	1.00 %	2.24 %	2.77 %	1.00 %	0.91 %	0.87 %
segment2c7	0.13 %	0.26 %	2.29 %	0.47 %	0.39 %	0.17 %	0.17 %
thy2c1	3.26 %	6.51 %	5.34 %	16.76 %	3.26 %	3.26 %	3.26 %
thy2c2	1.40 %	9.30 %	4.07 %	10.51 %	2.79 %	2.33 %	2.33 %
thyroids	5.12 %	9.77 %	8.82 %	19.02 %	5.12 %	5.12 %	5.12 %
wdbc	2.81 %	5.80 %	6.73 %	10.15 %	5.10 %	4.22 %	4.22 %
wine	3.37 %	7.30 %	39.34 %	36.86 %	2.81 %	4.49 %	4.49 %
wine2c1	1.69 %	3.93 %	13.53 %	14.69 %	2.25 %	1.69 %	1.69 %
wine2c2	2.25 %	4.49 %	17.17 %	27.74 %	2.81 %	2.25 %	2.25 %
wine2c3	3.37 %	8.99 %	18.59 %	28.93 %	2.25 %	3.37 %	3.37 %
B							
glass	32.71 %	40.42 %	34.35 %	56.44 %	35.98 %	37.38 %	36.45 %
glass2c3	8.41 %	8.06 %	10.53 %	10.41 %	7.01 %	7.01 %	7.01 %
glass2c5	20.09 %	24.77 %	20.66 %	32.20 %	21.03 %	21.50 %	20.56 %
ionosphere	13.68 %	15.67 %	14.19 %	31.76 %	16.33 %	16.62 %	15.76 %
segment2c3	1.95 %	4.94 %	4.55 %	8.40 %	4.07 %	5.02 %	5.67 %
segment2c4	1.65 %	4.11 %	4.79 %	7.52 %	3.16 %	3.38 %	3.68 %
segment2c5	3.16 %	6.97 %	7.97 %	10.70 %	6.28 %	6.97 %	7.62 %
tae	4.18 %	6.62 %	5.50 %	10.75 %	7.00 %	6.53 %	6.66 %
thy2c3	5.58 %	17.67 %	6.18 %	29.46 %	6.51 %	6.05 %	6.51 %
vehicle	30.14 %	40.69 %	40.04 %	52.54 %	32.74 %	34.16 %	38.42 %
vehicle2c1	5.67 %	13.27 %	16.78 %	19.50 %	8.63 %	8.87 %	8.75 %
vehicle2c4	3.90 %	14.07 %	17.81 %	20.80 %	9.22 %	9.46 %	9.46 %
wav2c1	14.06 %	16.04 %	16.76 %	17.19 %	16.82 %	16.24 %	16.24 %
wav2c2	16.64 %	18.88 %	18.97 %	19.98 %	19.28 %	19.12 %	19.12 %
wav2c3	13.84 %	15.52 %	15.74 %	16.36 %	16.22 %	15.58 %	15.58 %
waveform	22.54 %	24.46 %	26.63 %	26.83 %	22.14 %	22.52 %	22.52 %
wbcd	3.43 %	4.43 %	5.24 %	8.55 %	3.81 %	3.78 %	3.80 %
wisconsin	3.58 %	3.29 %	4.85 %	5.83 %	4.99 %	3.81 %	4.39 %
C							
bal	15.84 %	27.52 %	29.09 %	33.60 %	18.04 %	19.80 %	19.31 %
bal2c1	11.36 %	8.52 %	10.23 %	10.11 %	8.39 %	8.06 %	8.06 %
bal2c2	12.96 %	20.24 %	20.05 %	26.36 %	12.42 %	11.57 %	11.42 %
bal2c3	11.68 %	19.04 %	19.69 %	33.04 %	12.90 %	12.46 %	12.44 %
biopsia	17.14 %	19.77 %	22.57 %	24.15 %	18.11 %	19.28 %	18.79 %
bpa	36.81 %	43.19 %	37.00 %	46.02 %	39.42 %	42.03 %	42.03 %
glass2c6	19.63 %	37.85 %	19.81 %	37.14 %	21.96 %	22.43 %	22.90 %
heart-statlog	21.11 %	25.00 %	24.07 %	30.74 %	19.70 %	19.70 %	19.70 %
liver-disorders	37.97 %	41.45 %	41.45 %	46.67 %	42.03 %	41.74 %	41.74 %
monks-1	12.23 %	11.15 %	14.23 %	16.73 %	32.42 %	26.08 %	24.48 %
monks-2	17.97 %	18.14 %	20.80 %	18.33 %	37.60 %	39.09 %	39.02 %
monks-3	12.27 %	12.09 %	14.80 %	15.19 %	19.64 %	15.28 %	15.25 %
pim	25.91 %	28.52 %	33.24 %	31.02 %	28.39 %	28.26 %	28.91 %
sonar	17.31 %	21.15 %	21.63 %	26.44 %	23.08 %	20.19 %	21.63 %
transfusion	25.40 %	23.66 %	25.13 %	23.40 %	24.50 %	25.00 %	25.00 %
vehicle2c2	22.81 %	24.35 %	25.71 %	27.45 %	22.22 %	22.58 %	22.93 %
vehicle2c3	24.11 %	26.95 %	26.24 %	28.30 %	24.35 %	25.89 %	25.65 %
wdbc	24.75 %	27.40 %	29.82 %	28.53 %	27.78 %	29.80 %	29.80 %
rank A	1.75	5.28	4.95	6.75	3.05	3.15	3.08
position A	1	6	5	7	2	4	3
rank B	1.44	4.14	4.50	6.94	3.64	3.56	3.78
position B	1	5	6	7	3	2	4
rank C	2.22	4.03	4.78	5.72	3.61	3.81	3.83
position C	1	5	6	7	2	3	4
rank	1.80	4.51	4.75	6.48	3.42	3.49	3.54
position	1	5	6	7	2	3	4

Table E.1: Summary of the average error achieved by the CBR configurations when the most similar cluster is selected ($C=1$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

Data set	Flat	SOM 3 × 3	SOM 4 × 4	SOM 5 × 5	CAOS Davies	CAOS Dunn	CAOS Silhouette
A							
glass2c1	1.86 %	4.65 %	2.44 %	6.16 %	3.26 %	2.79 %	2.79 %
glass2c2	7.01 %	7.48 %	7.13 %	7.71 %	7.01 %	7.94 %	6.07 %
glass2c4	3.74 %	4.21 %	4.32 %	6.43 %	4.21 %	3.74 %	3.74 %
iris	4.67 %	18.67 %	7.83 %	24.33 %	4.67 %	4.67 %	4.67 %
iris2c1	4.67 %	10.67 %	4.83 %	21.00 %	4.67 %	4.67 %	4.67 %
iris2c2	0.00 %	0.00 %	0.00 %	6.17 %	0.00 %	0.00 %	0.00 %
iris2c3	5.33 %	9.50 %	5.50 %	27.00 %	4.67 %	4.67 %	4.67 %
segment	3.81 %	6.84 %	9.84 %	12.50 %	5.37 %	5.67 %	5.67 %
segment2c1	0.43 %	1.30 %	2.99 %	2.69 %	0.61 %	0.61 %	0.61 %
segment2c2	0.00 %	0.00 %	0.00 %	0.03 %	0.00 %	0.00 %	0.00 %
segment2c6	0.22 %	0.61 %	0.93 %	2.03 %	0.52 %	0.56 %	0.61 %
segment2c7	0.13 %	0.13 %	0.18 %	0.13 %	0.13 %	0.13 %	0.13 %
thy2c1	3.26 %	4.88 %	3.37 %	15.35 %	3.26 %	3.26 %	3.26 %
thy2c2	1.40 %	5.12 %	1.40 %	9.77 %	2.33 %	2.79 %	2.79 %
thyroids	5.12 %	9.30 %	5.00 %	14.65 %	5.58 %	5.12 %	5.12 %
wdbc	2.81 %	4.13 %	4.83 %	6.28 %	4.39 %	3.51 %	3.51 %
wine	3.37 %	6.18 %	28.93 %	19.10 %	3.93 %	4.49 %	4.49 %
wine2c1	1.69 %	3.37 %	7.87 %	8.01 %	1.69 %	1.69 %	1.69 %
wine2c2	2.25 %	3.37 %	12.64 %	13.20 %	2.25 %	2.25 %	2.25 %
wine2c3	3.37 %	6.74 %	15.87 %	18.26 %	2.25 %	2.25 %	2.25 %
B							
glass	32.71 %	36.45 %	33.29 %	48.13 %	34.58 %	33.18 %	33.64 %
glass2c3	8.41 %	8.29 %	8.29 %	8.64 %	7.94 %	7.94 %	7.94 %
glass2c5	20.09 %	23.83 %	20.33 %	27.57 %	21.03 %	24.30 %	24.30 %
ionsphere	13.68 %	15.95 %	13.60 %	26.07 %	15.38 %	15.38 %	15.95 %
segment2c3	1.95 %	3.59 %	3.59 %	5.66 %	2.64 %	2.73 %	2.86 %
segment2c4	1.65 %	3.24 %	3.44 %	4.66 %	2.42 %	2.68 %	2.94 %
segment2c5	3.16 %	4.92 %	6.31 %	8.03 %	5.28 %	5.63 %	5.41 %
tae	4.18 %	4.00 %	8.24 %	7.36 %	4.56 %	4.77 %	5.61 %
thy2c3	5.58 %	9.77 %	5.81 %	22.79 %	6.05 %	6.05 %	6.05 %
vehicle	30.14 %	35.08 %	35.31 %	44.59 %	30.61 %	31.80 %	35.11 %
vehicle2c1	5.67 %	8.92 %	12.53 %	14.18 %	6.62 %	6.74 %	6.86 %
vehicle2c4	3.90 %	6.15 %	12.17 %	15.43 %	4.96 %	5.67 %	5.67 %
wav2c1	14.06 %	15.14 %	16.27 %	15.97 %	15.42 %	15.58 %	15.58 %
wav2c2	16.64 %	17.45 %	17.83 %	19.04 %	21.68 %	21.58 %	21.58 %
wav2c3	13.84 %	14.72 %	15.03 %	15.56 %	15.08 %	15.04 %	15.04 %
waveform	22.54 %	24.40 %	24.77 %	25.61 %	21.68 %	21.58 %	21.58 %
wbcd	3.43 %	4.65 %	4.29 %	5.83 %	3.29 %	3.86 %	3.86 %
wisconsin	3.58 %	3.58 %	3.86 %	4.47 %	3.43 %	2.86 %	3.15 %
C							
bal	15.84 %	23.44 %	21.20 %	29.76 %	19.68 %	19.04 %	18.40 %
bal2c1	11.36 %	10.20 %	10.24 %	10.76 %	9.76 %	9.60 %	9.60 %
bal2c2	12.96 %	15.16 %	15.44 %	21.08 %	12.96 %	12.16 %	12.64 %
bal2c3	11.68 %	13.92 %	14.24 %	23.60 %	12.64 %	11.36 %	10.72 %
biopsia	17.14 %	19.86 %	21.47 %	21.79 %	18.01 %	17.14 %	18.70 %
bpa	36.81 %	42.75 %	37.83 %	45.51 %	36.52 %	37.39 %	37.39 %
glass2c6	19.63 %	33.18 %	20.56 %	32.83 %	20.09 %	20.09 %	20.09 %
heart-statlog	21.11 %	22.13 %	21.48 %	24.54 %	20.37 %	20.00 %	20.00 %
liver-disorders	37.97 %	37.97 %	40.87 %	41.16 %	37.97 %	40.29 %	40.58 %
monks-1	12.23 %	11.15 %	12.14 %	13.49 %	19.06 %	13.31 %	13.67 %
monks-2	17.97 %	18.14 %	17.05 %	18.14 %	36.94 %	37.94 %	37.77 %
monks-3	12.27 %	12.09 %	12.59 %	12.27 %	11.19 %	9.03 %	8.48 %
pim	25.91 %	28.91 %	29.39 %	30.53 %	27.86 %	27.34 %	26.17 %
sonar	17.31 %	18.27 %	20.67 %	19.23 %	18.75 %	17.31 %	17.79 %
transfusion	25.40 %	23.93 %	24.06 %	24.20 %	25.40 %	25.67 %	25.67 %
vehicle2c2	22.81 %	22.72 %	24.97 %	26.80 %	22.34 %	22.58 %	22.70 %
vehicle2c3	24.11 %	26.60 %	29.20 %	28.46 %	23.88 %	24.59 %	25.18 %
wpbc	24.75 %	27.27 %	27.90 %	28.03 %	25.25 %	24.24 %	24.24 %
rank A	2.30	5.03	4.93	6.68	3.05	3.13	2.90
position A	1	6	5	7	3	4	2
rank B	1.81	4.17	4.72	6.72	3.14	3.44	4.00
position B	1	5	6	7	2	3	4
rank C	2.92	4.36	5.17	6.17	3.39	2.86	3.14
position C	1	5	6	7	4	2	3
rank	2.34	4.54	4.94	6.53	3.19	3.14	3.33
position	1	5	6	7	3	2	4

Table E.2: Summary of the average error achieved by the CBR configurations when the two most similar clusters are selected ($C=2$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

Data set	Flat	SOM 3 × 3	SOM 4 × 4	SOM 5 × 5	CAOS Davies	CAOS Dunn	CAOS Silhouette
A							
glass2c1	1.86 %	4.19 %	2.33 %	5.93 %	2.79 %	1.86 %	1.86 %
glass2c2	7.01 %	7.48 %	6.54 %	7.71 %	7.01 %	7.48 %	6.54 %
glass2c4	3.74 %	4.21 %	4.44 %	5.72 %	3.74 %	4.21 %	3.74 %
iris	4.67 %	6.67 %	6.50 %	15.50 %	4.67 %	4.67 %	4.67 %
iris2c1	4.67 %	8.00 %	4.67 %	16.33 %	4.67 %	4.67 %	4.67 %
iris2c2	0.00 %	0.00 %	0.00 %	2.83 %	0.00 %	0.00 %	0.00 %
iris2c3	5.33 %	4.67 %	5.50 %	19.67 %	5.33 %	5.33 %	5.33 %
segment	3.81 %	5.58 %	7.88 %	10.32 %	4.37 %	4.81 %	4.81 %
segment2c1	0.43 %	1.08 %	1.10 %	1.56 %	0.61 %	0.61 %	0.52 %
segment2c2	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
segment2c6	0.22 %	0.43 %	0.69 %	1.08 %	0.26 %	0.39 %	0.35 %
segment2c7	0.13 %	0.09 %	0.14 %	0.13 %	0.13 %	0.13 %	0.13 %
thy2c1	3.26 %	4.65 %	3.37 %	12.56 %	3.26 %	2.79 %	3.26 %
thy2c2	1.40 %	4.88 %	1.40 %	8.49 %	2.33 %	2.33 %	2.33 %
thyroids	5.12 %	8.26 %	5.12 %	11.86 %	6.05 %	5.58 %	5.58 %
wdbc	2.81 %	4.17 %	3.78 %	5.67 %	3.87 %	4.04 %	4.04 %
wine	3.37 %	5.06 %	17.98 %	11.52 %	3.37 %	2.81 %	2.81 %
wine2c1	1.69 %	3.37 %	4.49 %	5.48 %	1.69 %	2.25 %	2.25 %
wine2c2	2.25 %	1.69 %	5.06 %	6.60 %	2.25 %	1.69 %	1.69 %
wine2c3	3.37 %	6.18 %	8.99 %	12.92 %	2.81 %	2.25 %	2.25 %
B							
glass	32.71 %	36.68 %	32.83 %	44.39 %	33.18 %	32.71 %	32.71 %
glass2c3	8.41 %	8.41 %	8.41 %	8.53 %	8.41 %	8.41 %	8.41 %
glass2c5	20.09 %	23.60 %	20.44 %	25.82 %	20.09 %	21.96 %	21.96 %
ionosphere	13.68 %	13.68 %	13.60 %	22.44 %	14.53 %	14.25 %	14.53 %
segment2c3	1.95 %	2.90 %	3.29 %	4.71 %	2.12 %	2.51 %	2.60 %
segment2c4	1.65 %	3.01 %	2.94 %	3.63 %	2.16 %	2.64 %	2.60 %
segment2c5	3.16 %	4.11 %	5.40 %	6.66 %	3.90 %	3.85 %	3.90 %
tae	4.18 %	4.03 %	6.04 %	6.22 %	3.60 %	3.81 %	4.87 %
thy2c3	5.58 %	8.37 %	5.47 %	20.35 %	6.05 %	6.05 %	6.05 %
vehicle	30.14 %	30.70 %	35.17 %	41.16 %	30.02 %	30.61 %	32.27 %
vehicle2c1	5.67 %	8.66 %	10.25 %	12.74 %	6.74 %	6.97 %	6.86 %
vehicle2c4	3.90 %	5.91 %	9.16 %	12.03 %	4.02 %	4.37 %	4.37 %
wav2c1	14.06 %	14.99 %	16.20 %	15.42 %	14.70 %	15.60 %	15.60 %
wav2c2	16.64 %	17.43 %	17.70 %	18.74 %	17.70 %	17.42 %	17.42 %
wav2c3	13.84 %	14.00 %	14.81 %	14.97 %	14.42 %	14.30 %	14.30 %
waveform	22.54 %	23.58 %	24.32 %	24.97 %	21.48 %	21.60 %	21.60 %
wbcd	3.43 %	4.33 %	3.65 %	4.86 %	3.29 %	3.86 %	3.86 %
wisconsin	3.58 %	3.58 %	4.01 %	4.69 %	3.00 %	3.00 %	3.29 %
C							
bal	15.84 %	22.00 %	19.92 %	28.00 %	20.00 %	18.88 %	17.92 %
bal2c1	11.36 %	10.96 %	10.72 %	11.52 %	9.76 %	10.40 %	10.40 %
bal2c2	12.96 %	14.56 %	16.20 %	18.16 %	13.12 %	12.48 %	11.84 %
bal2c3	11.68 %	12.00 %	14.40 %	19.60 %	12.80 %	12.16 %	12.48 %
biopsia	17.14 %	19.04 %	20.37 %	20.93 %	17.43 %	17.33 %	17.92 %
bpa	36.81 %	42.03 %	36.59 %	44.06 %	37.68 %	37.39 %	37.39 %
glass2c6	19.63 %	27.57 %	19.74 %	28.86 %	20.09 %	21.03 %	21.50 %
heart-statlog	21.11 %	21.20 %	21.57 %	22.22 %	20.74 %	20.74 %	20.74 %
liver-disorders	37.97 %	37.68 %	37.68 %	37.68 %	37.97 %	40.00 %	40.00 %
monks-1	12.23 %	11.15 %	11.69 %	12.23 %	12.77 %	11.87 %	12.41 %
monks-2	17.97 %	18.30 %	17.72 %	18.80 %	38.10 %	37.10 %	38.10 %
monks-3	12.27 %	12.09 %	11.96 %	11.91 %	8.84 %	10.29 %	10.11 %
pim	25.91 %	28.22 %	29.04 %	29.04 %	26.69 %	26.30 %	26.17 %
sonar	17.31 %	20.19 %	17.79 %	19.71 %	17.31 %	17.31 %	17.31 %
transfusion	25.40 %	24.87 %	24.73 %	25.13 %	25.94 %	25.00 %	25.00 %
vehicle2c2	22.81 %	22.22 %	24.35 %	25.68 %	21.99 %	22.81 %	23.05 %
vehicle2c3	24.11 %	26.30 %	24.82 %	26.57 %	23.76 %	25.06 %	25.53 %
wpbcc	24.75 %	25.76 %	27.90 %	28.79 %	24.75 %	25.25 %	25.25 %
rank A	2.58	4.68	4.65	6.65	3.30	3.25	2.90
position A	1	6	5	7	4	3	2
rank B	2.00	4.53	4.83	6.83	2.72	3.31	3.78
position B	1	5	6	7	2	3	4
rank C	2.97	4.39	4.03	6.06	3.56	3.31	3.69
position C	1	6	5	7	3	2	4
rank	2.52	4.54	4.51	6.52	3.20	3.29	3.44
position	1	6	5	7	2	3	4

Table E.3: Summary of the average error achieved by the CBR configurations when the three most similar clusters are selected ($C=3$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

Data set	Flat	SOM	SOM	SOM	CAOS	CAOS	CAOS
		3 × 3	4 × 4	5 × 5	Davies	Dunn	Silhouette
A							
glass2c1	1.86 %	2.79 %	1.86 %	4.53 %	1.86 %	2.33 %	2.33 %
glass2c2	7.01 %	7.13 %	7.01 %	7.24 %	7.01 %	7.01 %	7.01 %
glass2c4	3.74 %	3.74 %	3.97 %	5.14 %	3.74 %	3.74 %	3.74 %
iris	4.67 %	4.67 %	4.67 %	11.33 %	4.67 %	4.67 %	4.67 %
iris2c1	4.67 %	4.00 %	4.67 %	11.17 %	4.67 %	4.67 %	4.67 %
iris2c2	0.00 %	0.00 %	0.00 %	0.17 %	0.00 %	0.00 %	0.00 %
iris2c3	5.33 %	6.33 %	5.33 %	12.00 %	5.33 %	5.33 %	5.33 %
segment	3.81 %	4.72 %	6.34 %	7.55 %	3.90 %	4.20 %	4.20 %
segment2c1	0.43 %	0.78 %	0.73 %	1.11 %	0.43 %	0.52 %	0.52 %
segment2c2	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
segment2c6	0.22 %	0.38 %	0.51 %	0.76 %	0.22 %	0.17 %	0.17 %
segment2c7	0.13 %	0.13 %	0.11 %	0.13 %	0.13 %	0.13 %	0.13 %
thy2c1	3.26 %	3.72 %	3.26 %	10.23 %	3.26 %	3.26 %	3.26 %
thy2c2	1.40 %	3.26 %	1.40 %	6.28 %	1.40 %	1.86 %	1.86 %
thyroids	5.12 %	5.58 %	5.12 %	7.79 %	5.12 %	5.12 %	5.12 %
wdbc	2.81 %	3.16 %	3.65 %	4.61 %	2.99 %	3.16 %	3.16 %
wine	3.37 %	5.62 %	7.02 %	5.48 %	3.37 %	3.37 %	3.37 %
wine2c1	1.69 %	3.37 %	2.95 %	3.65 %	1.69 %	1.12 %	1.12 %
wine2c2	2.25 %	2.81 %	2.11 %	3.09 %	2.25 %	2.25 %	2.25 %
wine2c3	3.37 %	5.06 %	5.34 %	8.43 %	3.37 %	3.37 %	3.37 %
B							
glass	32.71 %	35.28 %	33.18 %	40.89 %	32.71 %	32.71 %	32.71 %
glass2c3	8.41 %	9.35 %	8.41 %	8.18 %	8.41 %	8.41 %	8.41 %
glass2c5	20.09 %	21.50 %	20.09 %	23.36 %	20.09 %	21.50 %	21.50 %
ionsphere	13.68 %	12.82 %	13.68 %	19.94 %	13.68 %	13.68 %	13.68 %
segment2c3	1.95 %	2.42 %	2.99 %	3.73 %	1.95 %	2.08 %	2.08 %
segment2c4	1.65 %	2.29 %	2.40 %	2.80 %	1.56 %	1.82 %	1.82 %
segment2c5	3.16 %	3.46 %	4.81 %	5.76 %	2.99 %	3.07 %	3.07 %
tae	4.18 %	4.12 %	4.37 %	4.78 %	4.13 %	4.13 %	4.18 %
thy2c3	5.58 %	6.98 %	5.58 %	14.77 %	5.58 %	6.05 %	6.05 %
vehicle	30.14 %	30.41 %	33.69 %	35.55 %	29.91 %	30.02 %	30.02 %
vehicle2c1	5.67 %	6.89 %	8.89 %	10.49 %	5.67 %	5.56 %	5.67 %
vehicle2c4	3.90 %	5.08 %	6.94 %	8.57 %	3.90 %	4.02 %	4.02 %
wav2c1	14.06 %	14.56 %	15.10 %	14.61 %	14.20 %	14.34 %	14.34 %
wav2c2	16.64 %	17.18 %	17.99 %	18.28 %	17.04 %	16.88 %	16.88 %
wav2c3	13.84 %	13.94 %	14.25 %	14.32 %	14.00 %	13.78 %	13.78 %
waveform	22.54 %	22.52 %	23.12 %	24.48 %	22.04 %	22.08 %	22.08 %
wbcd	3.43 %	3.43 %	3.15 %	4.33 %	3.58 %	3.58 %	3.58 %
wisconsin	3.58 %	3.97 %	3.90 %	4.01 %	3.29 %	3.29 %	3.29 %
C							
bal	15.84 %	21.48 %	20.64 %	25.12 %	19.04 %	18.56 %	18.40 %
bal2c1	11.36 %	11.04 %	10.88 %	10.80 %	8.96 %	9.92 %	9.92 %
bal2c2	12.96 %	12.04 %	12.96 %	15.16 %	12.32 %	12.80 %	11.84 %
bal2c3	11.68 %	10.72 %	11.52 %	17.20 %	12.48 %	13.44 %	13.12 %
biopsia	17.14 %	17.82 %	19.67 %	19.96 %	16.75 %	17.14 %	17.04 %
bpa	36.81 %	41.16 %	36.81 %	39.71 %	37.10 %	37.10 %	37.10 %
glass2c6	19.63 %	19.16 %	19.63 %	25.82 %	19.63 %	21.03 %	21.50 %
heart-statlog	21.11 %	22.13 %	19.91 %	22.96 %	20.74 %	20.37 %	20.37 %
liver-disorders	37.97 %	37.39 %	36.81 %	38.26 %	37.97 %	37.68 %	37.39 %
monks-1	12.23 %	11.15 %	12.28 %	12.05 %	9.89 %	6.65 %	7.01 %
monks-2	17.97 %	18.30 %	17.72 %	18.14 %	37.94 %	37.60 %	37.77 %
monks-3	12.27 %	12.09 %	11.91 %	12.27 %	9.03 %	9.03 %	9.39 %
pim	25.91 %	27.21 %	28.74 %	28.68 %	25.91 %	25.91 %	26.30 %
sonar	17.31 %	17.79 %	17.79 %	18.75 %	17.31 %	17.31 %	17.31 %
transfusion	25.40 %	25.13 %	25.13 %	25.00 %	25.53 %	25.53 %	25.53 %
vehicle2c2	22.81 %	22.61 %	23.35 %	24.94 %	23.05 %	23.17 %	23.29 %
vehicle2c3	24.11 %	25.65 %	25.27 %	26.51 %	24.11 %	24.23 %	24.82 %
wdbc	24.75 %	25.51 %	26.14 %	27.90 %	24.75 %	24.24 %	24.24 %
rank A	2.88	4.93	4.00	6.63	2.98	3.30	3.30
position A	1	6	5	7	2	3	3
rank B	2.86	4.58	5.00	6.61	2.50	3.11	3.33
position B	1	5	6	7	2	3	4
rank C	3.61	4.14	4.11	6.03	3.31	3.39	3.42
position C	1	6	5	7	2	3	4
rank	3.11	4.56	4.36	6.43	2.93	3.27	3.35
position	1	6	5	7	2	3	4

Table E.4: Summary of the average error achieved by the CBR configurations when the five most similar clusters are selected ($C=5$). The last rows show the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

Data set	Flat (Case Memory)	SOM	SOM	SOM	CAOS	CAOS	CAOS
		3 × 3	4 × 4	5 × 5	Davies	Dunn	Silhouette
A							
glass2c1	192	20.31 %	55.73 %	17.71 %	36.98 %	28.13 %	28.13 %
glass2c2	192	17.71 %	54.17 %	17.71 %	34.90 %	29.69 %	22.40 %
glass2c4	192	19.27 %	56.77 %	17.19 %	33.33 %	25.52 %	25.52 %
iris	135	14.81 %	50.37 %	21.48 %	35.56 %	35.56 %	35.56 %
iris2c1	135	21.48 %	60.74 %	20.74 %	28.89 %	28.89 %	28.89 %
iris2c2	135	24.44 %	74.81 %	21.48 %	28.15 %	28.15 %	28.15 %
iris2c3	135	22.96 %	62.22 %	20.74 %	27.41 %	26.67 %	26.67 %
segment	2079	16.69 %	13.95 %	6.83 %	20.25 %	16.35 %	16.35 %
segment2c1	2079	15.39 %	24.96 %	6.30 %	19.87 %	15.20 %	12.46 %
segment2c2	2079	15.06 %	15.15 %	6.06 %	19.10 %	16.50 %	14.91 %
segment2c6	2079	15.63 %	17.51 %	6.49 %	21.60 %	16.55 %	14.72 %
segment2c7	2079	15.68 %	20.49 %	6.78 %	19.10 %	14.67 %	13.56 %
thy2c1	193	18.13 %	80.31 %	16.06 %	30.57 %	20.73 %	20.73 %
thy2c2	193	17.10 %	78.76 %	16.58 %	31.09 %	21.76 %	21.76 %
thyroids	193	18.65 %	78.24 %	16.06 %	30.05 %	27.98 %	27.98 %
wdbc	512	15.82 %	41.60 %	9.57 %	25.00 %	20.51 %	20.51 %
wine	160	17.50 %	26.88 %	18.13 %	28.13 %	25.00 %	25.00 %
wine2c1	160	18.13 %	43.75 %	18.75 %	27.50 %	25.00 %	25.00 %
wine2c2	160	18.13 %	32.50 %	18.13 %	27.50 %	23.75 %	23.75 %
wine2c3	160	17.50 %	33.75 %	18.13 %	27.50 %	24.38 %	24.38 %
B							
glass	192	21.88 %	59.38 %	18.23 %	43.23 %	53.13 %	50.00 %
glass2c3	192	19.79 %	57.81 %	17.19 %	39.06 %	28.65 %	25.52 %
glass2c5	192	20.83 %	54.69 %	18.75 %	35.94 %	27.60 %	22.92 %
ionosphere	315	18.73 %	54.92 %	12.38 %	30.48 %	31.75 %	28.89 %
segment2c3	2079	14.96 %	28.81 %	6.11 %	19.87 %	15.15 %	13.23 %
segment2c4	2079	15.15 %	19.14 %	7.26 %	21.02 %	14.62 %	11.64 %
segment2c5	2079	15.97 %	15.25 %	7.12 %	17.51 %	12.94 %	12.22 %
tae	1699	16.24 %	23.48 %	7.65 %	12.71 %	16.07 %	6.77 %
thy2c3	193	18.13 %	77.72 %	16.06 %	31.61 %	22.28 %	21.24 %
vehicle	761	17.87 %	31.67 %	9.33 %	24.05 %	25.10 %	7.88 %
vehicle2c1	761	17.74 %	16.82 %	8.94 %	24.31 %	21.94 %	20.76 %
vehicle2c4	761	18.00 %	13.27 %	8.80 %	24.05 %	20.76 %	20.76 %
wav2c1	4500	15.47 %	10.29 %	6.40 %	13.24 %	11.42 %	11.42 %
wav2c2	4500	16.13 %	8.76 %	6.40 %	19.73 %	10.16 %	10.16 %
wav2c3	4500	16.13 %	9.09 %	7.09 %	13.84 %	10.33 %	10.33 %
waveform	4499	17.18 %	9.67 %	6.18 %	17.27 %	15.31 %	15.31 %
wbcd	629	18.60 %	15.74 %	8.74 %	26.07 %	22.58 %	21.78 %
wisconsin	629	19.87 %	12.56 %	10.33 %	25.60 %	21.94 %	19.40 %
C							
bal	562	15.12 %	12.99 %	8.36 %	20.46 %	12.10 %	11.74 %
bal2c1	562	15.84 %	12.81 %	8.36 %	24.02 %	11.92 %	11.92 %
bal2c2	562	15.48 %	14.23 %	8.54 %	23.13 %	11.74 %	11.21 %
bal2c3	562	15.66 %	14.77 %	7.65 %	22.78 %	11.92 %	11.92 %
biopsia	924	18.18 %	12.12 %	8.23 %	21.00 %	22.08 %	7.58 %
bpa	310	20.32 %	64.84 %	14.84 %	27.10 %	20.97 %	20.97 %
glass2c6	192	20.31 %	63.02 %	17.71 %	36.98 %	31.25 %	26.04 %
heart-stalog	243	17.28 %	16.46 %	14.40 %	25.51 %	18.93 %	18.93 %
liver-disorders	310	16.45 %	11.94 %	12.58 %	33.55 %	21.94 %	19.68 %
monks-1	500	25.60 %	13.60 %	10.40 %	8.80 %	7.60 %	7.20 %
monks-2	540	25.56 %	13.33 %	9.81 %	10.74 %	6.85 %	6.48 %
monks-3	498	25.70 %	14.06 %	10.24 %	12.85 %	7.23 %	7.03 %
pim	691	16.50 %	11.87 %	8.54 %	24.46 %	18.81 %	15.34 %
sonar	187	17.65 %	14.97 %	17.65 %	43.85 %	49.73 %	46.52 %
transfusion	673	16.05 %	10.55 %	9.06 %	24.22 %	18.28 %	18.28 %
vehicle2c2	761	17.35 %	29.43 %	9.33 %	22.73 %	19.32 %	18.79 %
vehicle2c3	761	19.19 %	20.11 %	9.07 %	25.76 %	20.50 %	18.27 %
wpc	178	16.85 %	29.21 %	17.42 %	28.65 %	20.79 %	20.79 %
operations A	100.00 %	18.02 %	46.13 %	15.05 %	27.62 %	23.55 %	22.82 %
rank A	7.00	2.20	5.60	1.25	5.05	3.65	3.25
position A	7	2	6	1	5	4	3
operations B	100.00 %	15.93 %	25.95 %	9.15 %	21.98 %	19.09 %	16.51 %
rank B	7.00	3.61	4.06	1.11	5.11	4.14	2.97
position B	7	3	4	1	6	5	2
operations C	100.00 %	16.75 %	19.02 %	10.11 %	21.83 %	16.60 %	14.93 %
rank C	7.00	3.75	3.89	1.64	5.22	3.78	2.72
position C	7	3	5	1	6	4	2
operations	100.00 %	18.11 %	32.54 %	12.25 %	25.51 %	21.15 %	19.38 %
rank	7.00	3.15	4.55	1.33	5.13	3.85	2.99
position	7	3	5	1	6	4	2

Table E.5: Summary of the average number of cases used by the CBR configurations when the most similar cluster is selected ($C=1$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A , B , C), and for all the data sets.

Data set	Flat (Case Memory)	SOM 3 × 3	SOM 4 × 4	SOM 5 × 5	CAOS Davies	CAOS Dunn	CAOS Silhouette
A							
glass2c1	192	35.42 %	71.35 %	21.88 %	57.29 %	43.75 %	43.75 %
glass2c2	192	31.77 %	69.27 %	22.92 %	59.38 %	47.40 %	34.90 %
glass2c4	192	32.81 %	71.88 %	21.88 %	52.08 %	41.15 %	41.15 %
iris	135	29.63 %	65.19 %	26.67 %	70.37 %	70.37 %	70.37 %
iris2c1	135	37.04 %	72.59 %	26.67 %	45.93 %	45.93 %	45.93 %
iris2c2	135	39.26 %	81.48 %	26.67 %	49.63 %	49.63 %	49.63 %
iris2c3	135	38.52 %	73.33 %	25.93 %	48.89 %	46.67 %	46.67 %
segment	2079	30.06 %	22.17 %	11.45 %	40.74 %	34.30 %	34.30 %
segment2c1	2079	30.11 %	32.37 %	11.01 %	36.60 %	30.16 %	26.12 %
segment2c2	2079	30.50 %	25.83 %	11.16 %	37.33 %	31.60 %	26.65 %
segment2c6	2079	30.45 %	27.13 %	10.82 %	40.36 %	31.55 %	29.63 %
segment2c7	2079	30.69 %	27.75 %	11.01 %	36.22 %	27.03 %	25.30 %
thy2c1	193	33.68 %	92.75 %	20.21 %	60.62 %	37.82 %	37.31 %
thy2c2	193	32.12 %	90.67 %	21.24 %	59.59 %	39.38 %	39.38 %
thyroids	193	33.16 %	91.71 %	20.21 %	57.51 %	52.85 %	52.85 %
wdbc	512	32.23 %	49.41 %	14.26 %	43.16 %	35.74 %	35.74 %
wine	160	31.25 %	41.25 %	22.50 %	45.00 %	37.50 %	37.50 %
wine2c1	160	31.25 %	62.50 %	22.50 %	41.88 %	36.25 %	36.25 %
wine2c2	160	31.88 %	49.38 %	22.50 %	44.38 %	34.38 %	34.38 %
wine2c3	160	32.50 %	50.00 %	22.50 %	43.13 %	36.25 %	36.25 %
B							
glass	192	38.54 %	73.96 %	22.92 %	73.44 %	89.58 %	82.81 %
glass2c3	192	34.38 %	71.88 %	22.40 %	63.02 %	47.40 %	41.15 %
glass2c5	192	33.85 %	69.27 %	23.44 %	57.29 %	42.71 %	34.90 %
ionosphere	315	36.83 %	79.05 %	17.46 %	51.75 %	53.33 %	53.97 %
segment2c3	2079	30.21 %	37.47 %	10.63 %	36.56 %	28.72 %	25.59 %
segment2c4	2079	31.12 %	27.71 %	11.64 %	38.67 %	28.04 %	22.17 %
segment2c5	2079	30.64 %	23.71 %	11.83 %	32.47 %	25.01 %	23.95 %
tae	1699	31.14 %	38.32 %	12.48 %	23.66 %	30.25 %	12.65 %
thy2c3	193	33.16 %	90.67 %	20.73 %	62.69 %	38.86 %	37.82 %
vehicle	761	31.27 %	39.55 %	13.93 %	45.07 %	48.09 %	13.27 %
vehicle2c1	761	31.93 %	24.18 %	13.93 %	40.60 %	38.90 %	36.53 %
vehicle2c4	761	34.03 %	22.86 %	13.53 %	39.68 %	32.72 %	32.72 %
wav2c1	4500	30.64 %	17.51 %	11.87 %	32.51 %	26.62 %	26.62 %
wav2c2	4500	31.49 %	16.04 %	11.16 %	28.76 %	27.58 %	27.58 %
wav2c3	4500	32.20 %	16.64 %	12.31 %	35.02 %	26.18 %	26.18 %
waveform	4499	32.23 %	17.36 %	10.96 %	28.76 %	27.58 %	27.58 %
wbcd	629	34.66 %	25.12 %	12.56 %	47.22 %	41.02 %	40.06 %
wisconsin	629	31.64 %	22.58 %	14.94 %	48.01 %	40.22 %	34.50 %
C							
bal	562	26.69 %	24.20 %	11.74 %	42.70 %	22.95 %	22.06 %
bal2c1	562	27.40 %	22.42 %	11.92 %	43.59 %	22.06 %	22.06 %
bal2c2	562	28.29 %	22.06 %	12.28 %	45.37 %	22.60 %	21.17 %
bal2c3	562	28.83 %	21.89 %	11.57 %	45.55 %	22.42 %	21.89 %
biopsia	924	35.17 %	20.35 %	13.53 %	37.88 %	39.29 %	13.31 %
bpa	310	36.77 %	78.39 %	20.97 %	50.97 %	37.10 %	37.10 %
glass2c6	192	34.90 %	75.52 %	22.40 %	61.46 %	51.04 %	41.15 %
heart-statlog	243	30.45 %	23.05 %	19.75 %	41.15 %	31.69 %	31.69 %
liver-disorders	310	27.10 %	18.71 %	16.77 %	55.81 %	37.74 %	34.19 %
monks-1	500	50.60 %	20.00 %	15.00 %	22.00 %	14.60 %	13.00 %
monks-2	540	50.56 %	19.81 %	14.63 %	21.11 %	13.70 %	12.59 %
monks-3	498	50.80 %	19.88 %	15.06 %	28.31 %	12.85 %	12.45 %
pim	691	32.13 %	19.97 %	13.31 %	42.55 %	34.59 %	27.06 %
sonar	187	27.27 %	21.39 %	21.93 %	83.42 %	90.37 %	86.10 %
transfusion	673	25.85 %	16.64 %	13.08 %	45.77 %	34.47 %	34.47 %
vehicle2c2	761	32.19 %	36.14 %	14.06 %	41.92 %	34.95 %	33.64 %
vehicle2c3	761	33.11 %	27.60 %	13.67 %	41.52 %	35.22 %	31.67 %
wdbc	178	29.21 %	39.89 %	22.47 %	48.88 %	34.83 %	34.83 %
operations A	100.00 %	32.72 %	58.40 %	19.70 %	48.50 %	40.48 %	39.20 %
rank A	7.00	2.45	5.05	1.00	5.20	3.88	3.43
position A	7	2	5	1	6	4	3
operations B	100.00 %	29.50 %	35.69 %	13.44 %	39.26 %	34.64 %	30.00 %
rank B	7.00	3.78	3.61	1.06	5.11	4.19	3.25
position B	7	4	3	1	6	5	2
operations C	100.00 %	30.37 %	26.39 %	14.21 %	40.00 %	29.62 %	26.52 %
rank C	7.00	3.89	3.42	1.44	5.56	3.92	2.78
position C	7	4	3	1	6	5	2
operations	100.00 %	33.06 %	43.03 %	16.91 %	45.63 %	37.41 %	34.19 %
rank	7.00	3.34	4.06	1.16	5.29	3.99	3.16
position	7	3	5	1	6	4	2

Table E.6: Summary of the average number of cases used by the CBR configurations when the two most similar clusters are selected ($C=2$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

Data set	Flat (Case Memory)	SOM			CAOS		
		3 × 3	4 × 4	5 × 5	Davies	Dunn	Silhouette
A							
glass2c1	192	47.40 %	80.21 %	26.56 %	77.08 %	57.29 %	57.29 %
glass2c2	192	45.31 %	76.56 %	27.08 %	76.04 %	62.50 %	45.83 %
glass2c4	192	45.83 %	84.38 %	26.56 %	69.79 %	54.69 %	54.17 %
iris	135	42.22 %	74.81 %	31.85 %	100.00 %	100.00 %	100.00 %
iris2c1	135	49.63 %	80.74 %	32.59 %	62.96 %	62.96 %	62.96 %
iris2c2	135	51.11 %	87.41 %	32.59 %	70.37 %	70.37 %	70.37 %
iris2c3	135	50.37 %	79.26 %	30.37 %	64.44 %	60.74 %	60.74 %
segment	2079	43.43 %	28.67 %	15.87 %	58.78 %	49.16 %	49.16 %
segment2c1	2079	43.00 %	48.73 %	15.68 %	54.26 %	43.10 %	37.09 %
segment2c2	2079	43.53 %	33.81 %	15.87 %	50.70 %	45.60 %	39.25 %
segment2c6	2079	43.67 %	34.97 %	15.06 %	58.01 %	46.75 %	43.19 %
segment2c7	2079	43.72 %	34.25 %	15.73 %	55.99 %	40.16 %	36.99 %
thy2c1	193	48.19 %	95.34 %	24.87 %	81.35 %	54.40 %	53.89 %
thy2c2	193	47.15 %	94.82 %	26.42 %	79.27 %	60.10 %	60.10 %
thyroids	193	47.15 %	96.89 %	24.87 %	79.79 %	73.58 %	73.58 %
wdbc	512	46.09 %	55.08 %	18.75 %	59.38 %	47.66 %	47.66 %
wine	160	44.38 %	57.50 %	26.88 %	61.25 %	48.13 %	48.13 %
wine2c1	160	46.25 %	69.38 %	27.50 %	55.63 %	46.25 %	46.25 %
wine2c2	160	46.25 %	72.50 %	26.88 %	63.13 %	45.00 %	45.00 %
wine2c3	160	45.63 %	76.25 %	26.88 %	56.88 %	46.25 %	46.25 %
B							
glass	192	52.08 %	81.77 %	28.13 %	95.31 %	100.00 %	100.00 %
glass2c3	192	47.40 %	79.69 %	27.08 %	83.33 %	61.98 %	53.65 %
glass2c5	192	46.35 %	79.17 %	28.13 %	76.04 %	54.69 %	46.88 %
ionosphere	315	48.89 %	88.25 %	21.90 %	71.11 %	75.87 %	74.60 %
segment2c3	2079	43.63 %	43.53 %	15.01 %	53.44 %	43.19 %	38.24 %
segment2c4	2079	44.49 %	35.16 %	16.45 %	53.63 %	42.14 %	33.62 %
segment2c5	2079	43.39 %	30.30 %	16.26 %	47.57 %	41.03 %	37.71 %
tae	1699	44.32 %	48.97 %	16.83 %	37.14 %	47.50 %	18.60 %
thy2c3	193	47.15 %	93.26 %	25.39 %	79.79 %	54.40 %	53.37 %
vehicle	761	44.55 %	46.12 %	18.27 %	66.23 %	69.12 %	18.79 %
vehicle2c1	761	44.55 %	31.27 %	18.40 %	59.66 %	56.90 %	53.09 %
vehicle2c4	761	47.04 %	29.30 %	18.13 %	54.27 %	44.28 %	44.28 %
wav2c1	4500	45.11 %	24.40 %	16.73 %	57.33 %	45.76 %	45.76 %
wav2c2	4500	45.00 %	22.78 %	15.96 %	55.36 %	42.09 %	42.09 %
wav2c3	4500	46.18 %	23.36 %	17.42 %	56.82 %	46.51 %	46.51 %
waveform	4499	45.99 %	23.54 %	15.80 %	38.83 %	36.81 %	36.81 %
wbcd	629	46.90 %	33.07 %	16.69 %	56.76 %	51.51 %	49.13 %
wisconsin	629	44.67 %	27.98 %	19.24 %	57.87 %	50.08 %	43.88 %
C							
bal	562	38.26 %	32.92 %	15.84 %	60.14 %	33.45 %	32.21 %
bal2c1	562	39.32 %	30.96 %	16.19 %	61.57 %	32.03 %	32.03 %
bal2c2	562	40.21 %	28.29 %	16.19 %	63.70 %	33.10 %	30.96 %
bal2c3	562	39.50 %	28.65 %	15.30 %	63.88 %	32.56 %	32.03 %
biopsia	924	49.03 %	27.60 %	18.51 %	54.87 %	55.95 %	19.05 %
bpa	310	50.65 %	94.19 %	26.45 %	70.97 %	52.58 %	52.58 %
glass2c6	192	48.44 %	83.85 %	27.08 %	79.17 %	66.15 %	53.65 %
heart-statlog	243	42.80 %	30.04 %	23.87 %	57.20 %	44.44 %	44.44 %
liver-disorders	310	38.06 %	24.84 %	20.32 %	72.26 %	52.58 %	48.71 %
monks-1	500	75.80 %	25.20 %	19.20 %	31.60 %	20.80 %	18.40 %
monks-2	540	75.74 %	24.44 %	18.89 %	28.15 %	18.89 %	17.59 %
monks-3	498	75.70 %	24.90 %	19.28 %	36.95 %	18.07 %	17.27 %
pim	691	45.73 %	27.06 %	18.09 %	58.61 %	48.34 %	37.92 %
sonar	187	37.43 %	27.27 %	25.67 %	99.47 %	99.47 %	97.86 %
transfusion	673	35.96 %	22.59 %	16.49 %	65.23 %	53.49 %	53.49 %
vehicle2c2	761	44.94 %	41.66 %	18.79 %	60.45 %	49.54 %	47.83 %
vehicle2c3	761	46.91 %	40.47 %	18.40 %	62.55 %	51.64 %	45.86 %
wdbc	178	41.01 %	46.07 %	27.53 %	69.10 %	46.07 %	46.07 %
operations A	100.00 %	46.02 %	68.08 %	24.44 %	66.75 %	55.73 %	53.89 %
rank A	6.93	2.60	4.90	1.00	5.28	3.90	3.40
position A	7	2	5	1	6	4	3
operations B	100.00 %	41.38 %	42.10 %	17.59 %	55.02 %	48.19 %	41.85 %
rank B	6.94	3.56	3.39	1.00	5.33	4.42	3.36
position B	7	3	4	1	6	5	2
operations C	100.00 %	43.27 %	33.05 %	18.10 %	54.79 %	40.46 %	36.40 %
rank C	7.00	3.94	3.00	1.25	5.64	4.17	3.00
position C	7	4	2	1	6	5	2
operations	100.00 %	46.67 %	51.15 %	21.48 %	63.06 %	51.57 %	47.19 %
rank	6.91	3.34	3.80	1.08	5.42	4.17	3.28
position	7	3	4	1	6	5	2

Table E.7: Summary of the average number of cases used by the CBR configurations when the three most similar clusters are selected ($C=3$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A , B , C), and for all the data sets.

Data set	Flat (Case Memory)	SOM 3 × 3	SOM 4 × 4	SOM 5 × 5	CAOS Davies	CAOS Dunn	CAOS Silhouette
A							
glass2c1	192	71.88 %	95.31 %	35.42 %	96.35 %	82.81 %	82.81 %
glass2c2	192	70.83 %	92.19 %	35.94 %	99.48 %	85.42 %	66.67 %
glass2c4	192	72.92 %	95.31 %	34.90 %	95.31 %	77.60 %	76.04 %
iris	135	68.89 %	88.15 %	40.00 %	100.00 %	100.00 %	100.00 %
iris2c1	135	74.81 %	90.37 %	42.22 %	100.00 %	100.00 %	100.00 %
iris2c2	135	73.33 %	94.81 %	42.22 %	100.00 %	100.00 %	100.00 %
iris2c3	135	76.30 %	91.85 %	40.74 %	100.00 %	99.26 %	99.26 %
segment	2079	67.05 %	40.16 %	24.39 %	89.51 %	75.08 %	75.08 %
segment2c1	2079	68.54 %	58.06 %	24.34 %	84.18 %	66.38 %	58.39 %
segment2c2	2079	70.47 %	46.03 %	25.16 %	85.76 %	74.75 %	63.73 %
segment2c6	2079	68.64 %	49.40 %	24.24 %	87.93 %	73.26 %	68.93 %
segment2c7	2079	68.98 %	45.94 %	24.58 %	88.41 %	63.35 %	58.20 %
thy2c1	193	75.13 %	100.00 %	34.20 %	97.93 %	79.27 %	78.76 %
thy2c2	193	74.09 %	100.00 %	36.27 %	97.93 %	83.94 %	83.94 %
thyroids	193	77.72 %	100.00 %	34.20 %	98.96 %	94.82 %	94.82 %
wdbc	512	71.29 %	64.06 %	28.52 %	90.04 %	69.73 %	69.73 %
wine	160	71.88 %	82.50 %	36.88 %	87.50 %	72.50 %	72.50 %
wine2c1	160	71.25 %	82.50 %	35.63 %	88.75 %	71.25 %	71.25 %
wine2c2	160	73.13 %	85.63 %	36.25 %	96.88 %	68.75 %	68.75 %
wine2c3	160	71.88 %	87.50 %	35.63 %	91.25 %	69.38 %	69.38 %
B							
glass	192	73.44 %	93.23 %	38.02 %	100.00 %	100.00 %	100.00 %
glass2c3	192	70.31 %	93.75 %	36.98 %	100.00 %	83.85 %	77.60 %
glass2c5	192	73.44 %	94.79 %	36.98 %	98.96 %	80.21 %	68.75 %
ionosphere	315	73.02 %	97.78 %	30.79 %	93.33 %	94.92 %	90.16 %
segment2c3	2079	69.17 %	53.34 %	23.23 %	85.91 %	69.65 %	61.04 %
segment2c4	2079	69.60 %	47.28 %	25.49 %	86.48 %	67.15 %	56.13 %
segment2c5	2079	68.98 %	42.33 %	25.16 %	81.34 %	70.37 %	63.25 %
tae	1699	69.75 %	71.16 %	25.13 %	59.80 %	76.99 %	30.43 %
thy2c3	193	73.58 %	97.93 %	34.20 %	99.48 %	79.27 %	77.72 %
vehicle	761	68.86 %	57.29 %	27.20 %	89.75 %	93.56 %	29.70 %
vehicle2c1	761	69.51 %	44.94 %	28.25 %	96.06 %	90.54 %	84.63 %
vehicle2c4	761	72.14 %	41.92 %	27.86 %	85.41 %	70.57 %	70.57 %
wav2c1	4500	71.62 %	36.91 %	26.33 %	88.56 %	75.71 %	75.71 %
wav2c2	4500	71.07 %	35.56 %	24.87 %	85.24 %	77.64 %	77.64 %
wav2c3	4500	72.87 %	36.11 %	27.62 %	88.58 %	71.82 %	71.82 %
waveform	4499	70.68 %	35.39 %	24.92 %	58.19 %	54.63 %	54.63 %
wbcd	629	70.91 %	47.69 %	24.80 %	91.73 %	72.66 %	71.07 %
wisconsin	629	68.20 %	48.97 %	27.66 %	94.91 %	73.93 %	65.98 %
C							
bal	562	64.41 %	47.86 %	23.13 %	91.46 %	54.09 %	52.14 %
bal2c1	562	67.26 %	46.26 %	24.38 %	91.10 %	52.31 %	52.31 %
bal2c2	562	66.55 %	42.35 %	25.09 %	90.75 %	53.91 %	51.07 %
bal2c3	562	66.90 %	41.28 %	23.84 %	93.77 %	53.38 %	52.49 %
biopsia	924	74.03 %	40.15 %	27.81 %	82.14 %	81.82 %	30.30 %
bpa	310	75.16 %	100.00 %	37.10 %	96.13 %	77.42 %	77.42 %
glass2c6	192	75.52 %	96.35 %	35.94 %	99.48 %	84.90 %	73.96 %
heart-statlog	243	68.72 %	41.56 %	32.92 %	91.77 %	68.31 %	68.31 %
liver-disorders	310	57.74 %	37.10 %	28.39 %	99.35 %	79.03 %	73.87 %
monks-1	500	100.00 %	34.60 %	27.00 %	46.40 %	32.40 %	29.20 %
monks-2	540	100.00 %	35.00 %	26.11 %	44.07 %	28.89 %	26.85 %
monks-3	498	100.00 %	33.33 %	26.71 %	48.19 %	28.11 %	27.51 %
pim	691	71.78 %	45.44 %	27.06 %	94.65 %	73.66 %	57.74 %
sonar	187	54.01 %	38.50 %	32.62 %	100.00 %	100.00 %	100.00 %
transfusion	673	53.79 %	34.47 %	24.37 %	94.50 %	82.17 %	82.17 %
vehicle2c2	761	68.33 %	51.77 %	28.52 %	88.30 %	76.08 %	73.06 %
vehicle2c3	761	70.30 %	51.77 %	27.73 %	97.24 %	80.29 %	71.62 %
wpc	178	63.48 %	57.87 %	35.96 %	97.75 %	70.22 %	70.22 %
operations A	100.00 %	71.95 %	79.49 %	33.58 %	93.81 %	80.38 %	77.91 %
rank A	6.68	2.95	3.95	1.00	5.78	4.05	3.60
position A	7	2	4	1	6	5	3
operations B	100.00 %	63.86 %	53.82 %	25.77 %	79.19 %	70.17 %	61.34 %
rank B	6.89	3.56	3.00	1.00	5.61	4.61	3.33
position B	7	4	2	1	6	5	3
operations C	100.00 %	64.90 %	43.78 %	25.73 %	77.35 %	58.85 %	53.51 %
rank C	6.81	4.19	2.81	1.00	5.75	4.17	3.28
position C	7	5	2	1	6	4	3
operations	100.00 %	71.68 %	63.25 %	30.39 %	89.41 %	74.79 %	68.84 %
rank	6.60	3.57	3.29	1.00	5.77	4.30	3.47
position	7	4	2	1	6	5	3

Table E.8: Summary of the average number of cases used by the CBR configurations when the five most similar clusters are selected ($C=5$). The last rows show the average of the percentage of information used, the average rank and the position in the ranking of each learner for each data set complexity (A, B, C), and for all the data sets.

References

- Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundations issues, methodological variations, and system approaches. *AI Communications*, 7:39–59.
- Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16.
- Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering for data mining applications. *SIGMOD Record ACM Special Interest Group on Management of Data*, pages 94–105.
- Aguilar-ruiz, J. S., Santos, J. C. R., and Toro, M. (2000). Data set editing by ordered projection. In *European Conference on Artificial Intelligence*, pages 251–255.
- Alahakoon, D., Halgamuge, S. K., and Srinivasan, B. (1998). A self-growing cluster development approach to data mining. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 3, pages 2901–2906.
- Allen, J. and van der Velden, R. (2007). *The flexible professional in the knowledge society*. Research Centre for Education and the Labour Market, Maastricht University.
- Anchor, K., Zydallis, J., and Gunsch, G. (2002). Extending the computer defense immune system: Network intrusion detection with a multiobjective evolutionary programming approach. In *1st Conf. on Artificial Immune Systems*, pages 12–21.
- Asuncion, A. and Newman, D. J. (2010). UCI machine learning repository.
- Bacardit, J. (2004). *Pittsburgh Genetic-Based Machine Learning in the Data Mining era: Representations, generalization, and run-time*. PhD thesis, Enginyeria i Arquitectura La Salle, Universitat Ramon Llull.
- Bacardit, J. and Llorà, X. (2009). Large scale data mining using genetics-based machine learning. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO '09*, pages 3381–3412. ACM.
- Bandyopadhyay, S., Maulik, U., and Mukhopadhyay, A. (2007). Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1506–1511.

- Basu, M. and Ho, T. K. (2006). *Data Complexity in Pattern Recognition*. Advanced Information and Knowledge Processing. Springer-Verlag, Inc., New York.
- Bélanger, M. and Martel, J. M. (2005). An Automated Explanation Approach for a Decision Support System based on MCDA. *AAAI Fall Symposium on Explanation-aware Computing*, pages 21–34.
- Bentley, P. J. and Wakefield, J. P. (1998). Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In Chawdhry, P., Roy, R., and Pant, R., editors, *Soft Computing in Engineering Design and Manufacturing*, pages 231–240. Springer-Verlag.
- Bernadó-Mansilla, E., Llorà, X., and Garrell, J. M. (2002). XCS and GALE: A comparative study of two learning classifier systems on data mining. In *Advances in Learning Classifier Systems*, volume 2321 of *LNAI*, pages 115–132. Springer.
- Bezdek, J. (1974). *Fuzzy Mathematics in Pattern Classification*. PhD thesis, Cornell University.
- Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.
- Bichindaritz, I. (2006). Memory organization as the missing link between case-based reasoning and information retrieval in biomedicine. *Computational Intelligence*, 22(3-4):148–160.
- Bloedorn, E., Talbot, L., and DeBarr, D. (2006). Data mining applied to intrusion detection: Mitre experiences. In Maloof, M., editor, *Machine Learning and Data Mining for Computer Security*, Advanced Information and Knowledge Processing, pages 65–88. Springer London.
- Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004). Finding knees in multi-objective optimization. In *8th Conference on Parallel Problem Solving from Nature (PPSN VIII)*. *Lecture Notes in Computer Science*, pages 722–731. Springer-Verlag.
- Branke, J., Deb, K., Miettinen, K., and Slowinski, R., editors (2008). *Multiobjective Optimization, Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*. Springer.
- Brown, M. (1994). *A Memory Model for Case Retrieval by Activation Passing*. PhD thesis, University of Manchester.
- Burke, B. E. (2009). Insider risk management: A framework approach to internal security. *IDC*.
- Butz, M. V., Lanzi, P. L., Llorà, X., and Loiacono, D. (2008). An analysis of matching in learning classifier systems. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 1349–1356, New York, NY, USA. ACM.
- Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics Simulation and Computation*, 3(1):1–27.

- Cano, J. R., García, S., and Herrera, F. (2008). Subgroup discovery in large size data sets pre-processed using stratified instance selection for increasing the presence of minority classes. *Pattern Recognition Letters*, 29:2156–2164.
- Cano, J. R., Herrera, F., and Lozano, M. (2006). On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining. *Applied Soft Computing*, 2006:323–332.
- Cantu-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.
- Castellanos-Garzón, J. A., García, C. A., and Miguel-Quintales, L. A. (2009). An evolutionary hierarchical clustering method with a visual validation tool. In *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence, IWANN '09*, pages 367–374. Springer-Verlag.
- Castells, M. and Martínez, C. (2001). *La era de la información: economía, sociedad y cultura*. La era de la información. Alianza, Madrid.
- Chakrabarti, D., Kumar, R., and Tomkins, A. (2006). Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06*, pages 554–560. ACM.
- Chang, P. and Lai, C. (2005). A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. *Expert Syst. Appl.*, 29(1):183–192.
- Chis, M. (2008). Hierarchical clustering using evolutionary algorithms. *Mathematical Methods for Knowledge Discovery and Data Mining*, pages 146–156.
- Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*.
- Coello, C. A. (2001). A Short Tutorial on Evolutionary Multiobjective Optimization.
- Coello, C. A. (2005). Recent trends in evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*, pages 7–32. Springer-Verlag.
- Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1:269–308.
- Coello, C. A. C. (2003). Evolutionary Multiobjective Optimization: Current and Future Challenges. In Benitez, J., Cordon, O., Hoffmann, F., and Roy, R., editors, *Advances in Soft Computing - Engineering, Design and Manufacturing*, pages 243–256. Springer, London.

- Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York, Inc.
- Colegio Oficial de Ingenieros de Telecomunicación y Asociación de Empresas de Electrónica (2007). *Informe Pafet V: Competencias profesionales y necesidades formativas en el sector de servicios que hacen un uso intensivo de las TIC*. Tecnologías de la Información y Telecomunicaciones de España, Madrid.
- Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature spaces analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.
- Corchado, E., Corchado, J. M., and Aiken, J. (2004). Ibr retrieval method based on topology preserving mappings. *Journal of Experimental & Theoretical Artificial Intelligence*, 16(3):145–160.
- Cordón, O., Herrera, F., Hoffmann, F., and Magdalena, L. (2001). *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, volume 19 of *Advances in Fuzzy Systems—Applications and Theory*. World Scientific Publishing Co. Pte. Ltd., Singapore.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, Cambridge, MA, USA, 2nd edition.
- Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 283–290. Morgan Kaufmann Publishers.
- Corral, G. (2009). *New Challenges in Detection and Management of Security Vulnerabilities in Data Networks*. PhD thesis, Ingeniería i Arquitectura La Salle (Universitat Ramon Llull).
- Corral, G., Armengol, E., Fornells, A., , and Golobardes, E. (2009a). Explanations of unsupervised learning clustering applied to data security analysis. *Neurocomputing*, 72:2754–2762.
- Corral, G., Fornells, A., Golobardes, E., and Abella, J. (2006). Cohesion factors: improving the clustering capabilities of consensus. In *Intelligent Data Engineering and Automated Learning - IDEAL, LNCS*, volume 4224, pages 488–495. Springer.
- Corral, G., Garcia-Piquer, A., Orriols-Puig, A., Fornells, A., and Golobardes, E. (2009b). Multi-objective evolutionary clustering approach to security vulnerability assessments. In *Hybrid Artificial Intelligence Systems*, volume 5572, pages 597–603.
- Corral, G., Garcia-Piquer, A., Orriols-Puig, A., Fornells, A., and Golobardes, E. (2011). Analysis of Vulnerability Assessment Results based on CAOS. *Applied Soft Computing Journal*, 11:4321–4331.
- Cowgill, M. C., Harvey, R. J., and Watson, L. T. (1999). A genetic algorithm approach to cluster analysis. *Computers & Mathematics with Applications*, 37(7):99–108.

- Crawley, E. F. (2001). *The CDIO Syllabus: A Statement of Goals for Undergraduate Engineering Education*. Massachusetts Institute of Technology.
- Czarnowski, I. (2011). Cluster-based instance selection for machine classification. *Knowledge and Information Systems*, 10.1007/s10115-010-0375-z.
- Dale, M. B. and Dale, P. T. (1992). Classification with multiple dissimilarity matrices. *Coenos*, 9:1–13.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or, the Preservation of Favoured Races in the Struggle for Life*. John Murray, London.
- Davies, D. and Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):224–227.
- Dawkins, J. and Hale, J. (2004). A systematic approach to multi-stage network attack analysis. *Second IEEE International Information Assurance Workshop*.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Declaration, B. (1999). The european higher education area. *Joint Declaration of the European Ministers of Education*.
- DeLooze, L. (2004). Classification of computer attacks using a self-organizing map. In *Proc. of the 2004 IEEE Workshop on Information Assurance*, pages 365–369.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Depren, M., Topallar, M., Anarim, E., and Ciliz, K. (2004). Network-based anomaly intrusion detection system using soms. In *Proc. of the IEEE 12th Signal Processing and Communications Applications Conference*, pages 76–79.
- Derrac, J., García, S., and Herrera, F. (2010). Stratified prototype selection based on a steady-state memetic algorithm: A study of scalability. *Memetic Computing*, 2(3):183–199.
- Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 551–556, New York, NY, USA. ACM.

- Dong, H., Hou, W., and Yin, G. (2010). An evolutionary clustering algorithm based on adaptive fuzzy weighted sum validity function. *Computational Sciences and Optimization, International Joint Conference on*, 2:357–361.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern classification*. John Wiley and Sons, Inc., New York.
- Dunn, J. C. (1974). Well separated clusters and optimal fuzzy partitions. In *Journal of Cybernetics*, volume 4, pages 95–104.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *National Academy Scientific*, 25(95):14863–14868.
- Fabre, M. G., Pulido, G. T., and Coello, C. A. C. (2010). Alternative fitness assignment methods for many-objective optimization problems. In *Proceedings of the 9th international conference on Artificial evolution*, pages 146–157, Berlin, Heidelberg. Springer-Verlag.
- Faceli, K., de Souto, M. C. P., de Araújo, D. S. A., and de Carvalho, A. C. P. L. F. (2009). Multi-objective clustering ensemble for gene expression data analysis. *Neurocomputing*, 72:2763–2774.
- Farina, M. and Amato, P. (2002). On the optimal solution definition for many-criteria optimization problems. In *Proceedings of the NAFIPS-FLINT International Conference 2002*, pages 233–238. IEEE Service Center.
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Menlo Park, CA, USA.
- Ferligoj, A. and Batagelj, V. (1992). Direct multicriterion clustering. *Journal of Classification*, 9:43–61.
- Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, New York, USA.
- Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16.
- Fornells, A., Golobardes, E., Martorell, J. M., and Garrell, J. M. (2008). Patterns out of cases using kohonen maps in breast cancer diagnosis. *International Journal of Neural Systems*, 18:33–43.
- Fornells, A., Golobardes, E., Martorell, J. M., Garrell, J. M., Bernadó, E., and Macià, N. (2007a). Measuring the applicability of self-organizing maps in a case-based reasoning system. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4478 of *LNCS*, pages 532–539. Springer-Verlag.

- Fornells, A., Golobardes, E., Martorell, J. M., Garrell, J. M., Bernadó, E., and Macià, N. (2007b). A methodology for analyzing the case retrieval from a clustered case memory. In *7th International Conference on Case-Based Reasoning*, volume 4626 of *LNAI*, pages 122–136. Springer-Verlag. Best paper nomination.
- Fowlkes, E. B. and Mallows, C. L. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383):553–569.
- Freitas, A. A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92.
- Fürnkranz, J. (1998). Integrative windowing. *Journal of Artificial Intelligence Research*, 8:129–164.
- Gan, G., Chaoqun, M., and Wu, J. (2000). *Data clustering theory, algorithms, and applications*. ASA-SIAM, Philadelphia.
- García, S. and Herrera, F. (2008). An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2009a). Assessment of competences in university degrees using data mining techniques. In *FINTDI 2009: Fomento e Innovación con Nuevas Tecnologías en la Docencia de la Ingeniería*, pages 45–49. IEEE.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2009b). Identification of subject typologies through artificial intelligence techniques to study the competences achievement of the new computer engineers. In *The 39th Annual Frontiers in Education (FIE) Conference*, pages T3D 1–2. IEEE.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., and Cugota, L. (2010a). Validación de competencias en titulaciones universitarias usando minería de datos. *Revista Iberoamericana de Tecnologías del Aprendizaje (RITA), Capítulo Español de la Sociedad de la Educación del IEEE*, 5(1):23–29.
- Garcia-Piquer, A., Fornells, A., Golobardes, E., Orriols-Puig, A., and Cugota, L. (2010b). Análisis de titulaciones universitarias basadas en competencias mediante una técnica de clustering evolutiva multiobjetivo. In *Proceedings of the III Congreso Español de Informática (CEDI 2010). Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA)*, pages 345–354.
- Garcia-Piquer, A., Fornells, A., Orriols-Puig, A., Corral, G., and Golobardes, E. (2011). Data Classification through an Evolutionary Approach Based on Multiple Criteria. *Knowledge and Information Systems*, 10.1007/s10115-011-0462-9.

- Gathercole, C. and Ross, P. (1994). Dynamic training subset selection for supervised learning in genetic programming. In *Parallel Problem Solving from Nature III*, pages 312–321. Springer-Verlag.
- Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison-Wesley, Inc., Boston, MA, USA.
- Golobardes, E. and Madrazo, L. (2009a). *Guía para la evaluación de competencias en el área de Ingeniería y Arquitectura*. Guías de evaluación de competencias. AQU Catalunya, Barcelona.
- Golobardes, E. and Madrazo, L. (2009b). *Guia per a l'avaluació de competències en l'àrea d'Enginyeria i Arquitectura*. Guies d'avaluació de competències. AQU Catalunya, Barcelona.
- Gonzalez, J. and Wagenaar, R. (2003). *Tuning Educational Structures in Europe. Final Report Phase I*. University of Deusto, Bilbao.
- Gonzalez, J. and Wagenaar, R. (2005). *Tuning Educational Structures in Europe. Final Report Phase II*. University of Deusto, Bilbao.
- Gupta, M., Rees, J., Chaturvedi, A., and Chi, J. (2006). Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. *Decision Support Systems*, 41(3):592–603.
- Hager, P. and Gonczi, A. (1994). General issues about assessment of competence. *Assessment and Evaluation in Higher Education*, 19.
- Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145.
- Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002a). Cluster validity methods: Part i. *SIGMOD Record*, 31(2):40–45.
- Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002b). Clustering validity checking methods: Part ii. *SIGMOD Record*, 31(3):19–27.
- Handl, J. and Knowles, J. (2004a). Evolutionary multiobjective clustering. *Lecture notes in computer science*, pages 1081–1091.
- Handl, J. and Knowles, J. (2004b). Multiobjective clustering with automatic determination of the number of clusters. In *Technical Report TR-COMPSYSBIO-2004-02*. UMIST.
- Handl, J. and Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 1(1):56–76.
- Handl, J. and Knowles, J. D. (2006). An investigation of representations and operators for evolutionary data clustering with a variable number of clusters. In Runarsson, T. P., Beyer, H.-G.,

- Burke, E., Merelo-Guervós, J. J., Whitley, L. D., and Yao, X., editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193, pages 839–849. Springer Berlin Heidelberg.
- Hartigan, J. and Wong, M. (1979). A k-means clustering algorithm. In *Applied Statistics*, pages 28:100–108.
- Hernández, J., Ramírez, M. J., and Ferri, C. (2004). *Introducción a la Minería de Datos*. Pearson, Madrid.
- Herrera, F., Carmona, C., González, P., and del Jesus, M. (2010). An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, 10.1007/s10115-010-0356-2.
- Ho, T. K. and Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(3):289–300.
- Hodges, K. and Wotring, J. (2000). Client typology based on functioning across domains using the cafas: Implications for service planning. *Journal of Behavioral Health Services and Research*, 27(3):257–270.
- Hofstede, F. T., Steenkamp, J. B., and Wedel, M. (1999). International market segmentation based on consumer-product relations. *Journal of Marketing Research*, 36(1):1–17.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70.
- Hore, P., Hall, L. O., and Goldgof, D. B. (2009). A scalable framework for cluster ensembles. *Pattern Recognition*, 42:676–688.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441.
- Hotho, A., Staab, S., and Stumme, G. (2003). Explaining Text Clustering Results Using Semantic Structures. *Lecture Notes in Computer Science*, 2838:217–228.
- Hruschka, E. R., Campello, R. J. G. B., and de Castro, L. N. (2004). Improving the efficiency of a clustering genetic algorithm. In *IBERAMIA*, pages 861–870.
- Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., and de Carvalho, A. C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 39(2):133–155.

- Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 146–151. ACM Press.
- Huang, Z. and Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, 7:446–452.
- Hubert, L. and Schultz, J. (1976). Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology*, 29(2):190–241.
- Iredi, S., Merkle, D., and Middendorf, M. (2000). Bi-criterion optimization with multi colony ant algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, pages 359–372. Springer.
- Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*, pages 2419–2426. IEEE.
- Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.
- John, G. H. and Langley, P. (1996). Static versus dynamic sampling for data mining. In *Knowledge Discovery and Data Mining*, pages 367–370.
- Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.
- Kargupta, H., Han, J., Yu, P. S., Motwani, R., and Kumar, V. (2009). *Next Generation of Data Mining*. Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press, USA.
- Kaski, S., Kangas, J., and Kohonen, T. (2003). *Bibliography of Self-Organizing Map Papers: 1998-2001*, <http://www.cis.hut.fi/research/refs/>.
- Kasprzak, E. M. and Lewis, K. E. (2001). Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. *Structural and Multidisciplinary Optimization*, 22(3):208–218.
- Kaufman, L. and Rousseeuw, P. J. (1990). Finding groups in data: An introduction to cluster analysis. *John Wiley & Sons*.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480.
- Kohonen, T. (2000). *Self-Organizing Maps*. Springer, Berlin, 3rd edition.
- Kokolo, I., Hajime, K., and Shigenobu, K. (2001). Failure of pareto-based moeas: Does non-dominated really mean near to optimal? In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 957–962. IEE Service Center.

- Konak, A., Coit, D. W., and Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007.
- Koza, J. R. (1992). *Genetic Programming. Programing of computers by means of natural selection*. MIT Press, Cambridge, MA, USA.
- Krishna, K. and Narasimha, M. (1999). Generic k-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, B29(3):433–439.
- Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In Neyman, J., editor, *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Berkeley, CA, USA.
- Kuo, R., Wang, H., Hu, T., and Chou, S. (2005). Application of ant k-means on clustering analysis. *Computers & Mathematics with Applications*, 50(10–12):1709–1724.
- Langdon, W. B. (1997). Fitness causes bloat in variable size representations. Technical report, Michigan State University, East Lansing, Michigan.
- Law, M. H., Topchy, A. P., and Jain, A. K. (2004). Multiobjective data clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 424–430.
- Le, K. and Silva, D. L. (2007). Obtaining better non-dominated sets using volume dominance. In *IEEE Congress on Evolutionary Computation*, pages 3119–3126.
- Le, T. V. (1995). Evolutionary fuzzy clustering. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 753–758.
- Legeny, C., Juhász, S., and Babos, A. (2006). Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393.
- Legány, C., Juhász, S., and Babos, A. (2006). Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393.
- Lenz, M., Burkhard, H. D., and Brückner, S. (1996). Applying case retrieval nets to diagnostic tasks in technical domains. In *Proceedings of the Third European Workshop on Advances in Case-Based Reasoning*, pages 219–233. Springer-Verlag.
- Leung, K. and Leckie, C. (2005). Unsupervised anomaly detection in network intrusion detection using clusters. In *Proc. 28th Australasian CS Conf.*, volume 38.

- Liu, Y., Yoshioka, M., Homma, K., and Shibuya, T. (2009). Efficiently finding the 'best' solution with multi-objectives from multiple topologies in topology library of analog circuit. In *Proceedings of the 2009 Asia and South Pacific Design Automation Conference, ASP-DAC '09*, pages 498–503. IEEE Press.
- Llorà, X., Sastry, K., Yu, T. L., and Goldberg, D. E. (2007). Do not match, inherit: fitness surrogates for genetics-based machine learning techniques. In *GECCO*, pages 1798–1805.
- Llorens, A. (2008). *Study of the professional profile of the engineers in the IT context*. Catalan Association of Telecommunication Engineers (ACET), Barcelona.
- Ma, P. C. H., Chan, K. C. C., Yao, X., and Chiu, D. K. Y. (2006). An evolutionary clustering algorithm for gene expression microarray data analysis. *IEEE Transactions on Evolutionary Computation*, 10(3):296–314.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. CA: University of California Press.
- Malek, M. and Amy, B. (2007). A pre-processing model for integrating cbr and prototype-based neural networks. In *Connectionism-symbolic Integration*. Erlbaum.
- Manning, C. and Schuetze, H. (2000). Foundations of statistical natural language processing. In *International Conference on Computational Linguistics*. MIT Press.
- Matake, N., Hiroyasu, T., Miki, M., and Senda, T. (2007). Multiobjective clustering with automatic k-determination for large-scale data. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 861–868. ACM.
- Maulik, U. and Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9):1455–1465.
- Mendel, G. (1865). Experiments in Plant Hybridization. In *Natural History Society of Brunn in Bohemia*.
- Messac, A., Sundaraj, G. J., Tappeta, R. V., and Renaud, J. E. (2000). Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA Journal*, 38(6):1084–1091.
- Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2007). Multiobjective genetic fuzzy clustering of categorical attributes. *International Conference on Information Technology*, 0:74–79.
- Nedjah, N., Abraham, A., and Mourelle, L. (2007). *Computational Intelligence in Information Assurance and Security*, volume 57 of *Studies in Computational Intelligence*. Springer, Berlin.
- Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. PhD thesis, Princeton University, New Jersey, USA.

- Nicholson, R., Bridge, D., and Wilson, N. (2006). Decision diagrams: Fast and flexible support for case retrieval and recommendation. In *8th European Conference on Case-Based Reasoning*, volume 4106 of *LNAI*, pages 136–150. Springer-Verlag.
- Obradovic, Z. and Vucetic, S. (2004). *Challenges in Scientific Data Mining: Heterogeneous, Biased, and Large Samples*, pages 381–401. AAAI Press, Menlo Park, CA.
- Olvera, J. A., Carrasco, J. A., and Martínez, J. F. (2010). Prototype selection methods. *Computación y Sistemas*, 13:449–462.
- Orriols-Puig, A., Bernadó-Mansilla, E., Sastry, K., and Goldberg, D. E. (2007). Substructural surrogates for learning decomposable classification problems: implementation and first results. In *GECCO (Companion)*, pages 2875–2882.
- Osyczka, A. (1985). Multicriteria Optimization for Engineering Design. In Gero, J. S., editor, *Design Optimization*, pages 193–227. Academic Press.
- Pareto, V. (1896). *Course d'Économie Politique*. F. Pichou, Lausanne and Paris.
- Park, Y. and Song, M. (1998). A genetic algorithm for clustering problems. In *Proceedings of the 3rd Annual Conference on Genetic Programming*, pages 568–575. Morgan Kaufmann.
- Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference of Machine Learning*, pages 727–734. Morgan Kaufmann Publishers Inc.
- Plaza, E. and Arcos, J.-L. (1990). A reflective architecture for integrated memory-based learning and reasoning. (*Richter, Wess, Althoff, Maurer, eds.*) *Proceedings First European Workshop on Case-Based Reasoning. Vol 2*, pages 329–334.
- Porter, B. (1986). PROTOS: An experiment in knowledge acquisition for heuristic classification tasks. *Proceedings First International Meeting on Advances in Learning, Les Arcs, France*, pages 159–174.
- Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401.
- Pudil, P., Ferri, F. J., Novovicova, J., and Kittler, J. (1994). Floating search methods for feature selection with nonmonotonic criterion functions. In *Pattern Recognition*, volume 2, pages 279–283.
- Ramadas, M., Ostermann, S., and Tjaden, B. C. (2003). Detecting anomalous network traffic with self-organizing maps. In *RAID'03: Proc. 6th Symposium on Recent Advances in Intrusion Detection*, volume 2820, pages 36–54.

- Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850.
- Rechenberg, I. (1973). *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart.
- Ripon, K. S. N., Tsang, C., Kwong, S., and Ip, M. (2006). Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm. In *18th International Conference on Pattern Recognition*, volume 1, pages 1200–1203.
- Rissland, E. L., Skalak, D. B., and Friedman, M. (1993). Case retrieval through multiple indexing and heuristic search. In *International Joint Conferences on Artificial Intelligence*, pages 902–908.
- Rosenberg, R. (1967). Simulation of genetic populations with biochemical properties. *Dissertation Abstracts International*, 28(7):67–17.
- Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. In *Journal of Computational and Applied Mathematics*, volume 20, pages 53–65.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational Applications in Math*, 20:53–65.
- Saha, S. and Bandyopadhyay, S. (2010). A new multiobjective clustering technique based on the concepts of stability and symmetry. *Knowledge and Information Systems*, 23:1–27.
- Salamó, M. and Golobardes, E. (2002). Deleting and building sort out reduction techniques for case base maintenance. *Proceedings of the European Conference on Case-Based Reasoning*, 1:365–379.
- Schaaf, J. W. (1995). Fish and Sink - an anytime-algorithm to retrieve adequate cases. In *Proceedings of the First International Conference on Case-Based Reasoning Research and Development*, volume 1010, pages 538–547. Springer-Verlag.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- Schölkopf, B., Smola, A., and Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319.
- Schwefel, H. P. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume 26 of *ISR*. Birkhaeuser, Stuttgart.
- Services, I. G. T. (2009). Ibm internet security systems x-force 2009 mid-year trend and risk report. *IBM Corporation*.

- Sheskin, D. J. (2007). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC, London, 4 edition.
- Shreider, Y. A. (1964). *Method of Statistical Testing: Monte Carlo Method*. Elsevier North-Holland.
- Sibson, R. (1973). An optimally efficient algorithm for the single link *cluster* method. *The Computer Journal*, 16(1):30–34.
- Sobo, I. M. (1984). *The Monte Carlo Method*. Mir Publishers, Moscow.
- Strehl, A. and Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.
- Strobbe, M., Laere, O. V., Dhoedt, B., Turck, F. D., and Demeester, P. (2011). Hybrid reasoning technique for improving context-aware applications. *Knowledge Information Systems*, 10.1007/s10115-011-0411-7.
- Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition*. Academic Press, Burlington, USA.
- Tibshirani, R., Walther, G., and Hastie, T. (2000). Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society*, 63:411–423.
- Topchy, A., Jain, A., and Punch, W. (2004). A mixture model for clustering ensembles. In *Proceedings of the SIAM International Conference on Data Mining*, pages 379–390.
- Tseng, L. Y. and Bien, S. (2000). A genetic clustering algorithm for data with non-spherical-shape clusters. *Pattern Recognition*, 33(7):1251–1259.
- Tzortzis, G. and Likas, A. (2008). The global kernel k-means clustering algorithm. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1977–1984.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- Veldhuizen, D. A. V. and Lamont, G. B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation Journal*, 8:125–147.
- Vernet, D. and Golobardes, E. (2003). An unsupervised learning approach for case-based classifier systems. *Expert Update. The Specialist Group on Artificial Intelligence*, 6(2):37–42.
- Vernet, D., Nicolas, R., Golobardes, E., Fornells, A., and Garcia-Piquer, A. (2010). Intelligent tutoring system framework for the acquisition of knowledge and competences. In *The 40th Annual Frontiers in Education (FIE) Conference*, pages T4G 1–2. IEEE.

- Wess, S., Althoff, K. D., and Derwand, G. (1994). Using k-d trees to improve the retrieval step in case-based reasoning. In *Selected papers from the First European Workshop on Topics in Case-Based Reasoning*, volume 837, pages 167–181. Springer-Verlag.
- Witten, I. H. and Frank, M. A. H. (2011). *DataMining: Practical machine learning tools and techniques with Java implementations*. 3rd Edition, Morgan Kaufmann Publishers, Burlington, USA.
- Yang, E., Erdogan, A., Arslan, T., and Barton, N. (2007). Multi-objective evolutionary optimizations of a space-based reconfigurable sensor network under hard constraints. In *Symp. on Bioinspired, Learning, and Int. Syst. for Security*, pages 72–75.
- Yang, Q. and Wu, J. (2001). Enhancing the effectiveness of interactive cas-based reasoning with clustering and decision forests. *Applied Intelligence*, 14(1):49–64.
- Yeung, K., Medvedovic, M., and Bumgarner, R. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):1–17.
- Yeung, K. and Ruzzo, W. (2001). Details of the adjusted rand index and clustering algorithms. supplement to the paper “an empirical study on principal component analysis for clustering gene expression data”. *Science*, 17(9):763–774.
- Zenko, B., Dzeroski, S., and Struyf, J. (2005). Learning predictive clustering rules. In *Knowledge Discovery in Inductive Databases*, volume 3933 of *Lecture Notes in Computer Science*, pages 234–250. Springer-Verlag.
- Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.
- Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland.
- Zou, X., Chen, Y., Liu, M., and Kang, L. (2008). A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 38(5):1402–1412.



Aquesta Tesi Doctoral ha estat defensada el dia ____ d _____ de ____
al Centre _____

de la Universitat Ramon Llull

davant el Tribunal format pels Doctors sotasignants, havent obtingut la qualificació:

President/a

Vocal

Vocal

Vocal

Secretari/ària

Doctorand/a
