



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Multi-manned Assembly Line Balancing Problem con tiempos de proceso dependientes (MALBP-DTT)

Enric Andreu Casas

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (<http://upcommons.upc.edu/tesis>) i el repositori cooperatiu TDX (<http://www.tdx.cat/>) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (<http://upcommons.upc.edu/tesis>) y el repositorio cooperativo TDR (<http://www.tdx.cat/?locale-attribute=es>) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing*). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (<http://upcommons.upc.edu/tesis>) and the cooperative repository TDX (<http://www.tdx.cat/?locale-attribute=en>) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (*framing*). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Doctorado Cadena de Suministro y Dirección de Operaciones

Multi-manned Assembly Line Balancing Problem con tiempos de proceso dependientes (MALBP-DTT)

TESIS DOCTORAL

Doctorando:	Andreu Casas, Enric
Directores de la tesis:	García Villoria, Alberto Pastor Moreno, Rafael
Fecha:	Septiembre 2023

RESUMEN

La tesis doctoral propuesta consiste en resolver el problema de equilibrado de líneas de montaje con múltiples operarios en una estación, también conocido como *Multi-manned Assembly Line Balancing Problem* (MALBP). Este tipo de líneas de montaje son habituales en la fabricación de productos de gran tamaño, como automóviles, en los que diferentes operarios realizan distintas tareas sobre la misma unidad de trabajo de manera simultánea.

En la mayoría de publicaciones relacionadas con el MALBP se asume que los operarios de una misma estación no se interfieren entre ellos y que los tiempos de proceso de las tareas son deterministas e invariables. Hay un número reducido de trabajos que proponen una variante, en la que los tiempos de proceso pueden verse incrementados en función del número de operarios que haya en la estación, lo cual permite representar situaciones más realistas.

El objetivo de la tesis doctoral es diseñar procedimientos de resolución, exactos y heurísticos, para la variante del MALBP con tiempos de proceso dependientes del número de operarios que hay en la estación (*Multi-manned Assembly Line Balancing Problem with Dependent Task Times*; MALBP-DTT), que puedan ser de interés para la comunidad científica y que mejoren a los procedimientos que ya han sido publicados en la literatura hasta la fecha.

En primer lugar, se desarrollan diferentes procedimientos con el objetivo de minimizar el número de trabajadores y estaciones a asignar en la línea, dado un tiempo de ciclo (tiempo disponible en cada estación para completar las tareas asignadas). Este problema principalmente se encuentra en situaciones en las que se desea diseñar una nueva línea de montaje en función de una demanda externa conocida. Se proponen resoluciones basadas en un modelo matemático, dos procedimientos que utilizan la matheurística de Relajar y Fijar, una heurística basada en la resolución de un problema de partición con restricciones (denominada "HEUR_PART"), y un conjunto de otras variantes del procedimiento HEUR_PART. Los experimentos computacionales realizados indican que HEUR_PART y la variante HEUR_PART_SGL son las propuestas que mejor funcionan.

A continuación, se aplica el procedimiento Empirically Adjusted Greedy Heuristics (EAGH) y se presenta una nueva variante (denominada "EAGH-CKTL"), que se basa en el uso de EAGH combinado con el concepto de cóctel de heurísticas. EAGH y EAGH-CKTL se utilizan para diseñar nuevas reglas de prioridad para resolver el MALBP-DTT usando HEUR_PART. En particular, EAGH-CKTL se aplica para construir nuevas reglas de prioridad que tengan un buen rendimiento como parte de un cóctel de heurísticas. Los experimentos computacionales realizados muestran la eficacia de utilizar tanto EAGH,

como EAGH-CKTL, en el proceso de diseño de reglas de prioridad eficientes: una de las reglas de prioridad diseñadas con EAGH presenta un mejor rendimiento que cualquier otra regla propuesta en HEUR_PART; mientras que otra regla, diseñada con EAGH-CKTL, evidencia una notable mejora en los resultados de HEUR_PART cuando se añade en su cóctel de heurísticas.

Asimismo, se desarrollan nuevas heurísticas que también resuelven problemas de partición con restricciones. Se diseñan dos procedimientos: MULTI_HEUR_PART, basado en la resolución de problemas de partición combinados; y OPWK_PART, basado en generar estaciones vecinas mediante intercambio y transferencia de tareas. OPWK_PART es el procedimiento que obtiene los mejores resultados comparado tanto con los otros procedimientos propuestos para este tipo de problema como con cualquiera de los procedimientos que se han publicado en la literatura para esta variante hasta el momento.

Un segundo problema de líneas de montaje a resolver consiste en producir el mayor número de unidades de producto utilizando un número fijo de estaciones. En la segunda parte de la tesis doctoral se estudia, por primera vez en la literatura, esta clase de problema: el MALBP-DTT con el objetivo de minimizar el tiempo de ciclo. Para ello, se presenta un modelo matemático y se desarrollan dos heurísticas basadas en la resolución de un problema de partición con restricciones: HEUR_CT_PART y OPCT_PART. En particular, OPCT_PART utiliza un procedimiento heurístico constructivo *greedy*, denominado GRDY_DTT2, y combina la resolución de un problema de partición con la configuración de estaciones vecinas mediante intercambio y transferencia de tareas. Los experimentos computacionales realizados indican que OPCT_PART es la propuesta que mejor funciona. Además, se muestra que OPCT_PART obtiene mejores resultados que cualquiera de los procedimientos publicados en la literatura que resuelven el MALBP con los mismos objetivos.

ABSTRACT

The proposed PhD thesis consists in solving the Multi-manned Assembly Line Balancing Problem (MALBP). This type of assembly lines is common in the manufacture of large products, such as automobiles, in which different workers perform different tasks on the same work unit simultaneously.

Most of the literature related to MALBP assumes that workers at the same station do not interfere with each other and that task processing times are deterministic and invariant. Nevertheless, few number of papers propose a variant in which processing times can be increased depending on the number of workers at the station, which allows more realistic situations to be represented.

The aim of the PhD thesis is to design exact and heuristic resolution procedures for the Multi-manned Assembly Line Balancing Problem with Dependent Task Times (MALBP-DTT) variant, which may be of interest to the scientific community and which improve the procedures that have been published in the literature up to now.

Firstly, different procedures are developed with the aim of minimising the number of workers and stations to be assigned on the line, given a cycle time (time available at each station to complete the assigned tasks). This problem is mainly encountered in situations where it is required to design a new assembly line in terms of a known external demand. This PhD thesis proposes resolutions on the basis of a mathematical model, two Relax-and-Fix procedures, a heuristic based on solving a partition problem with constraints (named "HEUR_PART") and a set of other variants of the HEUR_PART procedure. The computational experiments indicate that HEUR_PART and the HEUR_PART_SGL variant are the proposals that perform best.

Furthermore, this PhD thesis applies Empirically Adjusted Greedy Heuristics (EAGH) and presents a new procedure (named "EAGH-CKTL"), which is based on using EAGH combined with the cocktail of heuristics concept. EAGH and EAGH-CKTL are used to design new priority rules for solving MALBP-DTT through the HEUR_PART steps. In particular, EAGH-CKTL is applied for building new priority rules that have a good performance as part of a cocktail of heuristics. The computational experiments show the efficiency of using both EAGH and EAGH-CKTL in the process of designing efficient priority rules: one of the priority rules designed with EAGH presents a better performance than any other rule proposed in HEUR_PART; while another rule, designed with EAGH-CKTL, evidences a remarkable improvement in the HEUR_PART results when added in its cocktail of heuristics.

Moreover, new heuristics that also solve a partition problem with constraints are developed.

Two procedures are designed: MULTI_HEUR_PART, based on solving a combined partition problem; and OPWK_PART, based on generating neighbouring stations by means of task swaps and transfers. OPWK_PART is the procedure which obtains the best results, compared both with the other proposed procedures for solving the same problem and with any other procedure that has been published in the literature for this variant up to now.

A second problem to be solved for multi-manned assembly lines is to produce the highest number of workpieces using a fixed number of workstations. In the second part of the thesis this kind of problem is addressed for the first time in the literature: the MALBP-DTT with the objective of minimising the cycle time. For this purpose, a mathematical model is presented. Furthermore, two heuristics based on solving a partition problem with constraints are developed: HEUR_CT_PART and OPCT_PART. In particular, OPCT_PART uses a greedy constructive heuristic procedure, named GRDY_DTT2, and combines the resolution of a partition problem with the generation of neighbouring workstations by task swaps and transfers. The computational experiments indicate that OPCT_PART is the proposal that performs best. Additionally, it is shown that OPCT_PART obtains better results than the ones published in the literature for the MALBP with the same objectives.

GLOSARIO

- ACO* *Ant Colony Optimization*, optimización basada en colonias de hormigas.
- BDA* *Benders' decomposition algorithm*, algoritmo de descomposición de Benders.
- B&B* *Branch and Bound*, ramificación y acotación.
- CBC* *Combinatorial Benders Cut method*, método de corte combinatorio de Benders.
- EFO* *Electromagnetic Field Optimization*, optimización del campo electromagnético.
- FBS* *Filtered Beam Search algorithm*, algoritmo filtrado de búsqueda en haz.
- GA* *Genetic Algorithm*, algoritmo genético.
- MAL* *Multi-manned Assembly Line*, línea de montaje con múltiples operarios.
- MBO* *Migrating Birds Optimisation algorithm*, algoritmo de optimización basado en aves migratorias.
- PSO* *Particle Swarm Optimization*, optimización por nube de partículas.
- SA* *Simulated Annealing*, recocido simulado.
- TS* *Tabu Search*, algoritmo de búsqueda tabú.

SUMARIO

1. INTRODUCCIÓN	10
2. OBJETIVOS Y ALCANCE	12
2.1. Objetivos	12
2.2. Alcance	13
3. ESTADO DEL ARTE DEL PROBLEMA DE EQUILIBRADO DE LÍNEAS DE MONTAJE	14
3.1. Las líneas de montaje	14
3.1.1. Introducción	14
3.1.2. Conceptos principales	15
3.1.3. Clasificación de las líneas de montaje	16
3.2. Tipos de problemas de líneas de montaje	20
3.2.1. Problemas de diseño de líneas de montaje (ALDP)	20
3.2.2. Problemas de equilibrado de líneas de montaje (ALBP)	21
4. PROBLEMA DE EQUILIBRADO DE LÍNEAS CON MÁS DE UN OPERARIO POR ESTACIÓN (MALBP)	25
4.1. Introducción	25
4.2. Estado del arte	27
4.3. Definición del problema a resolver: MALBP con tiempos de proceso dependientes (MALBP-DTT)	35
5. PROCEDIMIENTOS DE RESOLUCIÓN PARA EL SMALBP-DTT1	38
5.1. Modelo matemático (MILP-DTT1)	39
5.1.1. Datos	39
5.1.2. Variables	40
5.1.3. Función objetivo	40
5.1.4. Restricciones	41
5.1.5. Cotas	43
5.2. Procedimientos de Relajar y Fijar	44
5.2.1. Relajar y Fijar, primera propuesta (1RF)	46
5.2.2. Relajar y Fijar, segunda propuesta (2RF)	48
5.3. Heurística basada en la resolución de un problema de partición con restricciones (HEUR_PART)	49
5.3.1. Generación de soluciones iniciales del SALBP-1	50
5.3.2. Generación de uniones de estaciones (U-estaciones)	52
5.3.2.1. Factibilidad de la unión de estaciones según precedencias	52
5.3.2.2. Factibilidad de la unión de estaciones sin incrementar trabajadores	53

5.3.3.	Resolución de un problema de partición con restricciones	55
5.4.	Experiencia computacional para comparar MILP-DTT1, 1RF, 2RF y HEUR_PART	57
5.4.1.	Características de la experiencia computacional	58
5.4.2.	Análisis de los resultados.....	60
5.4.2.1.	Comparación de los procedimientos MILP-DTT1, 1RF y 2RF, y HEUR_PART	60
5.4.2.2.	Comparación de HEUR_PART según las distintas reglas de prioridad que incorpora.....	62
5.5.	Variantes de HEUR_PART	63
5.5.1.	Descripción de las variantes de HEUR_PART	63
5.5.2.	Experiencia computacional para comparar HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE	64
5.6.	Empirically Adjusted Greedy Heuristics (EAGH)	65
5.6.1.	Diseño de reglas de prioridad aplicando EAGH a HEUR_PART_1	66
5.6.2.	Diseño de reglas de prioridad aplicando EAGH con un cóctel de heurísticas (EAGH-CKTL)	71
5.6.2.1.	Procedimiento EAGH-CKTL.....	71
5.6.2.2.	Diseño de reglas de prioridad aplicando EAGH-CKTL a HEUR_PART	72
5.6.3.	Experiencia computacional para comparar HEUR_PART, R13-R15 de EAGH y R16-R18 de EAGH-CKTL	75
5.7.	Heurística basada en la resolución de problemas de partición combinados (MULTI_HEUR_PART)	77
5.8.	Heurística OPWK_PART	82
5.8.1.	Modificación de estaciones mediante intercambio y transferencia de tareas (Generación de M-estaciones).....	83
5.8.1.1.	Factibilidad de pares de estaciones modificadas según precedencias.....	84
5.8.1.2.	Factibilidad de pares de estaciones modificadas sin aumentar el número de trabajadores.....	85
5.8.2.	Resolución de un problema de partición con restricciones	86
5.9.	Experiencia computacional para comparar HEUR_PART con R16, MULTI_HEUR_PART y OPWK_PART	88
5.9.1.	Características de la experiencia computacional	88
5.9.2.	Análisis de los resultados.....	90
5.9.2.1.	Comparación entre HEUR_PART con R16, MULTI_HEUR_PART y OPWK_PART.....	90
5.9.2.2.	Comparación entre OPWK_PART y Max_S.....	91
5.9.2.3.	Comparación entre OPWK_PART y el número óptimo de trabajadores y estaciones del SALBP-1	93

5.9.2.4. Comparación entre OPWK_PART y GanttSA	94
5.10. Conclusiones sobre los procedimientos de resolución del SMALBP-DDT1.....	96
6. PROCEDIMIENTOS DE RESOLUCIÓN PARA EL SMALBP-DDT2 _____	98
6.1. Modelo matemático (MILP-DDT2)	98
6.1.1. Datos.....	99
6.1.2. Variables.....	99
6.1.3. Función objetivo	100
6.1.4. Restricciones	100
6.2. Heurística HEUR_CT_PART	102
6.2.1. Generación de soluciones iniciales del SALBP-2.....	104
6.2.2. Generación de uniones de estaciones (U-estaciones).....	105
6.2.2.1. Factibilidad de la unión de estaciones según precedencias.....	105
6.2.2.2. Factibilidad de la unión de estaciones sin superar la concentración máxima de trabajadores.....	106
6.2.3. Resolución de un problema de partición con restricciones	107
6.3. Heurística OPCT_PART	109
6.3.1. Generación de soluciones iniciales del SMALBP-DDT2 (GRDY_DDT2).....	111
6.3.2. Modificación de estaciones mediante intercambio y transferencia de tareas (Generación de M-estaciones).....	113
6.3.2.1. Factibilidad de pares de estaciones modificadas según precedencias.....	114
6.3.2.2. Factibilidad de pares de estaciones modificadas sin exceder el tiempo de ciclo ni la concentración máxima de trabajadores	115
6.3.3. Resolución de un problema de partición con restricciones	116
6.4. Experiencia computacional para comparar MILP-DDT2, HEUR_CT_PART, GRDY_DDT2 y OPCT_PART	118
6.4.1. Características de la experiencia computacional	118
6.4.2. Análisis de los resultados.....	120
6.4.2.1. Comparación de los procedimientos propuestos.....	120
6.4.2.2. Comparación entre OPCT_PART, DSA e ISA	123
6.5. Conclusiones procedimientos de resolución SMALBP-DDT2.....	125
7. CONCLUSIONES FINALES, FUTURAS VÍAS DE INVESTIGACIÓN Y CONTRIBUCIONES _____	126
7.1. Conclusiones.....	126
7.2. Futuras vías de investigación	128
7.3. Contribuciones	130
8. REFERENCIAS _____	131
APÉNDICE _____	137
Apéndice 1 – Ejemplares de Otto <i>et al.</i> (2013) de calibración y entrenamiento.....	137

Apéndice 2 – Resultados de la experiencia computacional de MILP-DTT1, 1RF, 2RF, HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE	138
Apéndice 3 – Resultados de la experiencia computacional de EAGH y EAGH-CKTL143	
Apéndice 4 – Resultados de la experiencia computacional de MULTI_HEUR_PART y OPWK_PART	148
Apéndice 5 – Comparación del procedimiento GRDY_DTT1 con OPWK_PART y Max_S	153
Apéndice 5.1 – Comparación de GRDY_DTT1 con OPWK_PART	154
Apéndice 5.2 – Comparación de GRDY_DTT1 con Max_S	155
Apéndice 6 – Resultados de la experiencia computacional de MILP-DTT2, HEUR_CT_PART, GRDY_DTT2 y OPCT_PART	156

1. Introducción

Las líneas de montaje son comúnmente utilizadas para la producción en masa de bienes estandarizados. El proceso de producción se divide en fases, las cuales deben ser completadas por todas las unidades de un producto. Cada fase se realiza en una estación de trabajo donde, por un periodo de tiempo (conocido como tiempo de ciclo), uno o más trabajadores realizan las tareas que han sido asignadas a la estación.

Las líneas de montaje pueden ser clasificadas según diferentes criterios, en función de las características de la línea a las que se haga referencia. Según Ghosh y Gagnon (1989) y Becker y Scholl (2006), es posible categorizar las líneas de montaje en función de: el tipo de producto que procesa (simples, mixtas, multi-modelos), la duración de las tareas (deterministas, estocásticas, dependientes), la arquitectura (serial, con estaciones en paralelo, líneas paralelas, de dos lados, con múltiples operarios, circulares, en forma de U), el tipo de flujo de unidades de producto (síncrona, asíncrona, de alimentación), el tipo de operador (manual, robotizada) o la disciplina de entrada de las unidades de producto en la línea (fija o variable).

En la literatura, los problemas de líneas de montaje son habitualmente clasificados en dos grupos: problemas de diseño de líneas de montaje (*Assembly Line Design Problem*, ALDP) y problemas de equilibrado de líneas de montaje (*Assembly Line Balancing Problem*, ALBP). Por un lado, los problemas de diseño de líneas de montaje consisten en el diseño y la configuración de las líneas y estaciones. La resolución de los ALDP resulta difícil de automatizar debido a que se deben tomar un elevado número de decisiones interrelacionadas, cuya repercusión se extiende a múltiples objetivos. Por otro lado, los problemas de equilibrado de líneas de montaje consisten en distribuir un conjunto de tareas, necesarias para ensamblar el producto, entre el conjunto de estaciones que componen la línea de montaje.

Baybars (1986) distingue entre dos tipos de ALBP: el problema simple (*Simple Assembly Line Balancing Problem*, SALBP) y el problema general (*General Assembly Line Balancing Problem*, GALBP). Las líneas de montaje del problema SALBP se caracterizan por ser simples (procesan un único tipo de producto), deterministas (los tiempos de las tareas son conocidos e invariables), seriales (estaciones simples distribuidas en serie), síncronas (las estaciones tienen el mismo tiempo de ciclo) y de entrada fija (las unidades de producto llegan en intervalos de tiempos iguales). Todos los otros tipos de problemas de equilibrados de líneas no incluidos en el SALBP constituyen los GALBP.

Debido al hecho que las hipótesis del SALBP son generalmente restrictivas, los investigadores han intensificado sus esfuerzos examinando características adicionales de

los sistemas industriales, con el objetivo de abordar problemas más cercanos a la realidad (los GALBP). Entre la variedad de problemas GALBP se encuentra el problema de equilibrado de líneas de montaje con múltiples operarios en una estación (*Multi-manned Assembly Line Balancing Problem*, MALBP). El MALBP se diferencia de otros GALBP por contemplar la posibilidad de incorporar varios operarios en una misma estación. Los trabajadores, siempre y cuando no se interfieran y se respeten las relaciones de precedencias, pueden llevar a cabo distintas tareas de forma simultánea. Las líneas con múltiples operarios suelen encontrarse en la fabricación de productos de grandes dimensiones, como los automóviles.

En la mayoría de las publicaciones relacionadas con el MALBP se asume que los operarios de una misma estación no interfieren entre ellos y que los tiempos de las tareas son deterministas e invariables. La presente tesis doctoral estudia una variante del MALBP en la que los operarios pueden interferir entre ellos y, por consiguiente, el tiempo que necesitan para realizar una tarea puede verse incrementado. Considerar tiempos de proceso dependientes del número de operarios en la estación permite representar situaciones más cercanas a la realidad. Esta variante se denomina MALBP con tiempos de proceso dependientes (*Multi-manned Assembly Line Balancing Problem with Dependent Task Times*, MALBP-DTT).

2. Objetivos y alcance

2.1. Objetivos

Esta tesis doctoral estudia el problema de equilibrado de líneas de montaje con múltiples operarios en la estación (*Multi-manned Assembly Line Balancing Problem*, MALBP).

El objetivo es desarrollar procedimientos de resolución, que puedan ser de interés para la comunidad científica y que mejoren los que han sido publicados en la literatura hasta la fecha; todo esto para la variante del problema MALBP en la que los tiempos de proceso de las tareas dependen del número de operarios presentes en la estación donde estas son procesadas (MALBP-DTT).

Los objetivos específicos de esta tesis doctoral son:

1. Realizar un completo estado del arte sobre el MALBP:
 - Para cada una de las publicaciones relacionadas con el MALBP, analizar los objetivos a optimizar, los procedimientos de resolución propuestos y otros aspectos que puedan resultar de interés para la presente tesis.
 - Para las publicaciones del estado del arte que resuelvan el problema MALBP considerando que los operarios se pueden interferir durante la realización de las tareas, analizar cómo se definen dichas interferencias.
 - En particular, estudiar con detalle aquellos artículos que traten las interferencias entre operarios mediante la posibilidad de incrementar el tiempo de realización de las tareas según el número de trabajadores que haya en la estación.
2. Diseñar un conjunto de procedimientos que resuelvan el MALBP-DTT. Proponer métodos exactos, como por ejemplo la programación matemática, y, en el caso de que éstos no resulten ser eficientes, métodos heurísticos, metaheurísticos y/o matheurísticos. Se diseñarán procedimientos para el MALBP-DTT teniendo en consideración dos objetivos distintos:
 - Dado un tiempo de ciclo (tiempo disponible en cada estación para completar las tareas asignadas), minimizar el número de estaciones de la línea y el de trabajadores a asignarle.
 - Dado un número de estaciones, minimizar el tiempo de ciclo y el número

de trabajadores a asignar en la línea.

3. Definir una base de ejemplares a utilizar para evaluar, mediante un experimento computacional, cada uno de los procedimientos de resolución propuestos. Asimismo, comparar los resultados de los procedimientos diseñados en esta tesis doctoral con los resultados obtenidos mediante procedimientos propuestos en la literatura.
4. Obtener resultados del problema de equilibrado MALBP-DTT que puedan ser de interés para la comunidad científica y que mejoren los que han sido publicados en la literatura hasta el momento.

2.2. Alcance

La presente tesis doctoral se centra en el diseño de procedimientos exactos y heurísticos que resuelvan el problema de equilibrado MALBP con tiempos de proceso de las tareas dependientes del número de operarios que se encuentren en cada estación (MALBP-DTT).

Más concretamente, este trabajo consiste en el planteamiento y la resolución del problema, analizando los resultados y exponiendo las conclusiones, así como las posibles vías de investigación futuras.

El estudio del problema se realiza desde una perspectiva teórica. De este modo, se excluye del alcance de esta tesis doctoral la adaptación e implementación de los procedimientos de resolución diseñados a casos prácticos del sector industrial.

3. Estado del arte del problema de equilibrado de líneas de montaje

3.1. Las líneas de montaje

3.1.1. Introducción

Una línea de montaje es un proceso de producción que divide la fabricación de un bien en fases, que son completadas mediante una secuencia predefinida. Una línea de montaje está formada por una secuencia de estaciones de trabajo a través de las cuales fluyen las unidades de un producto. Se trata de uno de los procesos de fabricación más utilizados para la producción en masa de productos.

Según Corday (2014), durante el siglo XII, el Arsenal de Venecia empleó técnicas similares a la producción en línea para la construcción de buques de guerra. En su momento de mayor eficiencia, el Arsenal de Venecia llegó a producir alrededor de una embarcación al día.

El inventor Eli Whitney, en el año 1799, introdujo el concepto de la estandarización en la fabricación de armamento para el gobierno de los Estados Unidos. En esa época cada arma era fabricada manualmente por un único trabajador, por lo que cada arma era única y no podía arreglarse fácilmente si se rompía. Whitney, haciendo uso del concepto de las tolerancias, ordenó la fabricación de piezas que fueran intercambiables entre un arma y otra. Esto permitió reducir los tiempos de montaje y también facilitó la reparación y la sustitución de piezas.

En 1901, Ransom Olds, propietario de la compañía automovilística Oldsmobile, fue considerado un pionero al usar, por primera vez, el concepto de línea de montaje para la fabricación en masa de automóviles. La introducción del concepto de línea de montaje permitió a su empresa aumentar la producción en un 500% en un año. El modelo Curved Dash pudo producirse a un ritmo excepcionalmente alto de 20 unidades al día. Su coche fue el primero en producirse en grandes cantidades.

Unos años más tarde, en 1913, Henry Ford perfeccionó la línea de montaje con la instalación de cintas transportadoras para la producción del modelo de automóvil Ford T. Con este método, el Modelo Ford T podía producirse cada noventa minutos, es decir, un total de casi dos millones de unidades en uno de sus mejores años. Sus avances tuvieron una gran repercusión y fijaron las bases de la producción en línea de la era moderna.

A lo largo de las décadas de 1950 y 1960, ingenieros de todo el mundo experimentaron

con la robótica como medio de desarrollo industrial. General Motors instaló su propio brazo robótico para ayudar en las líneas de montaje en 1961. En 1969, el ingeniero de Stanford Victor Scheinman creó el brazo Stanford, un robot de 6 ejes que podía mover y ensamblar piezas con un patrón repetido continuo. Este invento amplió el uso de los robots en formas que siguen aplicándose en el ensamblaje moderno.

3.1.2. Conceptos principales

Según Becker y Scholl (2006), los conceptos más relevantes de los problemas en las líneas de montaje son:

- Tarea i : unidad de trabajo indivisible a la que se asocia un tiempo de proceso. El proceso completo de fabricación de un producto se compone de un conjunto de N tareas ($i = 1, \dots, N$).
- Tiempo de proceso de la tarea i : tiempo necesario para la realización de la tarea i .
- Relaciones de precedencia: se definen conforme a las restricciones sobre el orden en que pueden ser realizadas las distintas tareas en una línea de montaje. Una tarea no puede ser realizada si todas sus tareas precedentes no han sido realizadas previamente. Las relaciones de precedencia se pueden representar, por ejemplo, por medio de un grafo de precedencias.
- Estación de trabajo j : componentes de la línea de montaje en los que se procesan las tareas. Las estaciones de trabajo están ocupadas por uno o más operadores, humanos o robots, que tienen a su disposición maquinaria y mecanismos de proceso especializados para poder completar las tareas que les han sido asignadas. En una línea de montaje hay un máximo de UB^M estaciones de trabajo ($j = 1, \dots, UB^M$).
- Carga de trabajo de la estación j : conjunto de tareas asignadas a la estación de trabajo j .
- Tiempo de trabajo de la estación j : sumatorio de los tiempos de proceso de las tareas asignadas a la estación j .
- Tiempo de ciclo (CT): tiempo disponible en cada estación para completar las tareas asignadas.
- Tiempo ocioso de la estación j : tiempo en el que los trabajadores de una estación están inactivos, es decir, no procesan ninguna tarea. En el caso de que haya un

solo trabajador, el tiempo ocioso se calcula como la diferencia entre el tiempo de ciclo y el tiempo de trabajo de la estación j .

- **Eficiencia de la línea (η):** sumatorio de los tiempos de proceso de todas las tareas dividido por el tiempo total disponible. En el caso de que haya un solo operario en cada estación, el tiempo total disponible se puede obtener del producto del tiempo de ciclo CT por el número de estaciones totales.

3.1.3. Clasificación de las líneas de montaje

Las líneas de montaje se pueden clasificar según diferentes criterios, en función de las características de la línea a las que se haga referencia. A continuación, se presentan distintas clasificaciones propuestas en la literatura por Ghosh y Gagnon (1989), Becker y Scholl (2006) y Becker y Scholl (2009):

- **Clasificación según el tipo de producto que se procesa:**
 - **Líneas simples:** líneas que procesan un único tipo de producto. Las estaciones ejecutan las mismas tareas de manera cíclica.
 - **Líneas mixtas:** líneas mediante las cuales es posible producir diferentes modelos de un mismo producto simultáneamente. Las tareas a realizar son muy parecidas para cada uno de los modelos, lo cual implica que los tiempos de preparación para pasar de fabricar un modelo a otro son considerados negligibles.
 - **Líneas multi-modelos:** líneas que permiten fabricar diferentes productos en una misma línea; en este caso, los procesos de fabricación difieren significativamente. Por esta razón, cada producto se produce por lotes y se tienen en consideración los tiempos de preparación de la línea para cambiar de un producto a otro (*setup time*). En la Figura 1 se adjunta un ejemplo de las distintas líneas de montaje según el tipo de producto que se procesa.

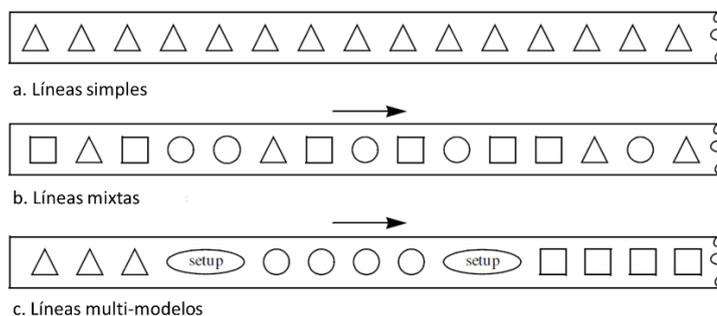


Figura 1. Representación de las distintas líneas de montaje según el tipo de producto procesado. Fuente: Becker y Scholl (2006)

- **Clasificación según la duración de las tareas**

- Líneas deterministas: los tiempos de proceso de las tareas se caracterizan por ser conocidos e invariantes.
- Líneas estocásticas: los tiempos de proceso de las tareas son variables aleatorias o probabilísticas. En este tipo de líneas la variabilidad en el tiempo de proceso es considerada significativa.
- Líneas dependientes: los tiempos de proceso de las tareas son dependientes de alguna variable del proceso. Por ejemplo, entre otros, del tipo de estación, de la secuencia en la que se realizan las tareas, del tipo de operador o de la cantidad de operadores en la estación.

- **Clasificación según su arquitectura**

- Línea serial: consiste en un conjunto de estaciones simples distribuidas en serie. Las unidades pasan consecutivamente de una estación a otra. La cinta transportadora suele ser el mecanismo utilizado para transportar las unidades de producto entre estaciones.
- Línea con estaciones en paralelo: línea única con la posibilidad de incorporar, en alguna fase, dos o más estaciones idénticas que realicen las mismas tareas de forma simultánea, como se representa en la Figura 2. Una de las ventajas de este tipo de línea es que permite procesar aquellas tareas cuya duración es mayor que el tiempo de ciclo. Asignando dichas tareas a las estaciones en paralelo, se consigue reducir el tiempo de proceso de manera proporcional al número de estaciones en paralelo.

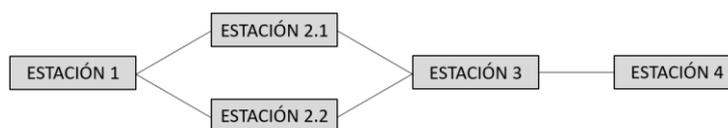


Figura 2. Representación de una línea única con las estaciones 2.1 y 2.2 en paralelo.
Fuente: propia

- Líneas paralelas: formado por distintas líneas colocadas en paralelo. Este tipo de líneas son de utilidad para la fabricación de modelos múltiples, para los que es posible asignar cada modelo, o familia de modelos, a una línea distinta. Asimismo, los operarios pueden trabajar en más de una estación a la vez (véase Figura 3), pudiendo reducir, de esta forma, el número de operarios respecto a si se procesara cada uno de los productos por separado.

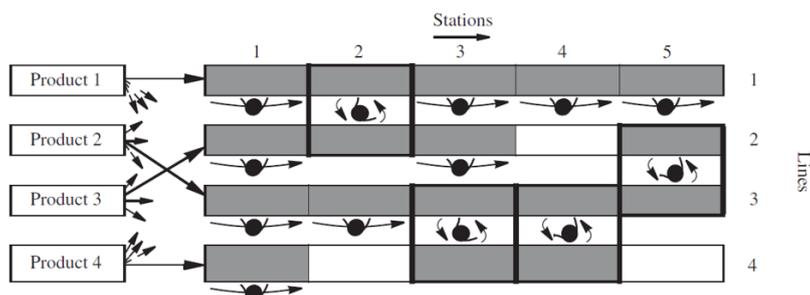


Figura 3. Representación de líneas paralelas con trabajadores que pueden trabajar en más de una línea. Fuente: Scholl y Boysen (2009)

- Línea de dos lados o *two-sided*: líneas de montaje con posibilidad de operar en la parte derecha o en la izquierda de la línea. En vez de tener una estación simple, se compone de pares de estaciones opuestas situadas en cada uno de los lados de la línea (estación derecha y estación izquierda) que trabajan de manera simultánea sobre una misma unidad de producto (véase Figura 4, dibujo intermedio).

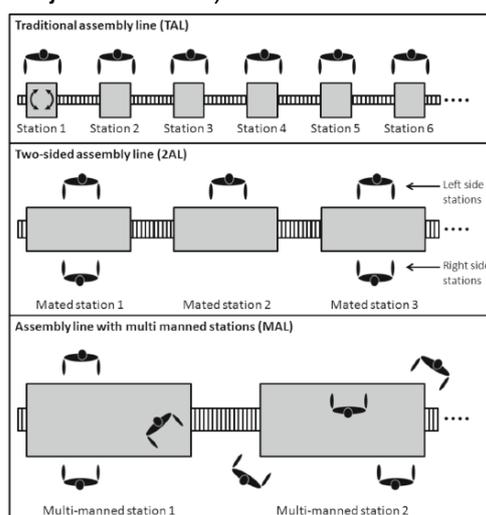


Figura 4. Comparación entre la configuración de la línea serial, la línea de dos lados y la línea *multi-manned*. Fuente: Kellegöz (2017)

- Líneas con múltiples operarios o *multi-manned*: líneas cuyas estaciones pueden estar compuestas por más de un operario (véase Figura 4, dibujo inferior). Cada uno de los operarios puede realizar simultáneamente tareas sobre la misma unidad de producto, teniendo en consideración que no interfieran entre ellos y que se cumplan las relaciones de precedencias entre tareas. Las líneas con múltiples operarios suelen ser empleadas para la fabricación de productos de grandes dimensiones, como los automóviles.
- Línea circular o cerrada: está formada por un cinta circular o cerrada que conecta todas las estaciones. Las unidades de producto circulan por la cinta mientras los operarios las extraen para procesarlas, y las devuelven

nuevamente a la cinta cuando han finalizado. El proceso termina en la estación donde se realiza la última tarea, después de la cual el operario deposita la unidad de producto fuera de la cinta.

- **Línea en forma de U:** línea cuya disposición en forma de U permite incrementar la flexibilidad del sistema de montaje, puesto que las estaciones pueden trabajar en dos segmentos de la línea de montaje que se encuentren uno delante del otro (como se puede observar en la Figura 5). Este tipo de configuración permite un mejor equilibrado de la carga de las estaciones, debido al mayor número de combinaciones de asignaciones posibles de tareas a estaciones.

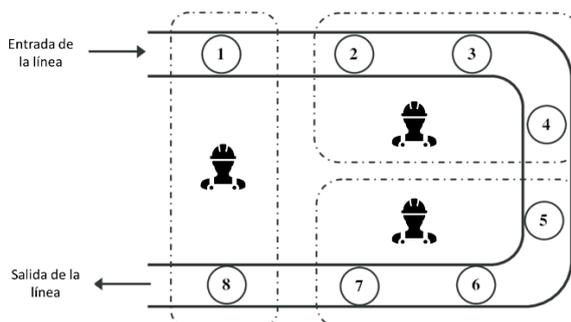


Figura 5. Representación ejemplo de una línea de montaje en forma de U. Fuente: propia

- **Clasificación según el tipo de flujo de las unidades de producto**
 - **Línea síncrona:** todas las estaciones tienen el mismo tiempo de ciclo. En consecuencia, las unidades de producto pasan de una estación a otra al mismo instante, haciendo innecesaria la instalación de estaciones de almacenamiento temporal o *buffers* entre estaciones.
 - **Línea asíncrona:** el tiempo de ciclo no es común, por lo que las unidades de producto se procesan con frecuencias diferentes en función de la estación. Es necesaria la incorporación de *buffers* entre estaciones para almacenar las unidades de producto en curso. Por consiguiente, en este tipo de configuración, también se debe decidir acerca de la localización y el tamaño de los *buffers*.
 - **Líneas de alimentación:** líneas suplementarias que procesan previamente ciertas componentes del producto, que luego alimentan la línea de montaje principal. Para este tipo de líneas es necesario realizar la sincronización de la tasa de producción.
- **Clasificación según el tipo de operador**
 - **Línea manual:** línea cuyos operadores son humanos, independientemente de si la línea está, o no, automatizada.
 - **Línea robotizada:** líneas automatizadas en las cuales los operadores son

robots.

- **Clasificación según la disciplina de entrada de las unidades de producto a la línea**
 - Línea de entrada fija: las unidades de producto llegan a la línea en intervalos de tiempo regulares. En el caso de las líneas síncronas, este intervalo equivale al tiempo de ciclo.
 - Línea de entrada variable: las unidades de producto llegan a la línea en intervalos de tiempo variables.

3.2. Tipos de problemas de líneas de montaje

Los problemas de líneas de montaje pueden ser clasificados en función del objetivo que se desea alcanzar y en función de las restricciones a considerar. En la literatura, dichos problemas suelen clasificarse en dos grupos: problemas de diseño de líneas de montaje (*Assembly Line Design Problem*, ALDP) y problemas de equilibrado de líneas de montaje (*Assembly Line Balancing Problem*, ALBP).

3.2.1. Problemas de diseño de líneas de montaje (ALDP)

Diseñar una línea de montaje es un proceso complejo debido a que se deben tomar un elevado número de decisiones interrelacionadas, cuya repercusión se extiende a múltiples objetivos. En consecuencia, la resolución de este tipo de problemas resulta difícil de automatizar.

En el diseño de una línea de montaje se deben tener en consideración conceptos como la distribución de los espacios de almacenamiento temporal, los sistemas de transporte entre estaciones y los equipos de manufactura para cada operación.

Además, en el proceso de resolución es conveniente tomar las decisiones de forma simultánea. Sin embargo, en la mayoría de los casos dichos problemas se resuelven de forma secuencial, lo cual implica que cada decisión posterior se ve condicionada por las que se adoptaron previamente.

El desarrollo de nuevas tecnologías ha confluído en la aparición de nuevos problemas de diseño de líneas de montaje, así como en la necesidad de rediseñar las líneas ya existentes. El incremento del grado de automatización, la aparición de nuevos equipos de proceso y la tendencia a reducir la vida útil de las líneas, para adaptarse a entornos cambiantes, son algunos de los factores a tener en consideración.

3.2.2. Problemas de equilibrado de líneas de montaje (ALBP)

Los problemas de equilibrado de líneas de montaje consisten en distribuir un conjunto de tareas, necesarias para la fabricación del producto, entre el conjunto de estaciones que componen la línea de montaje.

Los problemas de equilibrado de líneas suelen tener objetivos variados: minimizar el tiempo de ciclo, minimizar el número de estaciones, minimizar el coste global de la línea, equilibrar las cargas de trabajo entre estaciones, etcétera. De hecho, el objetivo también puede ser el resultante de combinar varios criterios. Para alcanzar dichos objetivos, es necesario tener en consideración las diferentes restricciones, entre las que destacan las relaciones de precedencias o incompatibilidad entre tareas.

Los distintos tipos de problemas de equilibrado de líneas de montaje emergen de las combinaciones de objetivos y restricciones. Una de las clasificaciones más conocidas y aceptadas corresponde a la realizada por Baybars (1986), en la que diferencian dos tipos de problemas: el problema simple (*Simple Assembly Line Balancing Problem*, SALBP) y el problema general (*General Assembly Line Balancing Problem*, GALBP).

- **Simple Assembly Line Balancing Problem (SALBP)**

El problema de equilibrado simple de líneas (SALBP) es el más conocido y estudiado en la literatura. Sus características principales son:

- Producción en masa de un único producto homogéneo.
- Tareas indivisibles.
- Relaciones de precedencia entre tareas que deben ser respetadas.
- No hay diferencias entre las estaciones, que están equipadas con los mismos componentes. Las tareas pueden ser asignadas a cualquier estación.
- Línea de montaje simple, es decir, procesa un único tipo de producto.
- Línea de montaje determinista, puesto que los tiempos de proceso de las tareas son conocidos e invariables.
- Línea de montaje serial, ya que consiste en un conjunto de estaciones simples distribuidas en serie.
- Línea de montaje síncrona, ya que todas las estaciones tienen el mismo tiempo de ciclo.
- Línea de montaje de entrada fija, debido a que las unidades de producto llegan en intervalos de tiempos regulares.

En función del objetivo establecido, es posible diferenciar entre cuatro variantes

del problema SALBP:

- SALBP-1: el objetivo es minimizar el número de estaciones necesarias dado un tiempo de ciclo máximo permitido. Se utiliza en situaciones en las que se desea diseñar una nueva línea de montaje en función de una demanda externa conocida.
- SALBP-2: partiendo de un número fijo de estaciones, el objetivo consiste en minimizar el tiempo de ciclo, es decir, maximizar la tasa de producción. Esta variante se suele utilizar cuando ya existe la línea de montaje.
- SALBP-E: el objetivo es maximizar la eficiencia de la línea. Dicho objetivo es equivalente a minimizar el producto del número de estaciones por el tiempo de ciclo.
- SALBP-F: consiste en encontrar una solución factible, dada una combinación cualquiera de tiempo de ciclo y número de estaciones.

- **General Assembly Line Balancing Problem (GALBP)**

El problema general de equilibrado de líneas (GALBP) está conformado por todos los tipos de problemas de equilibrados de líneas no incluidos en el SALBP. Las características de este tipo de problemas los hacen más cercanos a la realidad, lo cual conlleva que, a menudo, sean mucho más difíciles de resolver de forma óptima.

Entre la variedad de problemas GALBP, destacan los siguientes (Boysen *et al.*, 2022):

- Mixed-Model Assembly Line Blancing Problem: equilibrado de líneas con modelos mixtos, es decir, equilibrado de líneas en las que es posible producir diferentes modelos de un mismo producto de manera simultánea. Los distintos modelos presentan tareas comunes y se caracterizan por tener tiempos de preparación que se pueden considerar negligibles (Ding *et al.*, 2006).
- Multi-Model Assembly Line Blancing Problem: equilibrado de líneas en las que es posible fabricar diferentes productos, cuyos procesos de fabricación pueden diferir de manera significativa. Es necesario considerar los tiempos de preparación de la línea para cambiar de un producto a otro (Pastor *et al.*, 2002).
- Robotic Assembly Line Balancing Problem (RALBP): consiste en el equilibrado de líneas de montaje robotizadas (Chutima, 2022).
- Multiproduct Parallel Assembly Line Balancing Problem (PALBP): líneas paralelas, cada una de ellas a cargo de fabricar un tipo de producto. Usualmente, todas las líneas comparten el mismo tiempo de ciclo y el objetivo es la minimización del número de operarios, los cuales pueden

trabajar en dos líneas adyacentes (Lusa, 2008).

- Sequence-dependent Assembly Line Balancing and Scheduling Problems (SUALBSP): problema que resuelve, simultáneamente, el equilibrado de la línea y la secuenciación de las tareas en cada estación. Los tiempos de preparación entre dos tareas ejecutadas en sucesión directa por el mismo trabajador son considerados. Si estas tareas no corresponden a la misma posición de montaje o requieren herramientas adicionales, se produce un incremento en el tiempo de proceso de dichas tareas (tiempo de preparación). En este caso, los tiempos de preparación de las tareas dependen de la secuenciación (Andrés *et al.*, 2008).
- Sequence-dependent Assembly Line Balancing Problem (SDALBP): línea de montaje en la que los tiempos de proceso de las tareas varían según el estado de ensamblaje del producto. A diferencia del SUALBSP, los incrementos de tiempo de las tareas no dependen de la sucesión directa de tareas, sino que consideran el estado de ensamblaje de la unidad de producto (Scholl *et al.*, 2008).
- Alternative Subgraphs Assembly Line Balancing Problem (ASALBP): problema en el que se tiene en consideración que para generar un producto final no existe un plan de montaje único, sino que las unidades de producto se pueden ensamblar en diferentes órdenes alternativos. Cada variante de montaje consiste en un orden particular de proceso de tareas, que se representa mediante un subgrafo de precedencias. En consecuencia, las relaciones de precedencia no son fijas, sino que dependen del subgrafo seleccionado, que también determina los tiempos de proceso de las tareas (Capacho *et al.*, 2009).
- Ergonomic Assembly Line Balancing Problem: problema en el que se pone la atención en optimizar los aspectos ergonómicos de la línea de montaje. En la última década, los aspectos ergonómicos han ganado cada vez más atención en la práctica y la investigación, debido al envejecimiento de la mano de obra, especialmente en muchos países industrializados (Otto y Battaïa, 2017).
- Accessibility Windows Assembly Line Balancing Problem (AWALBP): línea en la que las áreas de trabajo de las estaciones no son lo suficientemente extensas como para que se pueda acceder a toda la unidad de trabajo al mismo tiempo. En este caso, se requiere considerar una denominada ventana de accesibilidad móvil, es decir, las posiciones de montaje que están al alcance de cada estación en un determinado momento del tiempo (García-Villoria *et al.*, 2018).
- U-line Assembly Line Balancing Problem (UALBP): líneas colocadas en forma de U, las cuales permiten una mayor flexibilidad puesto que las

estaciones pueden trabajar en dos segmentos diferentes de la línea de montaje (Delice *et al.*, 2017).

- Two-sided Assembly Line Balancing Problem (2ALBP): líneas de montaje de dos lados, con posibilidad de operar en la parte derecha o en la izquierda de la línea. Las tareas deben ser asignadas a uno de los dos lados de la línea (Make *et al.*, 2017).
- Multi-manned Assembly Line Balancing Problem (MALBP): líneas con la posibilidad de incorporar varios operarios en una misma estación. Los trabajadores, siempre y cuando no se interfieran y se respeten las relaciones de precedencias, pueden efectuar distintas tareas de forma simultánea.

Esta tesis doctoral consiste en la resolución de un caso particular del problema de equilibrado de líneas MALBP, de las cuales se darán más detalles en los apartados sucesivos. En la Figura 6 se muestra la clasificación del problema MALBP en el estudio de las líneas de montaje:

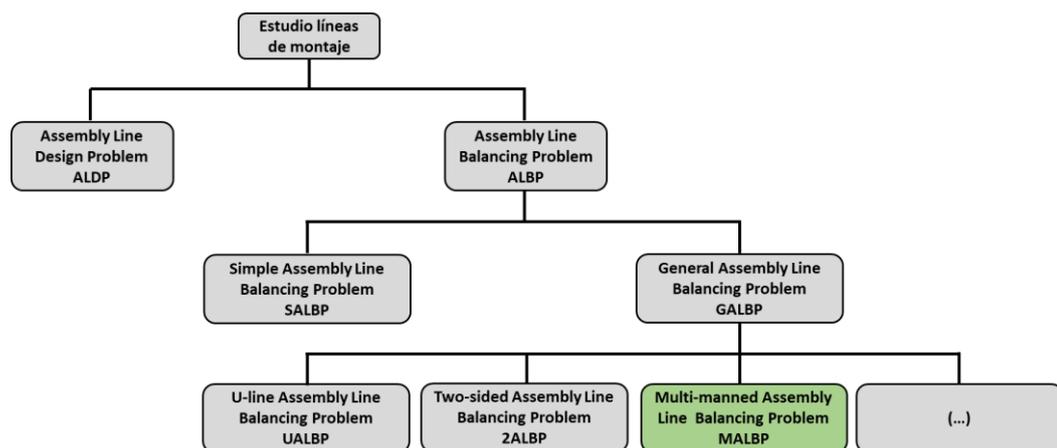


Figura 6. Clasificación del MALBP en el estudio de líneas de montaje. Fuente: propia

4. Problema de equilibrado de líneas con más de un operario por estación (MALBP)

4.1. Introducción

El problema de equilibrado de líneas MALBP (*Multi-manned Assembly Line Balancing Problem*) se caracteriza por considerar la posibilidad de incorporar varios operarios en una misma estación. Las líneas de montaje con múltiples operarios en una estación son comunes en las industrias que fabrican productos de grandes dimensiones, como por ejemplo el sector automovilístico (Figura 7).

En cada estación de trabajo del MALBP, durante un periodo de tiempo conocido como tiempo de ciclo, se pueden encontrar múltiples operarios realizando de forma simultánea distintas tareas sobre un mismo producto, siempre y cuando dichos trabajadores no se interfieran y se respeten las relaciones de precedencias.

Una de las ventajas de las líneas MAL (*Multi-manned Assembly Line*) es que pueden reducir el número de estaciones necesarias, disminuyendo así la longitud de la línea de ensamble y mejorando la utilización del espacio.

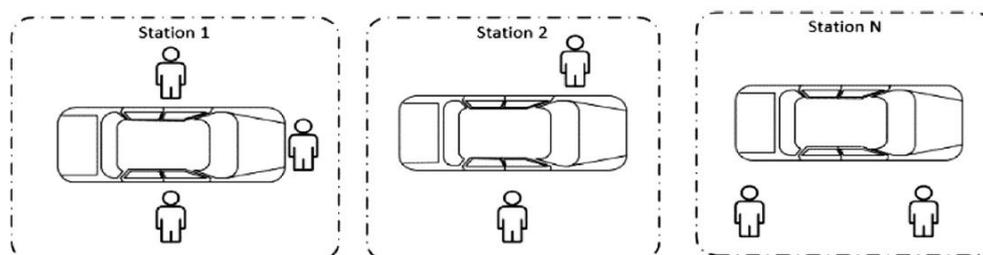


Figura 7. Representación ejemplo de una línea de montaje MALBP. Fuente: Yilmaz y Yilmaz (2015)

Las características principales del problema de equilibrado de líneas MALBP son:

- Cada tarea debe ser realizada por un único trabajador.
- Cada trabajador puede realizar una única tarea simultáneamente.
- El tiempo de proceso de una tarea es independiente del trabajador al que es asignada. Giglio *et al.* (2017) presentan una variante con trabajadores cualificados y no cualificados, que se diferencian por realizar las tareas con tiempos de proceso diferentes.
- Es posible encontrar más de un trabajador por estación. El número máximo de trabajadores que pueden ser asignados a una misma estación viene definido por

la máxima “concentración de operarios” factible (UB^0). Este valor máximo es considerado el mismo para todas las estaciones y se establece durante la fase de diseño del sistema según el tamaño del producto, la disponibilidad de las herramientas y el diseño de las estaciones.

- Los tiempos de desplazamiento de los operarios dentro de cada estación son considerados negligibles.
- No se permiten estaciones, ni líneas, en paralelo, puesto que se trata de una variante de una línea serial.
- Cada estación puede tener un número de trabajadores diferente asignado.
- Las relaciones de precedencia entre tareas deben ser respetadas. A diferencia de las líneas de montaje con un solo operario por estación, en el MALBP se debe tener en consideración que dichas relaciones de precedencia sean adicionalmente respetadas cuando los operarios de una misma estación trabajan de manera simultánea.
- Los tiempos de proceso de las tareas son considerados valores enteros y deterministas. Existe una variante en la que los tiempos son dependientes del número de operarios (Yazdanparast y Hajihosseini, 2011; Sepahi y Jalali Naini, 2014).
- Todas las herramientas y equipos necesarios están disponibles en cada estación. Chen *et al.* (2018) proponen una variante en la que se considera un número limitado de recursos.
- Se fija un número máximo de estaciones (UB^M), debido a las limitaciones de espacio.

Según el objetivo a optimizar, es posible distinguir entre las siguientes variantes del MALBP (véase Tabla 1):

- MALBP-1: se minimiza el número de estaciones a partir de un tiempo de ciclo fijado.
- MALBP-2: se minimiza el tiempo de ciclo a partir de un número fijo de estaciones.
- MALBP-E: se maximiza la eficiencia de la línea, minimizando el producto del número de estaciones por el tiempo de ciclo.
- MALBP-F: consiste en encontrar una solución factible, dado un tiempo de ciclo y un número de estaciones.

Número de estaciones	Tiempo de ciclo	
	Conocido	Minimizar
Conocido	MALBP-F	MALBP-2
Minimizar	MALBP-1	MALBP-E

Tabla 1. Tabla resumen de las 4 variantes del MALBP. Fuente: propia

En dichas variantes, el número de trabajadores puede considerarse una variable que forma parte de una función multiobjetivo, un dato conocido, o incluso el único objetivo a minimizar. La mayoría de los artículos publicados en la literatura solucionan la variante MALBP-1. Sólo unas pocas publicaciones resuelven la variante MALBP-2.

Según la homogeneidad o heterogeneidad de los productos que son procesados en la línea de montaje, los problemas de equilibrado de líneas MALBP se pueden clasificar en:

- SMALBP (Single-Model MALBP): la línea de montaje está configurada para la producción en masa de un único producto homogéneo.
- MMALBP (Mixed-Model MALBP): la línea de montaje realiza el ensamblaje de distintos modelos de un mismo producto. Esta clase de problemas se presenta, por ejemplo, en la fabricación de distintas líneas de autobuses (Roshani y Ghazi Nezami, 2017).
- MuMALBP (Multi-Model MALBP): la línea de montaje realiza el ensamblaje de distintos productos.

En la mayoría de artículos publicados en la literatura se estudia el ensamblaje de un único producto (SMALBP).

4.2. Estado del arte

Según los trabajos analizados, Dimitriadis (2006) estudia por primera vez el problema de equilibrado de líneas MALBP. Para la resolución del MALBP con un único producto homogéneo (SMALBP) utiliza una heurística de dos niveles basada en una modificación del procedimiento de Hoffmann (1963). El objetivo es minimizar el tiempo de espera total de cada uno de los trabajadores de la estación, lo cual es equivalente a la minimización del número de estaciones (SMALBP-1).

Cevikcan *et al.* (2009) tratan por primera vez un problema de equilibrado MALBP con modelos mixtos (MMALBP), es decir, con distintos modelos de un producto en la misma línea de montaje. En el artículo se formula un modelo matemático para minimizar el número de estaciones (MMALBP-1) y, debido a que el problema es *NP-hard (nondeterministic polynomial-time hard problem)*, se desarrolla un algoritmo heurístico de planificación que incluye la secuenciación y un sistema de traslado de operarios entre estaciones.

Becker y Scholl (2009) consideran un caso especial de SMALBP-1, en el que la estación se divide en posiciones de ensamblaje para evitar interferencias. En cada posición de ensamblaje de una estación sólo es posible asignar un trabajador. Se desarrolla un procedimiento para encontrar una solución exacta basado en el algoritmo *Branch and Bound* (B&B). El objetivo es minimizar el número de operarios.

Chang y Chang (2010) plantean un modelo matemático para el problema de equilibrado MMALBP-1, siendo el número de estaciones el objetivo a minimizar.

Qian y Fan (2011) presentan un modelo matemático para la resolución del problema de equilibrado de líneas SMALBP-1 y proponen un procedimiento metaheurístico basado en un algoritmo genético (*Genetic Algorithm*, GA). La minimización del número de operarios es el objetivo en ambos casos.

Fattahi *et al.* (2011) presentan una formulación matemática para el SMALBP-1, basada en un modelo propuesto por Kim *et al.* (2009), para resolver el problema de equilibrado de líneas de montaje de dos lados (2ALBP). El problema 2ALBP se diferencia del MALBP en que las estaciones del 2ALBP pueden contener un máximo de dos operarios por estación, situados a cada uno de los lados de la línea. La función a minimizar en Fattahi *et al.* (2011) contiene dos objetivos jerárquicos: minimizar los trabajadores como objetivo primario y minimizar las estaciones como secundario. Para resolver ejemplares de medio y gran tamaño, desarrollan un algoritmo metaheurístico basado en colonias de hormigas (*Ant Colony Optimization*, ACO).

Yazdanparast y Hajihosseini (2011) adaptan el modelo matemático de Fattahi *et al.* (2011) con la finalidad de incorporar, por primera vez, una variante del SMALBP-1 en la que los tiempos de las tareas son dependientes del número de trabajadores en las estaciones.

Roshani y Roshani (2012) resuelven por primera vez la variante SMALBP-2, es decir, minimizan el tiempo de ciclo dado un número fijo de estaciones. El número de trabajadores es el objetivo secundario a minimizar. Como proceso de resolución se utiliza de forma iterativa un algoritmo ACO (que Abdolreza Roshani ya había publicado en Fattahi *et al.* (2011) para el SMALBP-1). La primera iteración se resuelve usando una cota inferior del tiempo de ciclo. La heurística finaliza en el caso de que el número mínimo de estaciones encontrado sea inferior o igual al valor fijado; en caso contrario, se incrementa el tiempo de ciclo en una unidad y se realiza una nueva iteración.

Kellegöz y Toklu (2012) reformulan el concepto de incompatibilidad de Becker y Scholl (2009), permitiendo que dos tareas de una misma zona de montaje puedan ser realizadas por dos trabajadores diferentes. Para tratar las incompatibilidades entre tareas, se define una lista de tareas que no pueden ser realizadas simultáneamente. Se utiliza un algoritmo B&B para resolver el problema de equilibrado SMALBP-1 minimizando el número de trabajadores.

Kazemi y Sedighi (2013a) abordan la resolución del SMALBP orientada a minimizar el coste total de una unidad de producto. Por un lado, el coste de la mano de obra se corresponde con el salario asociado a la tarea mejor pagada. Por otro lado, el coste de

capital (maquinaria y materiales) es proporcional al número de estaciones. En el artículo se formaliza un modelo matemático y se desarrolla una metaheurística mediante un GA.

Ese mismo año, Kazemi y Sedighi (2013b) publican un nuevo artículo, basado en el anterior, pero con modelos mixtos como variante (MMALBP). Las diferencias respecto Kazemi y Sedighi (2013a) radican en la incorporación de los costes asociados a la duplicación de las tareas y en añadir un procedimiento de resolución que hace uso de un algoritmo de optimización por nube de partículas (*Particle Swarm Optimization*, PSO).

Roshani *et al.* (2013) plantean un algoritmo de búsqueda mediante recocido simulado (*Simulated Annealing*, SA) para resolver el SMALBP-1, minimizando el número de trabajadores y de estaciones, y maximizando la homogeneidad de carga de trabajo entre operarios, es decir, reduciendo las diferencias de carga entre trabajadores.

Sepahi y Jalali Naini (2014) siguen trabajando en la variante del SMALBP-1 introducida por Yazdanparast y Hajihosseini (2011), en la que los tiempos de proceso de las tareas dependen del número de operarios asignados en una estación. Por un lado, adaptan el modelo matemático de Yazdanparast y Hajihosseini (2011), cambiando la función objetivo para que sólo se minimice el número de estaciones. Por otro lado, proponen un procedimiento heurístico orientado a estación, llamado "Max_S", el cual tiene por objetivo minimizar el número de estaciones. Para cada estación se selecciona el menor número de operarios que permite asignar la mayor cantidad de tareas en la estación.

Kellegöz y Toklu (2015) desarrollan una nueva heurística constructiva para la resolución del problema SMALBP-1. Dicha heurística parte de una solución inicial con el número máximo de estaciones y trabajadores. En cada iteración, se resuelve el mismo problema con la única variación de disminuir en una unidad el número de trabajadores de una de las estaciones. Si no se genera una solución factible, se prueba el mismo procedimiento con la siguiente estación, según un orden de estaciones a probar definido a priori. En cambio, si la solución es factible, se repite el proceso usando esta nueva solución. El método se complementa con un GA en el que se varían los órdenes de prueba de las estaciones y los órdenes de prioridad de las tareas. Además, también se presenta una formulación matemática con el objetivo de minimizar el número de trabajadores.

Roshani y Giglio (2015a), siguiendo el concepto desarrollado por Kazemi y Sedighi (2013a), proponen un modelo matemático orientado a disminuir el coste total de una unidad de producto en una línea MAL, con un único producto homogéneo.

Roshani y Giglio (2015b) utilizan una adaptación del algoritmo SA de Roshani *et al.* (2013) para resolver la variante SMALBP-2. Dado un número máximo de estaciones, el objetivo primario es minimizar el tiempo de ciclo y el objetivo secundario es minimizar el número de

trabajadores. El procedimiento se basa en solucionar el problema de forma iterativa (mediante SA) incrementando en una unidad el tiempo de ciclo, hasta encontrar una solución factible con el número de estaciones fijado.

Zamzam *et al.* (2015) presentan un GA para la resolución del problema SMALBP-1 y proponen un nuevo indicador para determinar el número máximo de operarios que pueden ser asignados a una estación. Este indicador está relacionado con el área de trabajo recomendada para que un operario pueda efectuar las tareas que le han sido asignadas.

Yilmaz y Yilmaz (2015) formulan un modelo matemático para el SMALBP-1 en el que no sólo se minimiza el número de trabajadores y estaciones, sino que también se busca reducir la diferencia de carga de trabajo entre los operarios de una misma estación.

Yilmaz y Yilmaz (2016a) proponen un modelo matemático y una heurística para resolver una versión del SMALBP-1 en la que los trabajadores son distribuidos en equipos por adelantado, debido a la necesidad de realizar ciertas tareas especializadas que requieren más de un operario. La diferencia con trabajos anteriores radica en que en esta variante se asignan directamente equipos, formados por conjuntos de trabajadores, a estaciones.

Yilmaz y Yilmaz (2016b) muestran que algunos de los resultados obtenidos mediante la resolución del modelo matemático de Fattahi *et al.* (2011) son infactibles. El artículo propone restricciones adicionales a incorporar en el modelo matemático para corregir este error, impidiendo así la generación de soluciones infactibles.

Roshani y Giglio (2017) abordan la variante SMALBP-2 formulando un modelo matemático y proponiendo dos algoritmos basados en SA: el SA indirecto (ISA), que resuelve iterativamente el MALBP-1, y el SA directo (DSA), que resuelve directamente el MALBP-2. Por un lado, el ISA se basa en el mismo algoritmo que ambos autores publican en Roshani y Giglio (2015b). Por otro lado, el DSA comienza generando una solución inicial mediante una heurística orientada a estaciones y sucesivamente aplica un algoritmo SA, empleando también un GA para generar soluciones vecinas. Dado un número máximo de estaciones, los procedimientos minimizan el tiempo de ciclo como objetivo principal y el número de trabajadores como secundario. Los resultados muestran que el DSA es más eficaz que ISA en términos de calidad y tiempo de cálculo. Además, DSA obtiene mejores resultados que los procedimientos publicados anteriormente para resolver el MALBP-2.

Kellegöz (2017) propone una nueva formulación matemática para el SMALBP-1 basada en el modelo de Fattahi *et al.* (2011). Además, publica un algoritmo de búsqueda basado en un SA en el que, partiendo de una solución inicial, se generan soluciones vecinas mediante cuatro procedimientos. Estos cuatro procedimientos de generación de soluciones vecinas corresponden a combinar dos criterios (cambiar tareas entre estaciones o insertar

tareas en una estación) con dos posibles ordenaciones de las tareas (ordenadas mediante un criterio establecido o de forma aleatoria).

Giglio *et al.* (2017) presentan un modelo matemático para disminuir el coste total de una unidad de producto (costes de estaciones y trabajadores) en una línea MAL con un único producto homogéneo. Incorporan la variante de distinguir entre distintos tipos de trabajadores según su cualificación, estableciendo el tiempo que necesitan para realizar cada tarea y su salario.

Roshani y Ghazi Nezami (2017) proponen un procedimiento heurístico basado en un SA, muy similar al publicado en Roshani *et al.* (2013), pero para modelos mixtos (MMALBP-1). Además, también formalizan un modelo matemático para el MMALBP-1, basado en adaptar el modelo matemático de Fattahi *et al.* (2011).

Chen (2017) presenta un procedimiento heurístico que utiliza el algoritmo SA para resolver el SMALBP-1. Asimismo, el artículo también incluye la formulación de un modelo matemático basado en el de Fattahi *et al.* (2011), cuya principal diferencia radica en que los dos objetivos a minimizar (número de trabajadores y de estaciones) aparecen normalizados, es decir, se consideran de la misma importancia.

Chen *et al.* (2018) estudian una variante del problema SMALBP-1 en la que los recursos disponibles (maquinaria y herramientas) son limitados. Como dato, se indican los recursos que requiere cada una de las tareas. Dichos recursos son asignados a estaciones. Para resolver este problema proponen un modelo matemático y un procedimiento heurístico que hace uso de un GA.

Sahin y Kellegöz (2019a) abordan una variante del SMALBP-1 en la que los trabajadores de una estación se pueden desplazar a otras. Consideran los tiempos de desplazamiento de los trabajadores negligibles comparados con el tiempo de ciclo, y establecen una distancia máxima que dichos trabajadores pueden recorrer. Formalizan un modelo matemático y un algoritmo de optimización basado en el comportamiento del campo electromagnético (*Electromagnetic Field Optimization*, EFO).

Sahin y Kellegöz (2019b) resuelven el SMALBP orientado a minimizar el coste total de una unidad de producto. El artículo estudia una variante en la que para la realización de cada tarea es necesario hacer uso de un recurso específico. Los distintos recursos pueden ser reutilizados dentro de una misma estación. El artículo propone un modelo matemático y un algoritmo de PSO, con el objetivo de reducir el coste de abrir nuevas estaciones y el coste de los recursos utilizados en la línea de montaje.

Michels *et al.* (2019) proponen un modelo matemático basado en el de Fattahi *et al.* (2011) y resuelven el SMALBP-1, adaptando el algoritmo de Benders (1962) para proponer un

algoritmo de descomposición (*Benders' decomposition algorithm*, BDA) que usa el método de corte combinatorio de Benders (*Combinatorial Benders Cut method*, CBC).

Michels *et al.* (2020) proponen los mismos procedimientos que Michels *et al.* (2019), es decir, un modelo matemático y un algoritmo BDA que usa el método CBC, pero adaptados para resolver el SMALBP-2. En particular, se estudia una variante del MALBP-2 (con el objetivo primario de minimizar el tiempo de ciclo) en la que se fija un número máximo de trabajadores y el número de estaciones no se considera un dato sino una variable a minimizar como objetivo secundario.

Lopes *et al.* (2020) presentan un modelo matemático y un BDA para resolver una variante del SMALBP-1 con fronteras flexibles entre estaciones. En el MALBP general, los trabajadores están en la estación entre un tiempo de inicio y fin que es común para todos ellos. En este caso se elimina esta restricción y los trabajadores pueden comenzar a estar disponibles en instantes de tiempo distintos, siempre y cuando finalicen el trabajo asignado antes de superar el tiempo de ciclo y no excedan el máximo número de trabajadores que pueden operar de manera simultánea. El objetivo de Lopes *et al.* (2020) es minimizar la longitud de la línea.

Yang y Cheng (2020) resuelven el SMALBP-1 considerando los tiempos de preparación de las tareas. En dicha publicación, los tiempos de preparación son dependientes de la secuencia de tareas que se establece en cada estación. Se formula un modelo matemático y se desarrolla un algoritmo de SA basado en el de Roshani *et al.* (2013).

Roshani y Giglio (2020) desarrollan un procedimiento metaheurístico basado en un algoritmo de búsqueda tabú (*Tabu search*, TS) para resolver el SMALBP con el objetivo de minimizar el coste total de una unidad de producto. En particular, se minimizan los costes relacionados con el número de operarios y estaciones.

Esfandyari y Roshani (2020) proponen un modelo matemático para resolver el SMALBP-1, diferenciando entre trabajadores cualificados y no cualificados. En concreto, se considera un número de trabajadores cualificados fijo, mientras que los trabajadores no cualificados se contemplan para satisfacer elevadas demandas estacionales. El problema matemático minimiza el número de trabajadores no cualificados, como primer objetivo, y el número de estaciones como objetivo secundario.

Zhang *et al.* (2020) resuelven la variante del SMALBP-1 que toma en consideración el espacio necesario para realizar cada tarea, el cual se define con un valor numérico. La restricción adicional de esta variante se basa en que el sumatorio de los espacios de las tareas asignadas a una estación no supere el espacio máximo de cada estación (que es un dato del problema). El artículo presenta un modelo matemático y un ACO con el objetivo

primario de minimizar el número de trabajadores y el secundario de minimizar el número de estaciones.

Zamzam y Elakkad (2021) también estudia la variante del SMALBP-1 con las restricciones de espacio introducidas por Zhang et al. (2020), adaptando el GA presentado en Zamzam et al. (2015).

Zhang et al. (2021) resuelven el SMALBP-1 proponiendo un modelo matemático y un algoritmo de optimización basado en aves migratorias (*Migrating Birds Optimisation algorithm*, MBO), integrado con un método de reducción de los tiempos ociosos entre tareas. El objetivo es minimizar el número de trabajadores y, como objetivo secundario, el número de estaciones.

Yang et al. (2022) presentan un modelo matemático y un algoritmo de filtrado de búsqueda en haz (*Filtered Beam Search algorithm*, FBS) para resolver una variante del SMALBP-2 con trabajadores cualificados y no cualificados. Los procedimientos propuestos se centran en la resolución de un caso real de la industria de automoción, con la incorporación de un conjunto de restricciones propias de dicha empresa, en el que el objetivo a minimizar es el tiempo de ciclo.

Sahin y Kellegöz (2022) estudian la variante del SMALBP-1 introducida en Sahin y Kellegöz (2019a), en la cual los trabajadores pueden desplazarse entre estaciones. Se formula un modelo matemático y se presentan un conjunto de procedimientos exactos basados en usar el método CBC. Se minimizan el número de trabajadores y estaciones como objetivos primario y secundario, respectivamente.

A continuación, en la Tabla 2 se clasifican los artículos hasta aquí mencionados que estudian el problema de equilibrado MALBP según: el tipo de MALBP (todos los artículos que abordan la versión del MALBP orientada a minimizar el coste total se clasifican, en la Tabla 2, como MALBP-1 porque consideran un tiempo de ciclo fijo y minimizan los costes del número de estaciones, entre otros costes), el método de resolución propuesto (HP= procedimiento heurístico, el cual puede ser clasificado en *greedy* o metaheurístico (ACO, BDA, EFO, FBS, GA, MBO, PSO, SA o TS); EX= procedimiento exacto, el cual puede tratarse de un MM (modelo matemático), un B&B o un CBC), el objetivo a minimizar y el tiempo de proceso de las tareas:

Artículo	Tipo de MALBP	Método	Objetivo a minimizar	Tiempo de proceso de las tareas
Dimitriadis (2006)	SMALBP-1	HP (<i>greedy</i>)	Estaciones	Determinista
Cevikcan et al. (2009)	MMALBP-1	EX (MM)	Estaciones	Determinista
		HP (<i>greedy</i>)		
Becker y Scholl (2009)	SMALBP-1	EX (B&B)	Trabajadores	Determinista

Artículo	Tipo de MALBP	Método	Objetivo a minimizar	Tiempo de proceso de las tareas
Chang y Chang (2010)	MMALBP-1	EX (MM)	Estaciones	Determinista
Qian y Fan (2011)	SMALBP-1	EX (MM)	Trabajadores	Determinista
		HP (GA)		
Fattahi <i>et al.</i> (2011)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
		HP (ACO)		
Yazdanparast y Hajihosseini (2011)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Dependiente del número de trabajadores
Roshani y Roshani (2012)	SMALBP-2	HP (ACO)	1. Tiempo de ciclo 2. Trabajadores	Determinista
Kellegöz y Toklu (2012)	SMALBP-1	EX (B&B)	Trabajadores	Determinista
Kazemi y Sedighi (2013a)	SMALBP-1	EX (MM)	Costes de producción	Determinista
		HP (GA)		
Kazemi y Sedighi (2013b)	MMALBP-1	EX (MM)	Costes de producción	Determinista
		HP (PSO)		
Roshani <i>et al.</i> (2013)	SMALBP-1	HP (SA)	Trabajadores, estaciones y diferencias de carga operarios	Determinista
Sepahi y Jalali Naini (2014)	SMALBP-1	EX (MM)	Estaciones	Dependiente del número de trabajadores
		HP (greedy)		
Kellegöz y Toklu (2015)	SMALBP-1	EX (MM)	Trabajadores	Determinista
		HP (GA)		
Roshani y Giglio (2015a)	SMALBP-1	EX (MM)	Costes de producción	Determinista
Roshani y Giglio (2015b)	SMALBP-2	HP (SA)	1. Tiempo de ciclo 2. Trabajadores	Determinista
Zamzam <i>et al.</i> (2015)	SMALBP-1	HP (GA)	Trabajadores	Determinista
Yilmaz y Yilmaz (2015)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones 3. Diferencias de carga entre estaciones	Determinista
Yilmaz y Yilmaz (2016a)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
		HP (greedy)		
Yilmaz y Yilmaz (2016b)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
Roshani y Giglio (2017)	SMALBP-2	EX (MM)	1. Tiempo ciclo 2. Trabajadores	Determinista
		HP (SA)		
Kellegöz (2017)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
		HP (SA)		
Giglio <i>et al.</i> (2017)	SMALBP-1	EX (MM)	Costes de producción	Determinista
Roshani y Ghazi Nezami (2017)	MMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
		HP (SA)		
Chen (2017)	SMALBP-1	EX (MM)	Trabajadores, estaciones	Determinista
		HP (SA)		
Chen <i>et al.</i> (2018)	SMALBP-1	EX (MM)	1.Trabajadores	Determinista

Artículo	Tipo de MALBP	Método	Objetivo a minimizar	Tiempo de proceso de las tareas
		HP (GA)	2.Estaciones 3. Uso recursos	
Sahin y Kellegöz (2019a)	SMALBP-1	EX (MM) HP (EFO)	1.Trabajadores 2.Estaciones	Determinista
Sahin y Kellegöz (2019b)	SMALBP-1	EX (MM) HP (PSO)	Costes de producción	Determinista
Michels <i>et al.</i> (2019)	SMALBP-1	EX (MM) EX (CBC)	1.Trabajadores 2.Estaciones	Determinista
Michels <i>et al.</i> (2020)	SMALBP-2	EX (MM) EX (CBC)	1. Tiempo ciclo 2. Estaciones	Determinista
Lopes <i>et al.</i> (2020)	SMALBP-1	EX (MM) EX (BDA)	Longitud de la línea	Determinista
Yang y Chen (2020)	SMALBP-1	EX (MM) HP (SA)	1.Trabajadores 2.Estaciones Trabajadores, estaciones y diferencias de carga operarios	Determinista
Roshani y Giglio (2020)	SMALBP-1	HP (TS)	Costes de producción	Determinista
Esfandyari y Roshani (2020)	SMALBP-1	EX (MM)	1.Trabajadores 2.Estaciones	Determinista
Zhang <i>et al.</i> (2020)	SMALBP-1	EX (MM) HP (ACO)	1.Trabajadores 2.Estaciones	Determinista
Zamzam y Elakkad (2021)	SMALBP-1	HP (GA)	Trabajadores y estaciones	Determinista
Zhang <i>et al.</i> (2021)	SMALBP-1	EX (MM) HP (MBO)	1.Trabajadores 2.Estaciones	Determinista
Yang <i>et al.</i> (2022)	SMALBP-2	HP (FBS)	Tiempo ciclo	Determinista
Sahin y Kellegöz (2022)	SMALBP-1	EX (MM) EX (CBC)	1.Trabajadores 2.Estaciones	Determinista

Tabla 2. Tabla resumen de los artículos que tratan el MALBP. Fuente: propia

4.3. Definición del problema a resolver: MALBP con tiempos de proceso dependientes (MALBP-DTT)

Del análisis del estado del arte del MALBP, se concluye que hay un número reducido de publicaciones que tratan la resolución del problema de equilibrado MALBP considerando que los operarios pueden interferir durante la realización de las tareas.

Becker y Scholl (2009) son los primeros en tener en consideración que los operarios pueden molestarte. Estos autores dividen cada estación en espacios de trabajo independientes, en los que sólo es posible asignar un trabajador.

Kellegöz y Toklu (2012) reformulan este concepto, permitiendo que trabajadores distintos puedan realizar tareas diferentes en una misma zona de trabajo, siempre y cuando dichas tareas sean compatibles. Así pues, en este caso, las interferencias son tratadas a priori con una lista de incompatibilidades entre tareas.

Por otro lado, Yazdanparast y Hajhosseini (2011) y Sepahi y Jalali Naini (2014) tratan las interferencias entre operarios de forma genérica, incrementando el tiempo de realización de las tareas según el número de operarios que se encuentran en la estación.

En esta tesis doctoral se estudia la variante del problema de equilibrado MALBP con tiempos de proceso dependientes del número de operarios en la estación. En la presente tesis, esta variante toma por nombre MALBP-DTT (*Multi-manned Assembly Line Balancing Problem with Dependent Task Times*). Respecto a las características del MALBP (sección 4.1), el MALBP-DTT se diferencia por el hecho que los tiempos de proceso de las tareas no son deterministas, sino que dependen del número de operarios en la estación.

La literatura sobre el MALBP-DTT, según nuestro conocimiento, se reduce hasta la fecha a los dos artículos mencionados anteriormente. En el resto de publicaciones, se asume que el número de trabajadores asignados a cada estación no tiene ningún efecto sobre los tiempos de proceso de las tareas.

Considerar tiempos de proceso dependientes del número de operarios presentes en la estación permite representar situaciones más cercanas a la realidad, en las que los trabajadores se pueden interferir entre ellos y, por consiguiente, el tiempo que necesitan para realizar una tarea puede verse incrementado.

Teniendo presentes las variantes del MALBP según los objetivos a minimizar (sección 4.1), si el objetivo es minimizar el número de estaciones a partir de un tiempo de ciclo fijado, la variante se denomina MALBP-DTT1. En cambio, si se minimiza el tiempo de ciclo a partir de un número fijo de estaciones, el problema lleva por nombre MALBP-DTT2. En ambos casos, el número de trabajadores puede considerarse una variable a minimizar o bien un dato conocido. Según si la línea está configurada para la producción en masa de un producto homogéneo, de distintos modelos de un mismo producto o bien de distintos productos, el problema se denomina SMALBP-DTT, MMALBP-DTT o MuMALBP-DTT, respectivamente.

En la presente tesis doctoral se desarrollan un conjunto de procedimientos de resolución con multiobjetivos jerárquicos. En particular, se proponen procedimientos de resolución para la variante SMALBP-DTT1, con el objetivo primario de minimizar el número de operarios y el objetivo secundario de minimizar el número de estaciones (sección 5), y para la variante SMALBP-DTT2, con el objetivo de minimizar el tiempo de ciclo y el número de

operarios, como objetivos primario y secundario respectivamente (sección 6).

A continuación, se presenta un ejemplo ilustrativo del SMALBP-DTT1. Este ejemplar ha sido generado a partir del conocido ejemplo de Mertens (1967). Se considera un máximo de 3 operarios por estación ($UB^0 = 3$) y un tiempo de ciclo de 100 unidades de tiempo ($CT = 100$). La Figura 8 muestra el grafo de precedencias entre tareas para este ejemplar y, delimitado por los símbolos “<>”, el tiempo necesario para realizar cada tarea según el número de trabajadores (de 1 a 3) en la estación. Por ejemplo, para la tarea 7, “<49,55,65>” indica que son necesarias 49, 55 y 65 unidades de tiempo para realizar la tarea 7 en el caso de que haya 1, 2 o 3 trabajadores en la estación en la que es asignada la tarea 7, respectivamente.

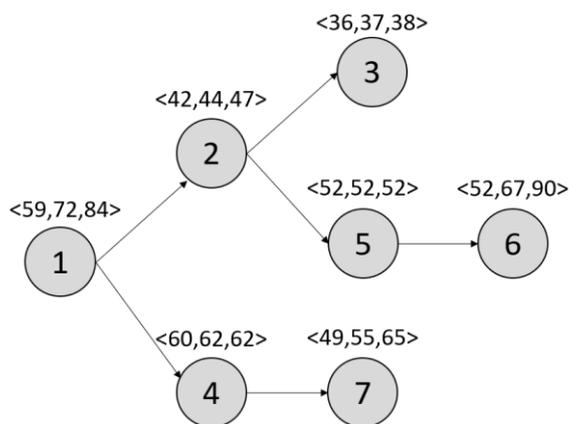


Figura 8. Grafo de precedencias y tiempos de proceso del ejemplo ilustrativo del SMALBP-DTT1. Fuente: propia

Una solución óptima para este ejemplar, minimizando el número de operarios como primer objetivo y el número de estaciones como objetivo secundario, sería utilizar un total de 5 operarios y 3 estaciones, como se muestra en la Figura 9. En particular, en la estación 1 hay un solo operario asignado (que procesa la tarea 1 durante 59 unidades de tiempo), en la estación 2 hay dos operarios (uno de los cuales ejecuta la tarea 4 en 62 unidades de tiempo y el otro realiza las tareas 2 y 5 en un total de 96 unidades de tiempo) y en la estación 3 hay otros dos operarios (uno de ellos elabora las tareas 3 y 7 durante 92 unidades de tiempo, mientras el otro realiza la tarea 6 durante 67 unidades de tiempo). El tiempo de ciclo final es de 96 unidades de tiempo.

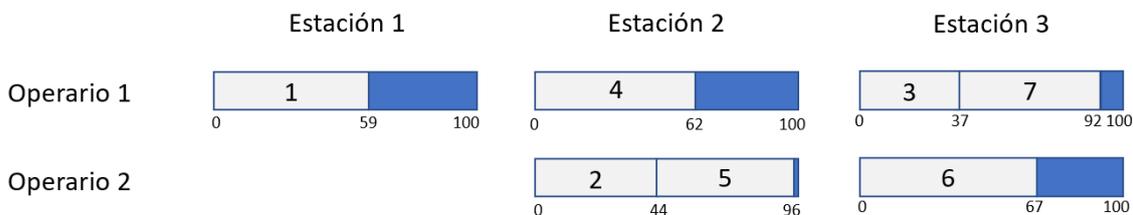


Figura 9. Asignación de tareas a operarios y estaciones, en una solución óptima del SMALBP-DTT1. Fuente: propia

5. Procedimientos de resolución para el SMALBP-DTT1

En el siguiente capítulo se exponen procedimientos de resolución para el SMALBP-DTT1. En la sección 5.1 se presenta un modelo matemático (denominado MILP-DTT1), que corrige y mejora los modelos que han sido propuestos en la literatura. En la sección 5.2 se formulan dos procedimientos de Relajar y Fijar (llamados 1RF y 2RF). En la sección 5.3 se presenta HEUR_PART, un procedimiento heurístico basado en la resolución de un problema de partición con restricciones.

La eficacia de estos cuatro procedimientos (MILP-DTT1, 1RF, 2RF y HEUR_PART) se evalúa en la experiencia computacional de la sección 5.4, con el objetivo de comparar si obtienen mejores resultados los métodos exactos (MILP-DTT1) o los heurísticos (1RF, 2RF y HEUR_PART). HEUR_PART resulta ser el procedimiento que muestra un mejor comportamiento.

Teniendo en cuenta los buenos resultados de HEUR_PART, se presentan nuevas heurísticas que también resuelven problemas de partición con restricciones. En la sección 5.5 se desarrollan un conjunto de variantes de HEUR_PART cuyo objetivo es mejorar los resultados obtenidos: HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE.

En la sección 5.6 se mejora HEUR_PART utilizando el procedimiento Empirically Adjusted Greedy Heuristics (EAGH). Además, se presenta EAGH-CKTL, un nuevo procedimiento para el diseño de heurísticas greedy que se basa en el uso de EAGH combinado con el concepto de cóctel de heurísticas. El objetivo de EAGH-CKTL es diseñar heurísticas greedy mediante EAGH que presenten un buen rendimiento como parte de un cóctel de heurísticas. EAGH (sección 5.6.1) y EAGH-CKTL (sección 5.6.2) se utilizan para diseñar nuevas reglas de prioridad para resolver el SMALBP-DTT1 mediante HEUR_PART, y sus resultados son comparados en la sección 5.6.3.

En las secciones 5.7 y 5.8 se presentan dos nuevos procedimientos que mejoran los resultados obtenidos por HEUR_PART: MULTI_HEUR_PART (sección 5.7), que se basa en la resolución de problemas de partición combinados; y OPWK_PART (sección 5.8), que se basa en modificar las soluciones mediante intercambio y transferencia de tareas entre estaciones.

En la sección 5.9 se realiza una experiencia computacional entre el procedimiento HEUR_PART mejorado con EAGH-CKTL (que se denominará HEUR_PART con R16), MULTI_HEUR_PART y OPWK_PART. El mejor de estos procedimientos, OPWK_PART,

se compara con los procedimientos propuestos en la literatura.

Finalmente, la sección 5.10 contiene las conclusiones de los procedimientos presentados a lo largo de la sección 5 para la resolución del SMALBP-DTT1.

5.1. Modelo matemático (MILP-DTT1)

En esta sección, se presenta un modelo matemático para el SMALBP-DTT1, llamado "MILP-DTT1". El MILP-DTT1 mejora y corrige el modelo propuesto por Yazdanparast y Hajihosseini (2011) y el de Sepahi y Jalali Naini (2014).

Los datos, las variables, la función objetivo y las restricciones se introducen en las secciones 5.1.1-5.1.4, respectivamente. En particular, como se explica en la sección 5.1.4, se añaden restricciones adicionales para evitar la posibilidad de tener soluciones infactibles (Ec. 12 y Ec. 13) y para romper simetrías (Ec. 14 y Ec. 15). Además, con el fin de reducir el tiempo de resolución, se propone un conjunto de cotas para reducir el número de variables y restricciones (sección 5.1.5). La experiencia computacional del MILP-DTT1 se detalla en la sección 5.4.

5.1.1. Datos

UB^M cota superior del número de estaciones (explicado en la sección 5.1.5);

$$j = 1, \dots, UB^M.$$

N número de tareas ($i = 1, \dots, N$). Se introduce una tarea N ficticia cuyo tiempo de proceso es cero (su utilidad es introducida en la sección 5.1.3). La tarea N es sucesora de todas las demás tareas reales y no tiene ninguna sucesora. Así pues, es asignada a la última estación utilizada.

UB^O cota superior del número de operarios por estación; $o = 1, \dots, UB^O$.

CT tiempo de ciclo.

P_{io} tiempo de proceso de la tarea i si hay o operarios en la estación en la que es procesada; $i = 1, \dots, N$, $o = 1, \dots, UB^O$ (siendo $P_{No} = 0 \forall o$).

IPR_i conjunto de predecesoras inmediatas de la tarea i ; $i = 1, \dots, N$.

APR_i conjunto de todas las predecesoras de la tarea i : $APR_i = IPR_i \cup \left(\bigcup_{h \in IPR_i} APR_h \right)$;

$$i = 1, \dots, N.$$

\overline{PR}	conjunto de parejas de tareas que no tienen relaciones de precedencia entre ellas: $\overline{PR} = \{(h \in \{1, \dots, N-1\}, i \in \{h+1, \dots, N\}) \mid h \notin APR_i \wedge i \notin APR_h\}$.
E_i	primera estación a la que puede ser asignada la tarea i (explicado en la sección 5.1.5); $i = 1, \dots, N$.
L_i	última estación a la que puede ser asignada la tarea i (explicado en la sección 5.1.5); $i = 1, \dots, N$.
T_j	conjunto de tareas que pueden ser asignadas a la estación j ; $j = 1, \dots, UB^M$: $T_j = \{1, \dots, N \mid E_i \leq j \leq L_i\}$.

5.1.2. Variables

$t_{ij} \in \mathbb{R}^+$	tiempo de proceso de la tarea i si es asignada a la estación j : $i = 1, \dots, N$; $j = E_i, \dots, L_i$.
$x_{ijk} \in \{0,1\}$	1 si la tarea i es asignada al operario k de la estación j , 0 en caso contrario: $i = 1, \dots, N$; $j = E_i, \dots, L_i$; $k = 1, \dots, UB^O$.
$y_{jo} \in \{0,1\}$	1 si hay o operarios en la estación j , 0 en caso contrario: $j = 1, \dots, UB^M$; $o = 1, \dots, UB^O$.
$w_{jk} \in \{0,1\}$	1 si se usa el operario k de la estación j , 0 en caso contrario: $j = 1, \dots, UB^M$; $k = 1, \dots, UB^O$.
$st_i \in \mathbb{R}^+$	tiempo de inicio de la tarea i : $i = 1, \dots, N$. Este tiempo es local para cada estación, lo que significa que cada vez que una unidad de producto llega a una estación se considera como tiempo 0.
$\theta_{hi} \in \{0,1\}$	variable auxiliar para asegurar que las tareas h e i no son realizadas de manera simultánea por el mismo operador: $(h, i) \in \overline{PR}$.

5.1.3. Función objetivo

La función objetivo minimiza el número de trabajadores como primer objetivo y el número de estaciones como segundo. El número de estaciones se determina mediante la estación a la que se asigna la tarea N ficticia, sucesora de todas las demás y con tiempo de proceso cero. El peso $\frac{1}{UB^M+1}$ garantiza que el segundo término, que representa el número de estaciones, sea el criterio de desempate entre soluciones de un mismo número de trabajadores.

$$[\text{MIN}] \sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} w_{jk} + \frac{1}{UB^M + 1} \cdot \sum_{j=E_N}^{L_N} \sum_{k=1}^{UB^O} j \cdot x_{Njk} \quad (\text{Ec. 1})$$

5.1.4. Restricciones

$$\sum_{j=E_i}^{L_i} \sum_{k=1}^{UB^O} x_{ijk} = 1 \quad i = 1, \dots, N \quad (\text{Ec. 2})$$

$$st_i + t_{ij} \leq CT + (P_{i,UB^O} - P_{i,1}) \cdot \left(1 - \sum_{k=1}^{UB^O} x_{ijk} \right) \quad i = 1, \dots, N; \quad j = E_i, \dots, L_i \quad (\text{Ec. 3})$$

$$\sum_{j=E_h}^{L_h} \sum_{k=1}^{UB} j \cdot x_{hjk} \leq \sum_{j=E_i}^{L_i} \sum_{k=1}^{UB^O} j \cdot x_{ijk} \quad i = 1, \dots, N; \quad h \in IPR_i \quad (\text{Ec. 4})$$

$$st_i \geq st_h + t_{hj} - (CT - P_{h,1} + P_{h,UB^O}) \cdot \left(2 - \sum_{k=1}^{UB^O} x_{hjk} - \sum_{k=1}^{UB^O} x_{ijk} \right) \quad i = 1, \dots, N; \quad h \in IPR_i; \quad j = \max(E_h, E_i), \dots, \min(L_h, L_i); \quad (\text{Ec. 5})$$

$$st_i \geq st_h + t_{hj} - (CT - P_{h,1} + P_{h,UB^O}) \cdot (3 - \theta_{hi} - x_{hjk} - x_{ijk}) \quad (h, i) \in \overline{PR}; \quad k = 1, \dots, UB^O; \quad j = \max(E_h, E_i), \dots, \min(L_h, L_i); \quad (\text{Ec. 6})$$

$$st_h \geq st_i + t_{ij} - (CT - P_{i,1} + P_{i,UB^O}) \cdot (3 - (1 - \theta_{hi}) - x_{hjk} - x_{ijk}) \quad (h, i) \in \overline{PR}; \quad k = 1, \dots, UB^O; \quad j = \max(E_h, E_i), \dots, \min(L_h, L_i); \quad (\text{Ec. 7})$$

$$\sum_{i=1}^N x_{ijk} \leq |T_j| \cdot w_{jk} \quad j = 1, \dots, UB^M; \quad k = 1, \dots, UB^O \quad (\text{Ec. 8})$$

$$\sum_{o=1}^{UB^O} o \cdot y_{jo} = \sum_{k=1}^{UB^O} w_{jk} \quad j = 1, \dots, UB^M \quad (\text{Ec. 9})$$

$$w_{jk} \leq w_{j,k-1} \quad j = 1, \dots, UB^M; \quad k = 2, \dots, UB^O \quad (\text{Ec. 10})$$

$$t_{ij} = \sum_{o=1}^{UB^O} P_{io} \cdot y_{jo} \quad i = 1, \dots, N; \quad j = E_i, \dots, L_i \quad (\text{Ec. 11})$$

$$\sum_{o=1}^{UB^0} y_{jo} \leq 1 \quad j = 1, \dots, UB^M \quad (\text{Ec. 12})$$

$$\sum_{i \in T_j} x_{ijk} \geq w_{jk} \quad j = 1, \dots, UB^M; \quad (\text{Ec. 13})$$

$$k = 1, \dots, UB^0$$

$$\sum_{i \in T_j} x_{ijk} \leq \sum_{i \in T_j} x_{ij,k-1} \quad j = 1, \dots, UB^M; \quad (\text{Ec. 14})$$

$$k = 2, \dots, UB^0$$

$$\sum_{o=1}^{UB^0} y_{jo} \leq \sum_{o=1}^{UB^0} y_{j-1,o} \quad j = 2, \dots, UB^M \quad (\text{Ec. 15})$$

Las restricciones (2) asignan cada tarea exactamente a un trabajador en una estación. Las restricciones (3) garantizan que la carga de trabajo de cada estación no supere el tiempo de ciclo. Las restricciones (4) y (5) aseguran que se cumplan las relaciones de precedencia. Las restricciones (6) y (7) garantizan que dos tareas no puedan procesarse simultáneamente si se han asignado al mismo trabajador. Las restricciones (8) impiden asignar tareas a trabajadores que no estén asignados a una estación. Las restricciones (9) fijan una relación entre las variables w_{jk} y y_{jo} . Las restricciones (10) establecen que los trabajadores sean utilizados en orden creciente de sus índices. Las restricciones (11) fijan el tiempo de proceso de cada tarea en función del número de trabajadores de la estación. Las restricciones (12) se añaden para evitar soluciones infactibles, como se explica en los párrafos sucesivos. Las restricciones (13) evitan el problema encontrado por Yilmaz y Yilmaz (2016b), en el que algunas soluciones utilizan trabajadores sin asignarles tareas. Las restricciones (14) rompen simetrías al imponer que el número de tareas asignadas a un trabajador no sea superior al número de tareas asignadas a un trabajador con un índice k inferior; y las restricciones (15) rompen simetrías al impedir que se asigne tareas a una estación si hay una vacía con un índice j inferior.

En relación con los modelos matemáticos de Yazdanparast & Hajihosseini (2011) y Sepahi & Jalali Naini (2014), se añaden las restricciones de las Ec. 12-15. En los párrafos siguientes se presenta un ejemplo ilustrativo de la necesidad de las restricciones Ec. 12, que se añaden para evitar soluciones infactibles en el modelo propuesto por Yazdanparast y Hajihosseini (2011) y Sepahi y Jalali Naini (2014). La instancia consta de 4 tareas sin ninguna relación de precedencia entre ellas. El número máximo de trabajadores en la misma estación es 3 ($UB^0 = 3$) y el tiempo de ciclo es 100 ($CT = 100$). La Tabla 3 contiene los tiempos de las tareas en función del número de trabajadores en la estación.

Tareas	Número de trabajadores en la estación		
	1	2	3
1	40	55	100
2	40	55	100
3	40	55	100
4	2	3	6

Tabla 3. Tiempos de proceso dependientes del número de trabajadores en la estación. Fuente: propia

A partir de los tiempos de proceso (Tabla 3) se puede observar que no es posible asignar todas las tareas en una sola estación. Sin embargo, si se resuelve este ejemplo utilizando el modelo matemático de Yazdanparast y Hajihosseini (2011) y Sepahi y Jalali Naini (2014), se genera una solución infactible que utiliza 3 trabajadores para procesar todas las tareas en una misma estación.

La infactibilidad puede observarse en la variable y_{jo} , que toma el valor de 1 si y sólo si hay o operarios en la estación j . En este caso, con 3 trabajadores en la estación, la solución correcta sería $y_{11} = 0$, $y_{12} = 0$ y $y_{13} = 1$. Sin embargo, la solución infactible que se obtiene con los modelos propuestos en la literatura es $y_{11} = 1$, $y_{12} = 1$ y $y_{13} = 0$, lo cual no es posible porque implicaría que hubiera un trabajador y dos trabajadores en la misma estación al mismo tiempo. Considerando la Ec. 11 se puede comprobar que utilizando $y_{11} = 1$ y $y_{12} = 1$, en lugar de $y_{13} = 1$, el tiempo de proceso de las tres primeras tareas se reduce de 100 a 95 y de 6 a 5 para la tarea 4:

$$t_{ij} = \sum_{o=1}^3 P_{io} \cdot y_{jo} = 40 \cdot y_{11} + 55 \cdot y_{12} + 100 \cdot y_{13} \quad i = 1, \dots, 3$$

$$t_{4j} = \sum_{o=1}^3 P_{4o} \cdot y_{jo} = 2 \cdot y_{11} + 3 \cdot y_{12} + 6 \cdot y_{13}$$

Esta reducción incorrecta de los tiempos de las tareas permite procesar la tarea 4 (con una duración falsa de 5) después de una de las tres primeras tareas (con una duración falsa de 95), sin superar el tiempo de ciclo $CT = 100$. La introducción de Ec. 12 previene soluciones infactibles, evitando que la suma de variables y_{jo} sea mayor que 1.

5.1.5. Cotas

El número de estaciones de cualquier solución SALBP-1 (en cada una de las cuales hay un solo operario asignado) es una cota superior del número de estaciones del SMALBP-DTT1 (UB^M). Se utiliza el procedimiento heurístico descrito en la sección 5.3.1, el cual resuelve el SALBP-1, para calcular su valor. Otro valor de UB^M , posiblemente menos

ajustado, sería $N - 1$ (recuérdese que N es una tarea ficticia y su tiempo de proceso es cero).

La cota inferior de la estación más temprana en la que se podría asignar la tarea i (E_i) se calcula utilizando las Ecs. 16-18:

$$E_i = \max(E_i^1, E_i^2) \quad (\text{Ec. 16})$$

$$E_i^1 = \left\lceil \frac{(P_{i1} + \sum_{h \in APR_i} P_{h1})}{(CT \cdot UB^0)} \right\rceil \quad (\text{Ec. 17})$$

$$E_i^2 = \max_{h \in APR_i | IPR_h = \emptyset} \left(\max_{p \in PT_{hi}} LB_p^M \right) \quad (\text{Ec. 18})$$

donde:

- PT_{hi} corresponde al conjunto de caminos de sucesores desde la tarea h a la tarea i (ambas incluidas). En este caso, un camino se define como el conjunto de tareas sucesoras que lo componen.
- LB_p^M es una cota inferior del número de estaciones (suponiendo que hay el número máximo de trabajadores en cada estación y, al mismo tiempo, utilizando los tiempos de proceso de un trabajador por estación P_{i1}) para realizar las tareas de la ruta $p \in PT_{hi}$ sin superar el tiempo de ciclo.

La cota superior que determina la última estación en la que se podría asignar la tarea i (L_i) se calcula utilizando las Ecs. 19-21:

$$L_i = \min(L_i^1, L_i^2) \quad (\text{Ec. 19})$$

$$L_i^1 = UB^M + 1 - \left\lceil \frac{(P_{i1} + \sum_{h \in ASC_i} P_{h1})}{(CT \cdot UB^0)} \right\rceil \quad (\text{Ec. 20})$$

$$L_i^2 = UB^M + 1 - \max_{p \in PT_{iN}} LB_p^M \quad (\text{Ec. 21})$$

donde:

- ASC_i es el conjunto de todas las predecesoras de la tarea i : $i = 1, \dots, N$.

5.2. Procedimientos de Relajar y Fijar

En esta sección se proponen dos enfoques del procedimiento matheurístico de Relajar y Fijar (*Relax-and-Fix*), presentado por primera vez en Dillenberger *et al.* (1994).

El procedimiento de Relajar y Fijar es una matheurística usada para la resolución de

programas matemáticos lineales enteros (*Integer Linear Program*, ILP) o mixtos (*Mixed-Integer Linear Program*, MILP). Se trata de una matheurística constructiva basada en resolver, iterativamente, programas lineales considerando la integridad de, únicamente, un número reducido de variables enteras y/o binarias.

Entre todas las variables enteras y/o binarias del programa matemático, se considera como entero únicamente un subconjunto y se relaja el resto de variables.

Cada iteración del procedimiento consta de dos pasos:

1. Decidir el subconjunto de variables de valor todavía no fijado, dentro de las variables enteras y/o binarias, que deben tomar valor entero en esta iteración. Relajar el resto de variables cuyo valor todavía no ha sido determinado.
2. Resolver el programa matemático resultante. Se obtiene un valor entero para el conjunto de variables que deben tomar valor entero en esta iteración. Fijar el valor de dichas variables enteras y/o binarias e ir al paso 1. En caso de que todas las variables enteras y/o binarias ya adopten valores enteros, el procedimiento finaliza.

Para complementar la explicación, se expone un ejemplo de la aplicación del procedimiento de Relajar y Fijar en un problema simple de equilibrado de líneas. Considérese las variables principales binarias de asignación de las tareas a las estaciones. En la primera iteración se relajan todas estas variables menos las relacionadas con la primera estación. Resolviendo el programa matemático resultante se obtiene un valor entero para las variables de asignación de las tareas a la primera estación y, así, se completa dicha estación. A continuación, se fijan estas variables al valor entero obtenido y se pasa a la segunda iteración, en la que se relajan todas las variables faltantes excepto las de asignación de las tareas a la segunda estación; y así sucesivamente.

En esta tesis doctoral se utiliza una variación del procedimiento de Relar y Fijar, en la que es posible fijar solo un subconjunto de variables entre las no relajadas. En el ejemplo introducido en el párrafo anterior consistiría en: no relajar las variables de asignación de las tareas a las, por ejemplo, dos (tres, cuatro, ..., J_{NR}) primeras estaciones; resolver el programa matemático resultante; y fijar, únicamente, las variables de asignación de las tareas a la primera (o primera y segunda, o ..., J_{fix}) estación.

Para resolver el SMALBP-DTT1, se utiliza una adaptación del modelo matemático MILP-DTT1 (sección 5.1). Dependiendo de qué variables se relajen y fijen en cada iteración, se pueden proponer distintos enfoques de Relajar y Fijar. En el MILP-DTT1 hay 4 tipos de variables binarias, que podrían relajarse y fijarse: x_{ijk} , y_{jo} , w_{jk} y θ_{hi} (el modelo no tiene variables enteras).

La relajación de subconjuntos de estas variables, se puede realizar a partir del índice de estaciones j , en el caso de x_{ijk} , y_{jo} y w_{jk} . Sea it la iteración en curso ($it = 1, 2, \dots$), J_{fix} el número de estaciones cuyas variables relacionadas se fijan en cada iteración y J_{NR} el número de estaciones cuyas variables vinculadas no se relajan en cada iteración ($J_{fix} \leq J_{NR}$). De esta forma no se relaja el subconjunto de variables que hacen referencia a las estaciones comprendidas entre $j = (it - 1) \cdot J_{fix} + 1, \dots, (it - 1) \cdot J_{FIX} + J_{NR}$, y si lo hace el subconjunto de variables relacionadas con las estaciones $j = (it - 1) \cdot J_{FIX} + J_{NR} + 1, \dots, UB^M$. A partir de la segunda iteración, las variables relativas a las estaciones $j = 1, \dots, (it - 1) \cdot J_{fix}$ ya están fijadas a los valores obtenidos en las resoluciones previas del modelo MILP-DTT1.

También hay que especificar el caso de las variables θ_{hi} , que se definen para todas las parejas de tareas incluidas en el conjunto \overline{PR} . Para relajar un subconjunto de estas variables a partir de J_{NR} y J_{fix} , se relajan aquellas variables que hacen referencia a parejas de tareas en las que ambas pueden ser asignadas sólo a estaciones relajadas ($E_h > (it - 1) \cdot J_{FIX} + J_{NR}$ y $E_i > (it - 1) \cdot J_{FIX} + J_{NR}$).

Los dos procedimientos de Relajar y Fijar propuestos en esta tesis para resolver el SMALBP-DTT1 (1RF y 2RF) se presentan en las secciones 5.2.1 y 5.2.2. Los parámetros J_{NR} y J_{FIX} deben ser calibrados, y sus valores se muestran en la sección 5.4. La experiencia computacional de 1RF y 2RF también se detalla en la sección 5.4.

5.2.1. Relajar y Fijar, primera propuesta (1RF)

En la primera propuesta de Relajar y Fijar (1RF), se relajan los cuatro grupos de variables binarias: x_{ijk} , y_{jo} , w_{jk} y θ_{hi} . A continuación, se presenta el modelo MILP-DTT1 adaptado para ser utilizado mediante el procedimiento matheurístico de Relajar y Fijar.

Variables relajadas: x_{ijk} , y_{jo} , w_{jk} y θ_{hi} .

Función objetivo: En lugar de utilizar la Ec. 1, se presenta la Ec. 22: una nueva función objetivo. Aunque la Ec. 1 y la Ec. 22 son equivalentes (ambas minimizan el número de trabajadores como objetivo primario y el número de estaciones como objetivo secundario), el uso de la Ec. 22 en 1RF mejora los resultados del procedimiento en relación al número de estaciones. En concreto, la Ec. 1 calcula el número de estaciones utilizando la variable x_{Njk} , que indica la estación donde se encuentra la tarea ficticia N (que es la última estación utilizada). En cambio, con la Ec. 22 el número de estaciones se calcula utilizando la variable y_{jo} .

$$[\text{MIN}] \sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} w_{jk} + \frac{1}{UB^M + 1} \cdot \sum_{j=1}^{UB^M} \sum_{o=1}^{UB^O} y_{jo} \quad (\text{Ec. 22})$$

Restricciones: Ecs. 2-10, Ecs. 12-15.

Además de las restricciones del MILP-DTT1 (Ecs. 2-10 y Ecs. 12-15), la relajación de las variables comentadas anteriormente hace que sea adecuada la incorporación de las siguientes restricciones en la iteración it , con el fin de mejorar la coherencia de los resultados obtenidos por el modelo con las variables relajadas:

$$\sum_{i \in T_j} P_{i1} \cdot x_{ijk} \leq CT \quad \begin{array}{l} j = (it - 1) \cdot J_{FIX} + J_{NR} + 1, \dots, UB^M ; \\ k = 1, \dots, UB^O \end{array} \quad (\text{Ec. 23})$$

$$st_gl_i = CT \cdot \sum_{j=E_i}^{L_i} \sum_{k=1}^{UB^O} (j - 1) \cdot x_{ijk} + st_i \quad i = 1, \dots, N \quad (\text{Ec. 24})$$

$$st_gl_i \geq st_gl_h + P_{h1} \quad i = 1, \dots, N; h \in IPR_i \quad (\text{Ec. 25})$$

$$t_{ij} \geq \sum_{o=1}^{UB^O} P_{io} \cdot y_{jo} \quad i = 1, \dots, N; j = E_i, \dots, L_i \quad (\text{Ec. 26})$$

La restricción (23) implica que la suma de los tiempos de proceso asignados a un trabajador no supere el tiempo de ciclo. Para que la restricción sea lineal, se utilizan los tiempos de proceso con un trabajador, P_{i1} , en lugar de la variable de los tiempos de proceso t_{ij} . Las restricciones (24) y (25) definen st_gl_i , variables adicionales que corresponden al tiempo de inicio global de la tarea i ($i = 1, \dots, N$). Estas variables se utilizan para tener en cuenta las relaciones de precedencia, utilizando los tiempos de proceso como si sólo hubiera un trabajador en la estación. Las restricciones (26) sustituyen a las restricciones (11) y modifican la definición de las variables t_{ij} (tiempo de proceso de la tarea i si es asignada a la estación j), que pasan a estar definidas entre el tiempo mínimo y máximo de la tarea ($P_{i,1} \leq t_{ij} \leq P_{i,UB^O}$).

Distribución del tiempo total de cálculo entre iteraciones: Debido a que el tamaño del modelo matemático a resolver disminuye a medida que avanzan las iteraciones (puesto que en cada iteración ya hay fijadas un número superior de variables), se define una distribución no lineal del tiempo, asignando mayores tiempos de ejecución a las primeras iteraciones:

$$\left\{ \begin{array}{ll} T^{MIN} + \frac{T_k - NI_k \cdot T^{MIN}}{\sum_{i=0}^{NI_k-1} i^\alpha} \cdot (NI_k - 1)^\alpha & \text{Si } NI_k \geq 2 \\ T_k & \text{Si } NI_k = 1 \end{array} \right\} \quad (\text{Ec. 27})$$

donde:

- T_k es el tiempo disponible al inicio de la iteración k .
- NI_k es el número de iteraciones restantes al inicio de la iteración k (también incluida): $NI_k = UB_k^M - k + 1$, donde UB_k^M es la cota superior del número de estaciones al inicio de la iteración k y se calcula mediante el procedimiento descrito en la sección 5.3.1.
- α es un parámetro que controla el desequilibrio de la distribución del tiempo entre las iteraciones restantes ($\alpha \geq 1$): a mayor valor, mayor desequilibrio.
- T^{MIN} es el tiempo mínimo del que se dispondrá en la última iteración.

Los valores de los parámetros α y T^{MIN} deben calibrarse para los experimentos computacionales de 1RF (sección 5.4).

5.2.2. Relajar y Fijar, segunda propuesta (2RF)

En la segunda propuesta de Relajar y Fijar (RF2), se relajan tres de los cuatro grupos de variables binarias del MILP-DTT1: y_{jo} , w_{jk} y θ_{hi} . A diferencia de 1RF, el grupo de variables x_{ijk} no se relaja. En esta subsección, como se realiza para 1RF, se presenta la adaptación del modelo MILP-DTT1 para su uso en la resolución del SMALBP-DTT1 mediante el procedimiento matheurístico de Relajar y Fijar.

Variables relajadas: y_{jo} , w_{jk} y θ_{hi} .

Función objetivo: En este caso se utiliza la Ec.1 (y no la Ec. 22) dado que la variable x_{Njk} (utilizada para calcular el número de estaciones) no se encuentra relajada y, en cambio, sí lo están las variables y_{jo} (utilizadas en la Ec. 22).

Restricciones: Ecs. 2-10, Ecs. 12-15, Ecs. 23-26. Se usan las mismas restricciones que 1RF.

Distribución del tiempo total de cálculo entre iteraciones: se utiliza el mismo enfoque no lineal presentado en la sección anterior para el 1RF. Los valores de los parámetros α y T^{MIN} deben calibrarse para los experimentos computacionales de 2RF (sección 5.4).

5.3. Heurística basada en la resolución de un problema de partición con restricciones (HEUR_PART)

A continuación, se presenta un procedimiento heurístico (denominado HEUR_PART) basado en la resolución de un problema de partición con restricciones. Recuérdese que el objetivo principal es minimizar el número de trabajadores y el objetivo secundario es minimizar el número de estaciones. Para ejemplificar el funcionamiento de HEUR_PART, en las secciones 5.3.1, 5.3.2 y 5.3.3 se utiliza el ejemplo ilustrativo introducido en la sección 4.3 (Figura 8). La experiencia computacional de HEUR_PART se detalla en la sección 5.4.

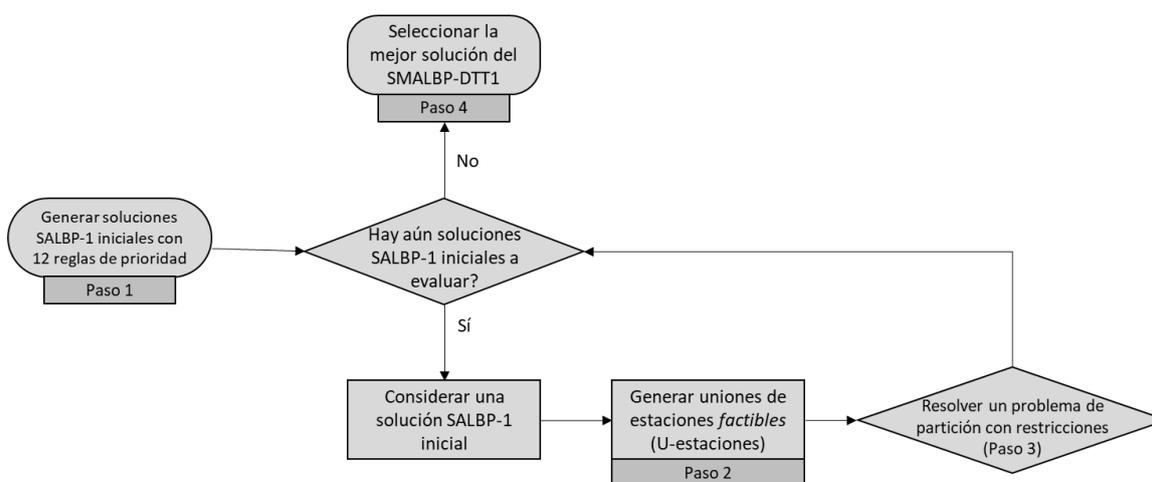


Figura 10. Diagrama de flujo del procedimiento HEUR_PART. Fuente: propia

La Figura 10 presenta el diagrama de flujo de HEUR_PART, que consta de los siguientes pasos que se detallan en los próximos subapartados:

Paso 1. Generar un conjunto de soluciones iniciales resolviendo SALBP-1 (que minimiza el número de trabajadores a partir de un tiempo de ciclo dado) con 12 procedimientos heurísticos diferentes (sección 5.3.1).

Para cada una de las soluciones del paso 1 aplicar los pasos 2 y 3.

Paso 2. Evaluar todas las posibles uniones de estaciones entre ellas y seleccionar sólo aquellas uniones que sean *factibles* (sección 5.3.2). Una unión de estaciones puede concebirse como una estación única en la que se procesan sus tareas. Para evitar ambigüedades, la estación resultante de una unión de estaciones se denominará a partir de ahora U-estación. Cada U-estación se considera factible si se respetan las relaciones de precedencia (sección 5.3.2.1) y si el número de trabajadores necesarios en la U-estación no es superior a la suma del número de trabajadores de cada estación por separado (sección 5.3.2.2). Más concretamente,

el procedimiento presentado en el apartado 5.3.2.2 evalúa la posibilidad de unir las estaciones con el menor número de trabajadores posibles. Es decir, no sólo sin aumentar el número total de trabajadores de cada estación por separado, sino incluso reduciéndolo.

Paso 3. Resolver un problema de partición con restricciones, utilizando programación matemática, con todas las U-estaciones factibles encontradas en el paso anterior (apartado 5.3.3). El modelo matemático genera una solución que minimiza el número de trabajadores como objetivo principal y, como objetivo secundario, el número de U-estaciones seleccionadas, considerando que cada tarea se incluye en una y sólo una U-estación.

Paso 4. La solución con el menor número de trabajadores y, en caso de empate, con el menor número de U-estaciones, se selecciona de entre todas las soluciones encontradas en el Paso 3.

A partir de este momento se hará referencia a la heurística que, dada una única regla de prioridad para ordenar las tareas a seleccionar, encuentra una solución SMALBP-DTT1 siguiendo los pasos de HEUR_PART, como "HEUR_PART_1". HEUR_PART_1 se utilizará más adelante en las secciones 5.6 y 5.7.

5.3.1. Generación de soluciones iniciales del SALBP-1

Para generar una solución inicial del SALBP-1 se utiliza un procedimiento heurístico *greedy* constructivo orientado a estaciones (Scholl y Voß, 1996). En primer lugar, se abre una estación. Iterativamente, se asigna la mejor tarea candidata (siguiendo una regla de prioridad) a la estación abierta en ese momento. Una tarea es candidata si no ha sido asignada previamente a una estación, todas sus tareas precedentes han sido asignadas y puede ser ejecutada sin exceder el tiempo de ciclo. En caso de que no haya más tareas candidatas disponibles, pero aún queden tareas por asignar, se cierra la estación abierta y se abre una nueva estación. El proceso finaliza cuando se han asignado todas las tareas.

Scholl y Voß (1996) establecen 12 reglas de prioridad para la ordenación de las tareas. Dichas reglas se presentan en la Tabla 4 (ISC_i = conjunto de sucesoras inmediatas de la tarea i ; ASC_i = conjunto de todas las sucesoras de la tarea i ; el resto de la notación se define en la sección 5.1.1).

	Reglas de prioridad	Cálculo (para SALBP, $o = 1$)
R1	Mayor tiempo para realizar la tarea y todas sus sucesoras.	$P_{io} + \sum_{h \in ASC_i} P_{ho}$
R2	Mayor tiempo de proceso.	P_{io}

	Reglas de prioridad	Cálculo (para SALBP, $o = 1$)
R3	Mayor número de tareas sucesoras.	$ ASC_i $
R4	Mayor número de tareas sucesoras inmediatas.	$ ISC_i $
R5	Menor cota inferior de la estación a la que puede ser asignada la tarea.	$E_i = \left\lceil \frac{P_{i,o} + \sum_{h \in APR_i} P_{h,o}}{CT} \right\rceil$
R6	Menor cota superior de la estación a la que puede ser asignada la tarea.	$L_i = N + 1 - \left\lfloor \frac{P_{i,o} + \sum_{h \in ASC_i} P_{h,o}}{CT} \right\rfloor$
R7	Menor diferencia entre las cotas superior e inferior de las estaciones a las que puede ser asignada una tarea.	$L_i - E_i$
R8	Menor índice de la tarea.	i
R9	Mayor proporción entre el tiempo para realizar la tarea y todas sus sucesoras, y el número de tareas sucesoras más 1.	$\frac{P_{i,o} + \sum_{h \in ASC_i} P_{h,o}}{ ASC_i + 1}$
R10	Menor proporción entre la cota superior de la estación a la que puede ser asignada una tarea y el número de sucesoras de la tarea más 1.	$\frac{L_i}{ ASC_i + 1}$
R11	Mayor proporción entre el tiempo de proceso y la cota superior de la estación a la que puede ser asignada una tarea.	$\frac{P_{i,o}}{L_i}$
R12	Mayor proporción entre el número de tareas sucesoras y la diferencia entre la cota superior e inferior de las estaciones a las que puede ser asignada una tarea.	$\frac{ ASC_i }{L_i - E_i}$

Tabla 4. Doce reglas de prioridad definidas en Scholl y Voß (1996). Fuente: propia

Dos soluciones son equivalentes si las secuencias de estaciones a las que se han asignado las tareas son permutaciones. Por ejemplo, la Figura 11 muestra las soluciones obtenidas con el ejemplo ilustrativo (Figura 8) siguiendo las reglas de prioridad quinta y octava, definidas en la Tabla 4. Estas soluciones son equivalentes porque la única diferencia entre ellas es el orden de las estaciones 5 y 6. En aras de la eficacia del procedimiento, no se seleccionará más de una solución entre las equivalentes en el Paso 1 de HEUR_PART (ya que las soluciones finales obtenidas a partir de soluciones equivalentes tendrán la misma calidad).

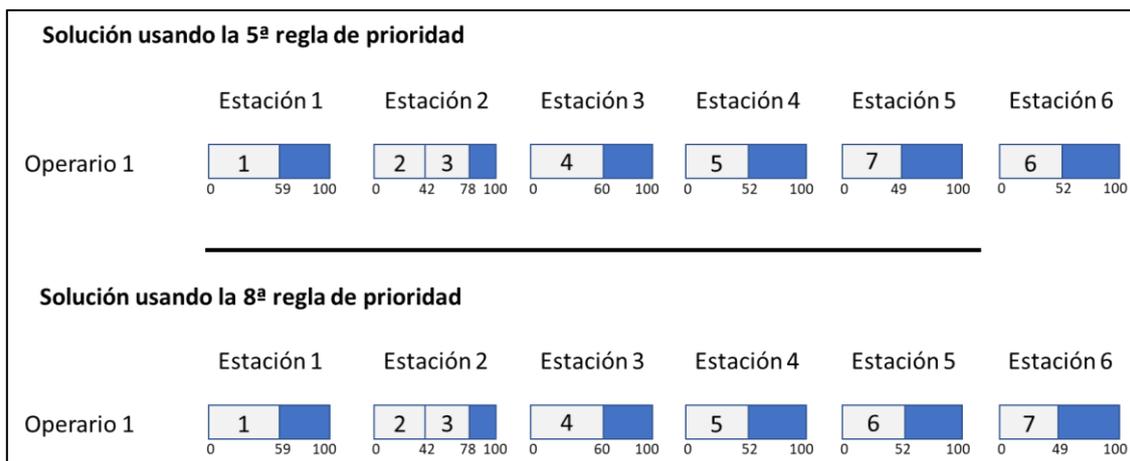


Figura 11. Asignación de tareas a estaciones para dos soluciones SALBP-1 consideradas equivalentes.
Fuente: propia

Se toma, como ejemplo para los siguientes apartados, la solución obtenida utilizando la octava regla de prioridad (Figura 11, solución inferior).

5.3.2. Generación de uniones de estaciones (U-estaciones)

El siguiente paso consiste en generar uniones de estaciones factibles, llamadas U-estaciones, que reduzcan o, al menos, no aumenten la suma del número de trabajadores de cada estación por separado. Para ello, se evalúan todas las uniones posibles entre 2 y UB^0 estaciones. En primer lugar, para cada unión candidata, se evalúa la factibilidad en términos de la satisfacción de las precedencias (sección 5.3.2.1). Si la unión es factible, se utiliza un modelo matemático adaptado de SMALBP, para evaluar la posibilidad de unir las estaciones con el menor número de trabajadores que, al mismo tiempo, no incremente la suma del número de trabajadores de cada estación por separado (sección 5.3.2.2).

5.3.2.1. Factibilidad de la unión de estaciones según precedencias

Una U-estación no es factible si en una sola estación (de las no incluidas en esta unión) hay, al mismo tiempo, tareas sucesoras y predecesoras con respecto a las tareas de la U-estación evaluada. En otras palabras, una U-estación es factible cuando no hay ninguna otra estación (fuera de la unión evaluada) con tareas sucesoras y predecesoras con respecto a las tareas de la U-estación evaluada.

Para la solución obtenida utilizando la octava regla de prioridad (Figura 11, solución inferior) del ejemplo ilustrativo (Figura 8), una U-estación no factible sería unir la estación 2 (tareas 2 y 3) con la estación 5 (tarea 6), porque en la estación 4 se encuentra la tarea 5, que es a la vez sucesora de la tarea 2 y predecesora de la tarea 6.

5.3.2.2. Factibilidad de la unión de estaciones sin incrementar trabajadores

Para las U-estaciones factibles según precedencias, se ejecuta iterativamente un modelo matemático que resuelve un SMALBP con una estación y un número fijo de trabajadores. El proceso iterativo consiste en resolver el modelo matemático aumentando el número de trabajadores (NW) en una unidad hasta encontrar una primera solución factible.

El número de trabajadores NW de la primera iteración es el menor posible, teniendo en cuenta que la suma de los tiempos de proceso de todas las tareas no sea superior al producto del tiempo de ciclo CT por el número de trabajadores. Si el modelo matemático no encuentra una solución factible, el número de trabajadores NW se incrementa en una unidad y el modelo se resuelve de nuevo. El número de trabajadores NW no debe ser superior al número de estaciones individuales que han sido unidas. Sólo si el procedimiento encuentra una solución factible, la unión de estaciones se incluye en la lista de U-estaciones candidatas a formar parte de la solución final. Para la resolución de los modelos matemáticos se establece un límite del tiempo de computación, de $\beta 1$ segundos; el valor $\beta 1$ utilizado para los experimentos computacionales se detalla en la sección 5.4.

A continuación, se presenta el modelo matemático que se resuelve iterativamente:

Datos

N número de tareas asignadas a la U-estación ($i = 1, \dots, N$).

CT tiempo de ciclo.

t_i tiempo de proceso de la tarea i (considerando que en la U-estación hay NW trabajadores):

$$t_i = P_{i,NW}; i = 1, \dots, N.$$

NW número de trabajadores en la U-estación ($k = 1, \dots, NW$):

$$CT \cdot NW \geq \sum_{i=1}^N t_i.$$

IPR_i conjunto de predecesoras inmediatas de la tarea i ; $i = 1, \dots, N$.

APR_i conjunto de todas las predecesoras de la tarea i :

$$APR_i = IPR_i \cup \left(\bigcup_{h \in IPR_i} APR_h \right); i = 1, \dots, N.$$

\overline{PR} conjunto de parejas de tareas que no tienen relaciones de precedencia entre ellas: $\overline{PR} = \{(h \in \{1, \dots, N-1\}, i \in \{h+1, \dots, N\}) | h \notin APR_i \wedge i \notin APR_h\}$.

Variables

$x_{ik} \in \{0,1\}$ 1 si la tarea i es asignada al operario k , 0 en caso contrario:
 $i = 1, \dots, N; k = 1, \dots, NW$.

$st_i \in \mathbb{R}^+$ tiempo de inicio de la tarea $i: i = 1, \dots, N$.

$\theta_{hi} \in \{0,1\}$ variable auxiliar para asegurar que las tareas h y i no sean realizadas de manera simultánea por el mismo operador: $(h, i) \in \overline{PR}$.

Función objetivo

Debido a que se trata de estudiar la factibilidad de una U-estación, la función objetivo se considera igual a una constante, de forma que la ejecución del modelo matemático finaliza cuando se encuentra la primera solución factible.

Restricciones

$$\sum_{k=1}^{NW} x_{ik} = 1 \quad i = 1, \dots, N \quad (\text{Ec. 28})$$

$$st_i + t_i \leq CT \quad i = 1, \dots, N \quad (\text{Ec. 29})$$

$$st_i \geq st_h + t_h \quad i = 1, \dots, N; \quad h \in IPR_i \quad (\text{Ec. 30})$$

$$st_i \geq st_h + t_h - CT \cdot (3 - \theta_{hi} - x_{hk} - x_{ik}) \quad (h, i) \in \overline{PR}; \quad k = 1, \dots, NW \quad (\text{Ec. 31})$$

$$st_h \geq st_i + t_i - CT \cdot (3 - (1 - \theta_{hi}) - x_{hk} - x_{ik}) \quad (h, i) \in \overline{PR}; \quad k = 1, \dots, NW \quad (\text{Ec. 32})$$

$$\sum_{i=1}^N x_{ik} \leq \sum_{i=1}^N x_{i,k-1} \quad k = 2, \dots, NW \quad (\text{Ec. 33})$$

Las restricciones (28) asignan cada tarea a un trabajador. Las restricciones (29) aseguran que la carga de trabajo de la U-estación no supere el tiempo de ciclo. Las restricciones (30) garantizan que se tengan en cuenta las relaciones de precedencia. Las restricciones (31) y (32) aseguran que dos tareas no se procesen simultáneamente si se han asignado al mismo trabajador. Las restricciones (33) rompen simetrías al imponer que el número de

tareas asignadas a un trabajador no sea superior al número de tareas asignadas a un trabajador con un índice k inferior.

Para la solución obtenida considerando la octava regla de prioridad (Figura 11, solución inferior) del ejemplo ilustrativo (Figura 8), la lista de U-estaciones que superan los dos filtros de factibilidad y, por lo tanto, son candidatas para formar parte de la solución final, se muestra en la Tabla 5 (recuérdese que el número máximo de trabajadores en una misma estación es $UB^0 = 3$ y que el tiempo de ciclo es $CT = 100$).

U-estaciones candidatas	Tareas incluidas	Número de trabajadores
Estaciones 2 + 3	2 + 3 + 4	2
Estaciones 2 + 4	2 + 3 + 5	2
Estaciones 2 + 6	2 + 3 + 7	2
Estaciones 3 + 4	4 + 5	2
Estaciones 3 + 5	4 + 6	2
Estaciones 4 + 6	5 + 7	2
Estaciones 5 + 6	6 + 7	2
Estaciones 2 + 3 + 4	2 + 3 + 4 + 5	2
Estaciones 2 + 4 + 6	2 + 3 + 5 + 7	2

Tabla 5. U-estaciones candidatas a ser parte de la solución final. Fuente: propia

En este ejemplo, para las uniones de las estaciones 2, 3 y 4, y de las estaciones 2, 4 y 6, la suma del número de trabajadores de cada estación por separado es igual a tres; pero sólo se necesitan dos trabajadores ($NW = 2$) cuando se realizan dichas uniones de estaciones. La Figura 12 muestra la solución de unir las estaciones 2, 3 y 4.

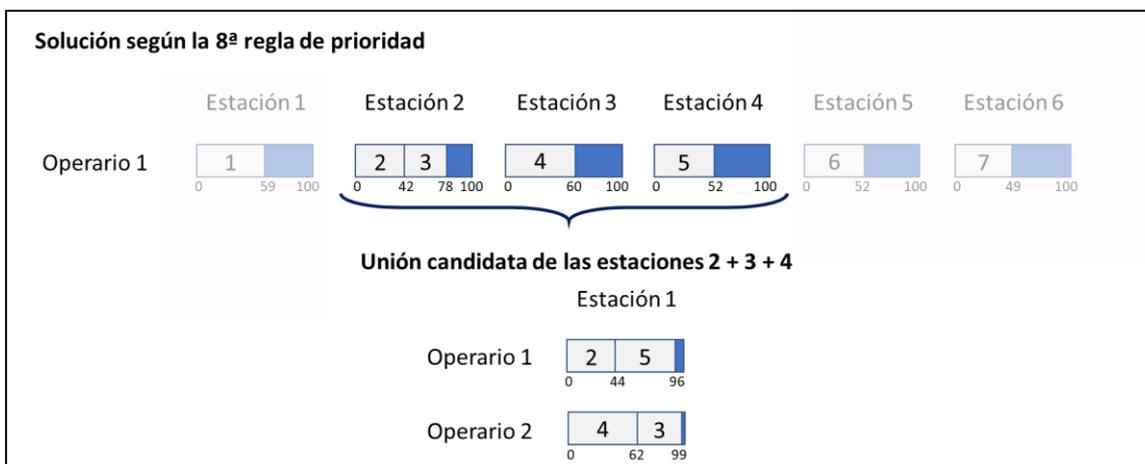


Figura 12. Solución obtenida combinando las estaciones 2, 3 y 4 de la octava regla de prioridad. Fuente: propia

5.3.3. Resolución de un problema de partición con restricciones

Con la lista de U-estaciones candidatas, incluyendo además como U-estaciones

candidatas cada una de las estaciones de la solución inicial del SALBP-1 de forma individual, se propone un modelo matemático para resolver un problema de partición con restricciones entre U-estaciones candidatas. Este modelo proporciona una solución que minimiza el número de trabajadores como objetivo principal y el número de U-estaciones como secundario.

Datos

J número de estaciones obtenidas en la resolución del SALBP-1 ($j = 1, \dots, J$).

S número de U-estaciones candidatas ($s = 1, \dots, S$).

NW_s número de trabajadores en la U-estación s ($s = 1, \dots, S$).

IS_{sj} 1 si en la U-estación s está incluida la estación j , 0 en caso contrario:

$$j = 1, \dots, J; s = 1, \dots, S.$$

\overline{INC} grupos de U-estaciones que no tienen estaciones en común pero que son infactibles por razones de precedencia.

R número de U-estaciones en cada grupo de \overline{INC} ($R = 2, \dots, J$).

$$\overline{INC} = \{(r_1, \dots, r_R) | r_1 \in \{1, \dots, S - (R - 1)\}, \dots, r_R \in \{r_1 + (R - 1), \dots, S\}, r_1, \dots, r_R \text{ infactibles}\}.$$

Dos U-estaciones ($R = 2$) son infactibles si en una de las U-estaciones hay, al mismo tiempo, tareas sucesoras y predecesoras con respecto a las tareas de la otra U-estación. Tres o más U-estaciones ($R = 3, \dots, J$) son infactibles si cada U-estación tiene relaciones de precedencia con sólo otras dos U-estaciones (más concretamente, tiene tareas predecesoras con respecto a las tareas de una U-estación y tareas sucesoras con respecto a las tareas de la otra U-estación) y no hay infactibilidades en ninguna combinación de $2, \dots, R - 1$ U-estaciones.

Variables

$x_s \in \{0,1\}$ 1 si se usa la U-estación candidata s , 0 en caso contrario: $s = 1, \dots, S$.

Función objetivo

La función objetivo minimiza el número total de trabajadores como primer objetivo y el número de U-estaciones seleccionadas como objetivo secundario:

$$[MIN] \sum_{s=1}^S NW_s \cdot x_s + \frac{1}{J+1} \cdot \sum_{s=1}^S x_s \quad (\text{Ec. 34})$$

Restricciones

$$\sum_{s=1}^S IS_{sj} \cdot x_s = 1 \quad j = 1, \dots, J \quad (\text{Ec. 35})$$

$$\sum_{s=r_1}^{r_R} x_s \leq R - 1 \quad (r_1, \dots, r_R) \in \overline{INC} \quad (\text{Ec. 36})$$

Las restricciones (35) garantizan que la solución final incluya cada una de las estaciones obtenidas en la resolución del SALBP-1, una sola vez. Las restricciones (36) impiden asignar U-estaciones infactibles por razones de precedencia.

La solución obtenida con la octava regla de prioridad del ejemplo ilustrativo utiliza 5 trabajadores y 3 U-estaciones (Figura 13): tarea 1 (estación 1), tareas 2, 3, 4 y 5 (unión de las estaciones 2, 3 y 4, véase Tabla 5) y tareas 6 y 7 (unión de las estaciones 5 y 6, véase Tabla 5).

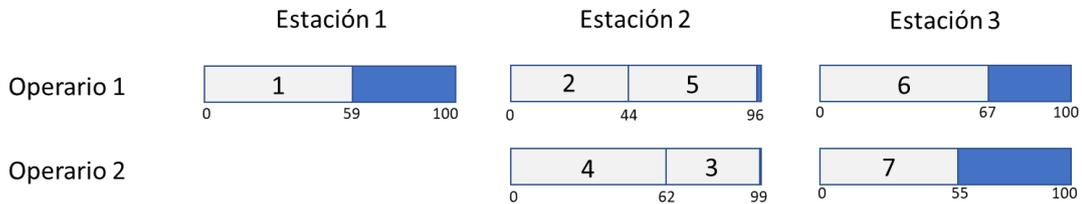


Figura 13. Asignación de tareas a trabajadores y estaciones, para la octava regla de prioridad del ejemplo ilustrativo resuelto con HEUR_PART. Fuente: propia

5.4. Experiencia computacional para comparar MILP-DTT1, 1RF, 2RF y HEUR_PART

En esta sección se evalúan los cuatro primeros procedimientos diseñados para resolver el SMALBP-DTT1: MILP-DTT1, 1RF, 2RF y HEUR_PART. El objetivo de esta experiencia computacional es comparar la eficacia de los métodos exactos (MILP-DTT1) respecto a los heurísticos (1RF, 2RF y HEUR_PART). Según los resultados obtenidos, en las siguientes subsecciones se profundiza el estudio procedimiento que presenta el mejor comportamiento.

En primer lugar, el modelo matemático MILP-DTT1 (sección 5.1), los procedimientos 1RF

y 2RF (sección 5.2) y la heurística HEUR_PART (sección 5.3) se comparan en la sección 5.4.2.1, utilizando los ejemplares indicados en la sección 5.4.1. A continuación, en el apartado 5.4.2.2, se analizan los resultados obtenidos por la heurística HEUR_PART con cada una de las reglas de prioridad que incorpora en el Paso 1 (sección 5.3.1) por separado.

5.4.1. Características de la experiencia computacional

Todas las instancias de los experimentos computacionales han sido ejecutadas en un ordenador con procesador Intel® Core™ i5 CPU 650 @ 3.33 GHz y, para resolver cada instancia, el tiempo de computación se ha fijado en un máximo de 3600 segundos.

En relación con las dos propuestas de los procedimientos de Relajar y Fijar (1RF y 2RF), se ha realizado una prueba computacional preliminar para determinar el número de estaciones que, en cada iteración, no deben ser relajadas, y el número de estaciones que deben ser fijadas (parámetros J_{NR} y J_{FIX} , respectivamente). En dicha prueba computacional preliminar también se han determinado los valores de los parámetros relacionados con la distribución del tiempo (parámetros α y T^{MIN}). Se han utilizado 50 ejemplares de Otto *et al.* (2013): las instancias utilizadas se detallan en el Apéndice 1. La Tabla 6 muestra los resultados de la prueba computacional realizada. Para comparar los resultados obtenidos por cada combinación de los parámetros J_{NR} y J_{FIX} , se utiliza la función objetivo del modelo matemático Ec. 1. Para cada combinación de parámetros testeada se muestra el porcentaje de soluciones factibles encontradas y el sumatorio de los valores de la función objetivo (considerando sólo aquellos ejemplares en los que todas las combinaciones de parámetros encuentran una solución factible). Los mejores resultados se obtienen con los valores siguientes: $J_{NR} = 1$, $J_{FIX} = 1$, $\alpha = 2$ y $T^{MIN} = 2$, para 1RF; y $J_{NR} = 1$, $J_{FIX} = 1$, $\alpha = 5$ y $T^{MIN} = 2$, para 2RF.

			$\alpha = 2$ $T^{MIN} = 2$	$\alpha = 2$ $T^{MIN} = 10$	$\alpha = 5$ $T^{MIN} = 2$	$\alpha = 5$ $T^{MIN} = 10$
1RF	$J_{NR} = 1$ $J_{FIX} = 1$	Porcentaje de soluciones factibles	88%	88%	88%	88%
		Sumatorio función objetivo*	260.32	260.61	261.79	261.79
	$J_{NR} = 2$ $J_{FIX} = 1$	Porcentaje de soluciones factibles	22%	22%	26%	26%
		Sumatorio función objetivo*	265.33	265.41	263.83	264.05
$J_{NR} = 2$ $J_{FIX} = 2$	Porcentaje de soluciones factibles	26%	26%	30%	30%	
	Sumatorio función objetivo*	267.98	267.98	268.13	268.20	
2RF	$J_{NR} = 1$ $J_{FIX} = 1$	Porcentaje de soluciones factibles	72%	72%	74%	74%
		Sumatorio función objetivo*	257.60	257.60	257.38	257.44
	$J_{NR} = 2$ $J_{FIX} = 1$	Porcentaje de soluciones factibles	20%	20%	20%	20%
		Sumatorio función objetivo*	262.59	262.62	262.00	262.07
$J_{NR} = 2$ $J_{FIX} = 2$	Porcentaje de soluciones factibles	22%	22%	24%	22%	
	Sumatorio función objetivo*	265.14	265.17	264.82	264.82	

Tabla 6. Resultados de la experiencia computacional previa de Relajar y Fijar (* considerando sólo los ejemplares en los que todas las combinaciones encuentran una solución factible). Fuente: propia

Adicionalmente también se ha realizado una prueba computacional previa para la

heurística HEUR_PART, con los mismos 50 ejemplares de Otto *et al.* (2013). El objetivo es determinar el límite de tiempo computacional β_1 , en cada iteración, para evaluar la factibilidad de unir estaciones sin aumentar el número de trabajadores (sección 5.3.2.2). La Tabla 7 muestra los resultados obtenidos (utilizando la función objetivo del modelo matemático: Ec. 1) y el tiempo de cálculo, con distintos valores de β_1 . Los mejores resultados se obtienen con $\beta_1 = 5$ segundos.

	$\beta_1 = 5$ segundos	$\beta_1 = 10$ segundos	$\beta_1 = 30$ segundos
Sumatorio función objetivo	1250.64	1250.64	1251.09
Tiempo medio de cálculo (s)	2720	2990	3430

Tabla 7. Resultados de la experiencia computacional previa de HEUR_PART. Fuente: propia

Las instancias utilizadas en los experimentos computacionales de las secciones 5.4.2.1 y 5.4.2.2 también proceden de Otto *et al.* (2013). Se usan un total de 250 instancias: 125 instancias con 50 tareas y 125 con 100 tareas. Cabe recordar que el MILP-DTT1 utiliza una tarea adicional ficticia con un tiempo de proceso cero y sucesora de todas las demás tareas reales; por tanto, en la resolución de las instancias de Otto *et al.* (2013) mediante MILP-DTT1 se consideran 51 y 101 tareas, respectivamente. Con el fin de testear los procedimientos diseñados con una amplia gama de instancias diferentes, las instancias de Otto *et al.* (2013) se han seleccionado de acuerdo con los siguientes criterios:

- *Order strength*: indicador del porcentaje de relaciones de precedencia (arcos) que hay entre las tareas. Se calcula como el cociente entre el número de arcos del ejemplar y el máximo número de arcos posibles. Se fija un order strength igual a 0.6 para más de la mitad de las instancias (63/125 para las instancias de 50 tareas y 72/125 para las de 100 tareas) y de 0.2 para el resto.
- Dificultad: indicador del nivel de dificultad del ejemplar, medida a partir de 10000 ejecuciones en las que se resuelve el problema SALBP de forma aleatoria. La dificultad del ejemplar se define en función del porcentaje de ejecuciones que encuentran la solución óptima:
 - Trivial: si las 10000 ejecuciones encuentran el óptimo.
 - Menos difícil: si el porcentaje de ejecuciones que encuentran el óptimo es igual o superior al 50%.
 - Difícil: si el porcentaje de ejecuciones que encuentran el óptimo es superior o igual al 5% e inferior al 50%.
 - Muy difícil: si el óptimo se encuentra en menos del 5% de las ejecuciones, sin llegar al 0.5%.
 - Extremadamente difícil: si el óptimo se encuentra en el 0.5% de las ejecuciones como máximo.

- Abierto: aquellos ejemplares en los que no se puede determinar su dificultad puesto que se desconoce la solución óptima.

Para ambos conjuntos de 50 y 100 tareas, el 12% de las instancias se seleccionan muy difíciles (15/125), el 68% extremadamente difíciles (85/125) y el 20% abiertas (25/125).

Debido a que las instancias de Otto *et al.* (2013) están diseñadas para SALBP, los ejemplares se han adaptado para incluir tiempos de tarea que dependan del número de trabajadores en la estación. Estos tiempos de tarea se han calculado, siguiendo el planteamiento de Sepahi y Jalali Naini (2014), incrementando los tiempos de proceso en una unidad por cada trabajador adicional en la estación ($P_{i,o+1} = P_{io} + 1$). Además, se establece un máximo de 4 trabajadores en la misma estación ($UB^0 = 4$).

5.4.2. Análisis de los resultados

5.4.2.1. Comparación de los procedimientos MILP-DTT1, 1RF y 2RF, y HEUR_PART

La Tabla 8 presenta una comparación de los resultados obtenidos, entre el modelo matemático MILP-DTT1 (sección 5.1), los procedimientos 1RF y 2RF (sección 5.2) y HEUR_PART (sección 5.3). La mejor solución es la que presenta el menor número de trabajadores y, en caso de empate, el menor número de estaciones. Dado que el modelo matemático no siempre encuentra una solución factible y que los procedimientos SALBP-1 son rápidos, si la solución encontrada por el MILP-DTT1 es peor (o no se encuentra solución) que la solución obtenida por el procedimiento SALBP-1 (sección 5.3.1), se utiliza la solución SALBP-1 a efectos de comparación; la misma operativa se aplica con los dos procedimientos de Relajar y Fijar (1RF y 2RF). La Tabla 8 muestra el porcentaje de ejemplares en los que cada procedimiento alcanza (y no alcanza) la mejor solución encontrada por cualquier procedimiento, el número medio de trabajadores y estaciones obtenido, la diferencia (*gap*) media respecto a la cota inferior y respecto a la mejor solución de cada ejemplar, y la media del tiempo de cálculo. La cota inferior de cada instancia se calcula utilizando la Ec. 37 para el número de trabajadores (LBW) y la Ec. 38 para el número de estaciones (LBJ). En ambos casos, el *gap* de una instancia se calcula utilizando la Ec. 39, siendo la Ec. 1 la utilizada para calcular el valor de la función objetivo. Los resultados obtenidos por cada procedimiento en cada una de las 250 instancias de Otto *et al.* (2013) se muestran en el Apéndice 2.

$$LBW = \left\lceil \frac{\sum_{i=1}^N P_{i,1}}{CT} \right\rceil \quad (\text{Ec. 37})$$

$$LBJ = \left\lceil \frac{LBW}{UB^0} \right\rceil \quad (\text{Ec. 38})$$

$$\text{Gap} = \frac{\text{Valor f. objetivo}_{\text{solución evaluada}} - \text{Valor f. objetivo}_{\text{cota inferior/ mejor solución}}}{\text{Valor f. objetivo}_{\text{cota inferior/ mejor solución}}} \quad (\text{Ec. 39})$$

		MILP-DTT1	1RF	2RF	HEUR_PART
50 tareas	Mejor solución alcanzada	4.8%	0%	0%	99.2%
	Mejor solución no alcanzada	95.2%	100%	100%	0.8%
	Número medio de trabajadores	20.65	20.65	20.65	20.42
	Número medio de estaciones	19.74	20.65	20.63	8.42
	Gap medio con la cota inferior	8.13%	8.29%	8.28%	5.13%
	Gap medio con la mejor solución	3.04%	3.72%	3.71%	0.01%
	Tiempo medio de cálculo (s)	3480	1680	1860	2100
100 tareas	Mejor solución alcanzada	0%	0%	0%	100%
	Mejor solución no alcanzada	100%	100%	100%	0%
	Número medio de trabajadores	29.91	29.91	29.91	29.77
	Número medio de estaciones	29.91	29.90	29.85	11.60
	Gap medio con la cota inferior	6.04%	6.04%	6.01%	2.66%
	Gap medio con la mejor solución	2.80%	2.80%	2.77%	0%
	Tiempo medio de cálculo (s)	3600	1920	2280	3120
Global	Mejor solución alcanzada	2.4%	0%	0%	99.6%
	Mejor solución no alcanzada	97.6%	100%	100%	0.4%
	Número medio de trabajadores	25.28	25.28	25.28	25.09
	Número medio de estaciones	24.82	25.27	25.24	10.01
	Gap medio con la cota inferior	7.08%	7.16%	7.14%	3.89%
	Gap medio con la mejor solución	2.92%	3.26%	3.24%	0.004%
	Tiempo medio de cálculo (s)	3540	1800	2070	2610

Tabla 8. Comparación de los resultados obtenidos por MILP-DTT1, 1RF, 2RF y HEUR_PART. Fuente: propia

Analizando los resultados se concluye que HEUR_PART obtiene el mejor resultado en el 99.2% de las instancias de 50 tareas y en todas las instancias de 100 tareas. Los valores del número medio de trabajadores y estaciones de la Tabla 8 muestran que el número medio de trabajadores es bastante similar en todos los procedimientos. Esto es debido a que si la solución obtenida por el MILP-DTT1 o los dos procedimientos Relajar y Fijar (1RF y 2RF) es infactible o peor que la solución SALBP-1 (sección 5.3.1), que también determina el número inicial de trabajadores en HEUR_PART (como se explica en la sección 5.3), se toma esta solución SALBP-1. En particular, el uso de la solución SALBP-1 inicial sucede en 232 ejemplares para el MILP-DTT1, 220 ejemplares para 1RF y 221 ejemplares para 2RF. Así pues, la diferencia entre los procedimientos propuestos se acentúa en el número de estaciones, donde los resultados de HEUR_PART son significativamente inferiores (11.32 estaciones menos en comparación con el segundo mejor procedimiento propuesto para instancias de 50 tareas; 18.25 estaciones menos para instancias de 100 tareas).

Con relación al *gap* medio con la mejor solución de cada instancia y con la cota inferior, HEUR_PART presenta mejores resultados que los otros tres procedimientos propuestos. La diferencia entre los *gaps* de HEUR_PART y los *gaps* de los otros procedimientos es baja, debido a que el número de trabajadores (que es el objetivo principal) es muy similar, como se ha explicado anteriormente. Por último, en términos de tiempo de cálculo, la primera propuesta del procedimiento de Relajar y Fijar (1RF) presenta un tiempo medio de ejecución menor que HEUR_PART, pero estas diferencias (810 segundos más de tiempo de cálculo medio) no suelen ser significativas en la resolución práctica de un ALBP.

5.4.2.2. Comparación de HEUR_PART según las distintas reglas de prioridad que incorpora

Si se analizan los resultados obtenidos con HEUR_PART para cada solución inicial del SALBP-1 por separado, es posible identificar las reglas de prioridad con las que se obtienen los mejores resultados. La Tabla 9 muestra el número medio de trabajadores y de estaciones obtenidos con HEUR_PART en las 250 instancias de Otto *et al.* (2013) mencionadas en el apartado 5.3.2.1, con cada una de las 12 reglas de prioridad con las que se inicializa la heurística.

	HEUR_PART	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
Número medio de trabajadores	25.09	25.62	25.41	25.74	25.87	26.01	25.72	25.83	25.93	25.83	25.74	25.28	25.81
Número medio de estaciones	10.01	10.78	12.02	10.69	10.30	9.79	10.45	11.58	9.50	11.84	10.58	11.73	10.83

Tabla 9. Resultados obtenidos en HEUR_PART con las doce reglas de prioridad utilizadas. Fuente: propia

El rendimiento de las 12 reglas de prioridad incluidas en HEUR_PART puede evaluarse de dos formas: considerando los resultados de cada regla de prioridad de forma individual (y en ese caso R11 es la que obtiene un mejor valor medio del número de trabajadores) o considerando la combinación (cóctel) de los resultados de las distintas reglas de prioridad. El resultado de un cóctel de reglas es el mejor de los resultados individuales obtenidos con cada una de ellas.

La Tabla 10 clasifica el rendimiento de las reglas de prioridad utilizadas en HEUR_PART tanto de forma individual como combinada. Por ejemplo, la regla R7 es la octava regla con mejor rendimiento singularmente (es decir, la quinta peor) pero, si se consideran los resultados combinados, resulta que R7 es la cuarta mejor. En otras palabras, la combinación de los resultados de cuatro reglas de prioridad que obtiene el mejor valor medio del número de trabajadores es aquella compuesta por R11, R1, R2 y R7. Las reglas R11, R2 y R1 son las que presentan un mejor rendimiento considerando tanto los resultados singulares como los combinados.

Mejores reglas considerando resultados individuales		Mejor regla											Peor regla	
		R11	R2	R1	R6	R10	R3	R12	R7	R9	R4	R8	R5	
Considerando resultados combinados	Mejores reglas	R11	+R1	+R2	+R7	+R10	+R6	+R5	+R12	+R8	+R4	+R9	+R3	
	Número medio de trabajadores	25.28	25.18	25.15	25.14	25.12	25.12	25.10	25.10	25.10	25.09	25.09	25.09	
	Número medio de estaciones	11.73	10.88	10.75	10.46	10.36	10.22	10.25	10.11	10.04	10.04	10.02	10.01	

Tabla 10. Clasificación de las reglas de prioridad en función de su rendimiento para resolver HEUR_PART.

Fuente: propia

5.5. Variantes de HEUR_PART

En esta sección se propone un conjunto de variantes de HEUR_PART cuyo objetivo es mejorar los resultados obtenidos: HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE. En la sección 5.5.1 se describen los nuevos procedimientos diseñados; y en la sección 5.5.2 se detalla la experiencia computacional que los compara con HEUR_PART.

5.5.1. Descripción de las variantes de HEUR_PART

Se han diseñado cuatro nuevos procedimientos de resolución, variantes de la heurística HEUR_PART, que son introducidos a continuación:

- **HEUR_PART_SGL:** la variación respecto a HEUR_PART se encuentra en el Paso 2 (sección 5.3.2.2), y consiste en resolver iterativamente el modelo matemático para encontrar la solución con el menor número de trabajadores sólo en U-estaciones candidatas en las que haya al menos una estación con una única tarea. Para las U-estaciones candidatas restantes se ejecuta el modelo matemático una sola vez con el número de trabajadores igual a la suma del número de trabajadores de cada estación por separado. Esta variante es propuesta como resultado de analizar experimentos computacionales previos realizados con 200 soluciones que reducían el número de trabajadores en el Paso 2: se observó que en 198 de esas 200 soluciones, dicha reducción se producía en U-estaciones que contenían como mínimo una estación inicial con una única tarea.
- **HEUR_PART_OTB:** la variación respecto a HEUR_PART se encuentra en el Paso 1 (sección 5.3.1), y consiste en seleccionar todas las soluciones iniciales de SALBP-1 con el menor número de trabajadores. De esta forma se aplica el Paso 2 (sección 5.3.2) y el Paso 3 (sección 5.3.3) sólo a las soluciones seleccionadas en el Paso 1.
- **HEUR_PART_OTB_SGL:** consiste en combinar HEUR_PART_OTB y

HEUR_PART_SGL.

- **HEUR_PART_SE:** se trata de la variante HEUR_PART_OTB modificada de forma que, en el Paso 2 (sección 5.3.2.2), no hay posibilidad de reducir el número de trabajadores. En otras palabras, el modelo matemático se resuelve una sola vez para cada U-estación con el número de trabajadores igual a la suma del número de trabajadores de cada estación por separado.

Las diferencias entre la heurística HEUR_PART y las variantes propuestas se resumen en la Tabla 11.

	Paso 1: ¿número de soluciones iniciales de SALBP-1 consideradas?	Paso 2: ¿reducción del número de trabajadores?
HEUR_PART	Todas	Sí
HEUR_PART_SGL	Todas	Sólo en U-estaciones con al menos una estación inicial con una única tarea
HEUR_PART_OTB	Sólo las soluciones con el menor número de trabajadores	Sí
HEUR_PART_OTB_SGL	Sólo las soluciones con el menor número de trabajadores	Sólo en U-estaciones con al menos una estación inicial con una única tarea
HEUR_PART_SE	Sólo las soluciones con el menor número de trabajadores	No

Tabla 11. Diferencias entre HEUR_PART y sus variantes. Fuente: propia

5.5.2. Experiencia computacional para comparar HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE

En la Tabla 12 se muestra una comparación entre HEUR_PART y sus variantes: HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE. Se utilizan los mismos 250 ejemplares de Otto *et al.* (2013) de las secciones 5.4.2.1 y 5.4.2.2, con las mismas características (para más detalle, ver la sección 5.4.1). Igual que en HEUR_PART, la experiencia computacional se realiza considerando que el parámetro β_1 es igual a 5 segundos. La estructura de la Tabla 12 es la misma que la de la Tabla 8. Los resultados obtenidos por cada variante de HEUR_PART en las 250 instancias de Otto *et al.* (2013) se muestran en el Apéndice 2.

La variante HEUR_PART_SGL obtiene los mismos resultados que HEUR_PART con un tiempo medio de cálculo inferior (1330 segundos menos para instancias de 50 tareas y

1945 segundos menos para instancias de 100 tareas). En relación al hecho de evaluar la posible reducción del número de trabajadores en el Paso 2 de HEUR_PART sólo en U-estaciones en las que haya al menos una estación con una única tarea es una simplificación que no modifica los resultados aunque reduce el tiempo medio de cálculo (HEUR_PART comparado con HEUR_PART_SGL y HEUR_PART_OTB comparado con HEUR_PART_OTB_SGL). Por otro lado, considerar sólo las soluciones SALBP-1 iniciales con el menor número de trabajadores en vez de considerarlas todas obtiene peores resultados (HEUR_PART_OTB comparado con HEUR_PART y HEUR_PART_OTB_SGL comparado con HEUR_PART_SGL).

		HEUR_PART	HEUR_PART_SGL	HEUR_PART_OTB	HEUR_PART_OTB_SGL	HEUR_PART_SE
50 tareas	Mejor solución alcanzada	99.2%	99.2%	81.6%	81.6%	62.4%
	Mejor solución no alcanzada	0.8%	0.8%	18.4%	18.4%	37.6%
	Número medio de trabajadores	20.42	20.42	20.44	20.44	20.65
	Número medio de estaciones	8.42	8.42	8.70	8.70	8.36
	Gap medio con la cota inferior	5.13%	5.13%	5.25%	5.25%	6.64%
	Gap medio con la mejor solución	0.01%	0.01%	0.15%	0.15%	1.11%
	Tiempo medio de cálculo (s)	2100	770	910	530	420
100 tareas	Mejor solución alcanzada	100%	100%	72%	72%	66.4%
	Mejor solución no alcanzada	0%	0%	28%	28%	33.6%
	Número medio de trabajadores	29.77	29.77	29.83	29.83	29.91
	Número medio de estaciones	11.60	11.60	12.31	12.31	12.07
	Gap medio con la cota inferior	2.66%	2.66%	2.82%	2.82%	2.95%
	Gap medio con la mejor solución	0%	0%	0.17%	0.17%	0.38%
	Tiempo medio de cálculo (s)	3120	1175	1430	870	720
Global	Mejor solución alcanzada	99.6%	99.6%	76.8%	76.8%	64.4%
	Mejor solución no alcanzada	0.4%	0.4%	23.2%	23.2%	35.6%
	Número medio de trabajadores	25.09	25.09	25.13	25.13	25.28
	Número medio de estaciones	10.01	10.01	10.50	10.50	10.21
	Gap medio con la cota inferior	3.89%	3.89%	4.03%	4.03%	4.79%
	Gap medio con la mejor solución	0.004%	0.004%	0.16%	0.16%	0.74%
	Tiempo medio de cálculo (s)	2610	2610	1170	700	570

Tabla 12. Tabla comparativa entre HEUR_PART y sus variantes. Fuente: propia

5.6. Empirically Adjusted Greedy Heuristics (EAGH)

En esta sección se aplica el procedimiento Empirically Adjusted Greedy Heuristics (EAGH) a HEUR_PART. EAGH fue presentado por primera vez en Corominas (2005) y ha sido escasamente utilizado en la literatura: para abordar el problema general de equilibrado de líneas de montaje con *setups* (Martino y Pastor, 2010), el problema de *flow-shop* (Corominas y Pastor, 2011) y para mejorar el algoritmo paramétrico de Clarke y Wright

(1964) (Corominas *et al.*, 2010; Corominas *et al.*, 2014).

Además, en este apartado se presenta un nuevo procedimiento para el diseño de heurísticas *greedy*, que se basa en el uso de EAGH combinado con el concepto de cóctel de heurísticas. Este nuevo procedimiento se denomina "EAGH-CKTL" y puede considerarse como una variante del procedimiento EAGH. El objetivo de EAGH-CKTL es diseñar heurísticas *greedy* mediante EAGH que presenten un buen rendimiento, y esta es la novedad, como parte de un cóctel de heurísticas. El concepto de cóctel de heurísticas para la resolución de problemas difíciles se describe en Corominas *et al.* (2012). De acuerdo con la literatura revisada, EAGH-CKTL es el primer procedimiento que aplica EAGH aprovechando el concepto de cóctel de heurísticas.

En las siguientes subsecciones, EAGH y EAGH-CKTL se utilizan para diseñar nuevas reglas de prioridad para resolver el SMALBP-DTT1 mediante HEUR_PART_1 (recuérdese que el término HEUR_PART_1 se utiliza para hacer referencia a la heurística que, dada una única regla de prioridad, encuentra una solución de SMALBP-DTT1 siguiendo los pasos de HEUR_PART). En concreto, el objetivo es doble: en primer lugar, diseñar tres nuevas reglas de prioridad mediante EAGH que muestren un buen rendimiento por sí solas; en segundo lugar, y aplicando el nuevo procedimiento EAGH-CKTL, generar tres reglas de prioridad adicionales que presenten un buen rendimiento en un cóctel de heurísticas.

La sección 5.6.1 introduce la aplicación de EAGH a HEUR_PART_1 y presenta las tres primeras reglas de prioridad que se han diseñado. La sección 5.6.2 expone el procedimiento EAGH-CKTL y muestra el resultado de aplicarlo para construir otras tres reglas de prioridad que se utilizarán en HEUR_PART. La sección 5.6.3 evalúa la eficacia de las distintas reglas de prioridad propuestas mediante una experiencia computacional.

5.6.1. Diseño de reglas de prioridad aplicando EAGH a HEUR_PART_1

En esta sección se definen nuevas reglas de prioridad a utilizar en HEUR_PART_1, para resolver el SMALBP-DTT1. En particular, y tal y como se hace en el procedimiento HEUR_PART_1, las nuevas reglas de prioridad se aplican en la generación de la solución inicial SALBP-1 (mediante un procedimiento constructivo *greedy* orientado a estaciones). El diseño de las nuevas reglas de prioridad se realiza utilizando el procedimiento Empirically Adjusted Greedy Heuristics (EAGH), presentado en Corominas (2005).

La principal característica de una heurística *greedy* es que en cada iteración se toma una decisión irreversible en función de un indicador asociado a cada posible decisión. En el caso de HEUR_PART_1, la decisión irreversible consiste en seleccionar la tarea que, en cada iteración, debe asignarse a una estación para generar una solución inicial SALBP-1 (que luego se utilizará en los pasos posteriores de HEUR_PART_1), y los valores de los

indicadores para tomar dicha decisión se determinan mediante una regla de prioridad. Para esta decisión, se pueden considerar varios indicadores diferentes (por ejemplo, las reglas de prioridad presentadas en la Tabla 4). Cuando se consideran dos o más indicadores, pueden generarse infinitos indicadores mixtos combinándolos y, por tanto, puede obtenerse un conjunto infinito H de heurísticas. El conjunto de heurísticas H puede definirse mediante una función h , que depende del conjunto de atributos de la decisión (es decir, en el problema aquí planteado del conjunto de reglas de prioridad R_j ($j = 1, \dots, 12$) de la Tabla 4) y de varios parámetros reales ($\Pi = \{\pi_1, \pi_2, \dots\}$).

El procedimiento EAGH busca la mejor función heurística $h \in H$. La definición de “mejor” depende de una función objetivo f a optimizar y de un conjunto de instancias de entrenamiento T . Sea $X_{k,\Pi}$ la solución de la instancia $k \in T$ obtenida utilizando la función $h(R_j, \Pi)$, entonces puede definirse la función φ como Ec. 40.

$$\varphi(\Pi) = \sum_{k \in T} f(X_{k,\Pi}) \quad (\text{Ec. 40})$$

Encontrar la mejor función heurística $h(R_j, \Pi)$ equivale a encontrar los valores de los parámetros Π que minimizan φ . No se espera que la función φ tenga ninguna propiedad especial. Por lo tanto, para calibrar los valores de los parámetros Π de la función φ , sólo se puede aplicar un algoritmo de optimización directa (es decir, un algoritmo que sólo tenga en cuenta los valores de la función φ). EAGH utiliza el algoritmo de Nelder y Mead (N&M), también conocido como algoritmo de los poliedros flexibles (Nelder y Mead, 1965).

El algoritmo N&M se basa en $n + 1$ puntos ($n = |\Pi|$) que son los vértices de un poliedro regular (también llamado hipertetraedro) en el espacio n -dimensional. Las coordenadas de los vértices corresponden a los valores de los parámetros Π . Cada vértice se evalúa mediante la función φ . El algoritmo N&M parte de un hipertetraedro inicial. Para construirlo, hay que definir un vértice inicial (V_0) y la longitud de las aristas del hipertetraedro inicial (δ).

En cada iteración del algoritmo N&M se generan uno o varios puntos y se calcula el valor de la función φ en cada uno de ellos. Según la comparación entre los valores de φ obtenidos para los nuevos puntos y los valores de φ de los vértices del hipertetraedro actual, como se explica a continuación, sus vértices se desplazan por el espacio n -dimensional.

Sean $n + 1$ vértices del hipertetraedro (V_1, \dots, V_{n+1}); antes de comenzar una iteración, los vértices del hipertetraedro se etiquetan y ordenan de forma que $\varphi(V_1) \leq \dots \leq \varphi(V_{n+1})$. En caso de empate, se da prioridad al vértice más antiguo. Una iteración de N&M comienza con la creación de V_r , el punto de reflexión de V_{n+1} : $V_r = \bar{V} + \rho \cdot (\bar{V} - V_{n+1})$, donde \bar{V} es el centroide de los n mejores puntos ($\bar{V} = \sum_{p=1}^n V_p / n$) y $\rho > 0$ es un parámetro. En función de

$\varphi(V_r)$, el hipertetraedro puede realizar cuatro posibles movimientos:

1. Expansión. Si $\varphi(V_r) < \varphi(V_1)$, se calcula $V_e = \bar{V} + \chi \cdot (V_r - \bar{V})$, donde $\chi > 1$ es un parámetro. Si $\varphi(V_e) < \varphi(V_r)$, se sustituye V_{n+1} por V_e ; en caso contrario, se sustituye V_{n+1} por V_r .
2. Reflexión. Si $\varphi(V_1) \leq \varphi(V_r) < \varphi(V_n)$, sustituir V_{n+1} por V_r .
3. Contracción exterior. Si $\varphi(V_n) \leq \varphi(V_r) < \varphi(V_{n+1})$, calcular $V_c = \bar{V} + \gamma \cdot (V_r - \bar{V})$, donde $0 < \gamma < 1$ es un parámetro. Si $\varphi(V_c) \leq \varphi(V_r)$, sustituye V_{n+1} por V_c ; en caso contrario, contrae todos los vértices excepto V_1 : $V_p = V_1 + \sigma \cdot (V_p - V_1)$ para $p = 2, \dots, n + 1$, donde $0 < \sigma < 1$ es un parámetro.
4. Contracción interior. Si $\varphi(V_{n+1}) \leq \varphi(V_r)$ entonces calcula $V_{cc} = \bar{V} + \gamma \cdot (V_{n+1} - \bar{V})$. Si $\varphi(V_{cc}) < \varphi(V_{n+1})$, sustituye V_{n+1} por V_{cc} ; en caso contrario, encoge todos los vértices excepto V_1 como en 3).

En cuanto al valor de los parámetros $\rho, \chi, \gamma, \sigma$, como se indica en Lagarias *et al.* (1998), las opciones universales utilizadas en los algoritmos N&M estándar son $\rho = 1, \chi = 2, \gamma = 0.5, \sigma = 0.5$.

El N&M repite el proceso iterativamente hasta que se alcanza un óptimo local. Se considera que se alcanza un óptimo local cuando el hipertetraedro es suficientemente pequeño. Para más detalles sobre el algoritmo N&M, véase Nelder y Mead (1965) y Lagarias *et al.* (1998).

El procedimiento EAGH consta de los siguientes pasos:

Paso 1. Diseño de la función $h(R_j, \Pi)$.

Paso 2. Calibración de los parámetros Π de la función $h(R_j, \Pi)$ mediante el algoritmo directo N&M (es decir, optimizando $\varphi(\Pi)$) con el conjunto de instancias de entrenamiento T . Para iniciar el algoritmo N&M, es necesario determinar el vértice inicial (V_0) y la longitud de las aristas del hipertetraedro inicial (δ).

Paso 3. Validación de los resultados aplicando la función $h(R_j, \Pi)$ obtenida en la etapa anterior a un conjunto dado de instancias de validación V . El conjunto V no debe compartir ninguna instancia con el conjunto T . La calidad de las soluciones obtenidas se evalúa con la función f .

En el diseño de nuevas reglas de prioridad mediante EAGH para aplicar en HEUR_PART_1, los elementos de EAGH se definen del siguiente modo:

- La función h se utiliza como regla de prioridad para especificar el orden de las tareas (en función del valor decreciente de h) para el procedimiento constructivo

greedy que encuentra la solución inicial SALBP-1 de HEUR_PART_1. La función h será una combinación de algunas de las reglas de prioridad R_j ($j = 1, \dots, 12$) presentadas en la Tabla 4. Las funciones $h(R_j, \Pi)$ propuestas se explican posteriormente.

- La función f a minimizar se calcula con la Ec. 41. El peso $\frac{1}{UB^M+1}$ asegura que el número de estaciones sea el criterio de desempate.

$$f = N^{\circ}Operarios + \frac{N^{\circ}Estaciones}{UB^M + 1} \quad (\text{Ec. 41})$$

- El conjunto de entrenamiento T está compuesto por 50 instancias seleccionadas aleatoriamente de Otto *et al.* (2013) (las instancias utilizadas se detallan en el Apéndice 1) y 250 instancias (aquellas utilizadas en las experiencias computacionales previas, véase sección 5.4.1) constituyen el conjunto de validación V .

En cuanto al Paso 1 de EAGH, se proponen tres funciones h diferentes y, por lo tanto, se diseñan tres nuevas reglas de prioridad. Las funciones h se han construido teniendo en cuenta el rendimiento de cada una de las 12 reglas de prioridad (ver Tabla 9). La regla R8 (menor índice de la tarea) se ha excluido porque no se basa en los atributos de las tareas, sino sólo en cómo se indexan. Además, a partir de las pruebas computacionales preliminares durante el diseño de las funciones h , la regla R4 (mayor número de tareas sucesoras inmediatas) mostró un rendimiento muy deficiente y, por lo tanto, tampoco se ha incluido en las funciones h definitivas propuestas.

La función h_1 (presentada en la Ec. 42) combina R11 (multiplicada por π_1), R1 (multiplicada por π_2 y normalizada dividiéndola por el tiempo de ciclo), R10 (multiplicada por π_3), y las cotas inferior (E_i) y superior (L_i) de la estación a la que podría asignarse una tarea, pudiendo representar R7 ($L_i - E_i$) pero también cualquier otra combinación que utilice R6 (L_i ; multiplicada por π_4) y/o R5 (E_i ; multiplicada por π_5). Nótese que restar una regla de prioridad significa que dicha regla prioriza valores menores (las tareas se ordenan en base al valor decreciente de h_1).

$$h_1 = \pi_1 \cdot \frac{P_{i,1}}{L_i} + \pi_2 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} - \pi_3 \cdot \frac{L_i}{|ASC_i| + 1} - \pi_4 \cdot L_i + \pi_5 \cdot E_i \quad (\text{Ec. 42})$$

La función h_2 (presentada en la Ec. 43) combina las reglas de prioridad R1 (multiplicada por π_1 y normalizada), R2 (multiplicada por π_2 y normalizada), R3 (multiplicada por π_3) y una variación de R5 que no considera el redondeo al entero superior más próximo, sino que trabaja con el número real (multiplicado por π_4).

$$h_2 = \pi_1 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} + \pi_2 \cdot \frac{P_{i,1}}{CT} + \pi_3 \cdot |ASC_i| - \pi_4 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{CT} \quad (\text{Ec. 43})$$

Finalmente, h_3 (formulada en la Ec. 44) combina las reglas de prioridad R11 (multiplicada por π_1), R10 (multiplicada por π_2), R12 (multiplicada por π_3) y R9 (multiplicada por π_4).

$$h_3 = \pi_1 \cdot \frac{P_{i,1}}{L_i} - \pi_2 \cdot \frac{L_i}{|ASC_i| + 1} + \pi_3 \cdot \frac{|ASC_i|}{L_i - E_i} + \pi_4 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{|ASC_i| + 1} \quad (\text{Ec. 44})$$

En relación con el Paso 2 del procedimiento EAGH, para cada función h (h_1, h_2, h_3) se evaluaron diferentes hipertetraedros iniciales durante las pruebas computacionales preliminares realizadas con el conjunto de instancias de entrenamiento T . La definición de los hipertetraedros iniciales se realizó de manera iterativa considerando los resultados que se iban obteniendo con los hipertetraedros iniciales previos. En la Tabla 13 se indican las longitudes iniciales de las aristas δ y los vértices iniciales V_0 que fueron testeados. Para cada hipertetraedro inicial se muestran los resultados obtenidos.

h	δ	V_0	f utilizando $h(R_j, \Pi)$
h_1	1	(1,0,0,0)	6511.40
	1	(0,1,0,0)	6537.37
	1	(0,0,1,0)	6575.21
	1	(0,0,0,1)	6581.03
	1	(0,0,0,1)	6576.00
	1	(0,0,0,1,1)	6553.82
	1	(1,0.5,0,0,0)	6536.94
	1	(1,0,0,0.5,0.5)	6540.02
	2	(1,0,0,0,0)	6532.20
	2	(1,0.5,0,0,0)	6534.98
h_2	1	(1,0,0,0)	6472.45
	1	(0,1,0,0)	6453.94
	1	(0,0,1,0)	6510.02
	1	(0,0,0,1)	6498.37
	1	(0.5,1,0,0)	6439.23
	1	(1,0.5,0,0)	6487.05
	1	(0.25,1,0,0)	6411.11
	1	(0.75,1,0,0)	6450.61
	2	(0.25,1,0,0)	6429.69
	2	(0.5,1,0,0)	6448.97
h_3	1	(1,0,0,0)	6481.21
	1	(0,1,0,0)	6470.54
	1	(0,0,1,0)	6491.07
	1	(0,0,0,1)	6503.60
	1	(0.5,1,0,0)	6452.83
	1	(1,1.25,0,0)	6469.58
	2	(0.5,1,0,0)	6472.27
	2	(1,1.25,0,0)	6460.40
	2	(0.25,1,0,0)	6468.03
	2	(0.5,1,0.25,0)	6491.25

Tabla 13. Hipertetraedros iniciales evaluados en la experiencia computacional previa de EAGH. Fuente: propia

Los valores de δ y V_0 que proporcionan los mejores resultados son: para h_1 , $V_0 = (1,0,0,0,0)$ y $\delta = 1$; para h_2 , $V_0 = (0.25,1,0,0)$ y $\delta = 1$; para h_3 , $V_0 = (0.5,1,0,0)$ y $\delta = 1$.

Los resultados finales de aplicar el algoritmo N&M para calibrar los parámetros Π de las funciones h_1, h_2, h_3 se muestran en la Tabla 14 (dichos valores se han redondeado al segundo decimal). Los tiempos de cálculo necesarios para calibrar los parámetros Π fueron los siguientes: h_1 empleó 41.22 horas (y realizó 76 iteraciones), h_2 necesitó 41.06 horas (con 122 iteraciones) y h_3 utilizó 24.03 horas (y 39 iteraciones). Aunque el tiempo necesario para calibrar las funciones h podría considerarse elevado, debe tenerse en cuenta que el calibrado sólo se realiza una vez (Corominas *et al.*, 2010). Se denominan R13, R14 y R15, respectivamente, a las funciones h_1, h_2 y h_3 resultantes.

Reglas de prioridad resultantes mediante el algoritmo N&M en EAGH	
R13	$h_1 = 2.35 \cdot \frac{P_{i,1}}{L_i} + 0.77 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} - 0.15 \cdot \frac{L_i}{ ASC_i + 1} - 0.04 \cdot L_i + 1.07 \cdot E_i$
R14	$h_2 = 1.09 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} + 1.96 \cdot \frac{P_{i,1}}{CT} + 0.15 \cdot ASC_i + 0.06 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{CT}$
R15	$h_3 = 1.32 \cdot \frac{P_{i,1}}{L_i} - 1.61 \cdot \frac{L_i}{ ASC_i + 1} + 0.19 \cdot \frac{ ASC_i }{L_i - E_i} + 0.02 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{ ASC_i + 1}$

Tabla 14. Reglas de prioridad resultantes tras aplicar el algoritmo de Nelder y Mead en EAGH. Fuente: propia

En la sección 5.6.3 se muestran los resultados de resolver las instancias del conjunto de validación V utilizando las nuevas reglas de prioridad R13, R14 y R15 (Paso 3).

5.6.2. Diseño de reglas de prioridad aplicando EAGH con un cóctel de heurísticas (EAGH-CKTL)

5.6.2.1. Procedimiento EAGH-CKTL

En todos los artículos publicados en la literatura que utilizan el procedimiento EAGH (Corominas, 2005; Corominas *et al.*, 2010; Martino y Pastor, 2010; Corominas y Pastor, 2011; Corominas *et al.*, 2014), se han diseñado nuevas heurísticas *greedy* que proporcionan buenos resultados por sí solas. En esta tesis doctoral se introduce un nuevo procedimiento, denominado EAGH-CKTL, que se basa en diseñar heurísticas *greedy* mediante EAGH como parte de un grupo (cóctel) de heurísticas.

La idea de considerar un grupo de heurísticas, también conocido como cóctel de heurísticas, para resolver un problema de optimización combinatoria se introduce en Corominas *et al.* (2012). En la literatura, un problema de optimización se resuelve comúnmente utilizando sólo la heurística con el mejor rendimiento medio. Como se indica en Corominas *et al.* (2012), este enfoque está justificado cuando los algoritmos de

resolución presentan tiempos de cálculo elevados considerando el problema a resolver. Sin embargo, si un problema de optimización combinatoria no requiere altos tiempos de cálculo para su resolución, un enfoque que obtiene mejores resultados es resolver el problema utilizando varias o incluso todas las heurísticas rápidas disponibles (manteniendo siempre la mejor solución encontrada). La razón que subyace al cóctel de heurísticas es que una heurística que puede no tener el mejor rendimiento medio puede producir, de todos modos, la mejor solución en determinadas instancias.

Según la bibliografía revisada, EAGH-CKTL es el primer procedimiento que utiliza EAGH combinado con el concepto de cóctel de heurísticas. Dado un conjunto de heurísticas conocidas ($\theta = \{h'_1, h'_2, \dots\}$), EAGH-CKTL aplica el procedimiento EAGH (descrito en el apartado anterior) diferenciándose en la definición de la función f a optimizar. En EAGH-CKTL la función f considera los resultados obtenidos al resolver el problema de optimización combinatoria con el cóctel de heurísticas, que contiene el conjunto θ de heurísticas conocidas y la nueva heurística *greedy* h a diseñar ($f = \text{opt}(f_h, f_{h'_1}, f_{h'_2}, \dots) \mid \text{opt} \in \{\min, \max\}$ según si se quiere minimizar o maximizar f). Así, en EAGH-CKTL optimizar la función f significa encontrar una nueva heurística *greedy* que mejore los resultados obtenidos al resolver el problema de optimización combinatoria con el cóctel de las heurísticas θ ya existentes. En otras palabras, en EAGH-CKTL, la búsqueda de los valores de los parámetros Π de $h(R_j, \Pi)$ que minimicen φ conduce al diseño de una nueva heurística *greedy* que tenga un buen rendimiento como parte del cóctel de heurísticas.

En la sección 5.6.2.2 se expone la aplicación de EAGH-CKTL en el diseño de nuevas reglas de prioridad a aplicar en HEUR_PART_1 para resolver el SMALBP-DTT1.

5.6.2.2. Diseño de reglas de prioridad aplicando EAGH-CKTL a HEUR_PART

El procedimiento HEUR_PART_1, dada una regla de prioridad, requiere una media de entre 220 y 260 segundos para resolver SMALBP-DTT1, tiempos que no suelen ser significativos para resolver el SMALBP-DTT1 en líneas de montaje de la vida real. Por tanto, EAGH-CKTL puede aplicarse para generar nuevas reglas de prioridad. En este caso, la eficacia de las nuevas reglas de prioridad se mide con la solución alcanzada resolviendo HEUR_PART_1 con un cóctel de reglas de prioridad.

Para diseñar nuevas reglas de prioridad mediante el procedimiento EAGH-CKTL, deben definirse los siguientes elementos de EAGH (como se explica en la sección 5.6.1): la función h que se utilizará como regla de prioridad para el procedimiento *greedy* que encuentra la solución inicial SALBP-1 de HEUR_PART_1, la función objetivo f que se optimizará, el conjunto de entrenamiento T , el conjunto de validación V y el hipertetraedro inicial para el algoritmo N&M (es decir, el vértice inicial V_0 y la longitud inicial de las aristas δ). Además, para aplicar EAGH-CKTL, debe especificarse el cóctel: el conjunto de θ

heurísticas conocidas.

El conjunto θ está formado por las 12 reglas de prioridad utilizadas en HEUR_PART. Por tanto, el cóctel de heurísticas contiene esas 12 reglas de prioridad más la regla de prioridad h a diseñar. De esta forma, la función f a minimizar se define como la Ec. 45. Dicha función tiene en cuenta los resultados de resolver HEUR_PART_1 no sólo con la nueva regla de prioridad actual (mostrada en Ec. 45 como "new_R") sino también con las otras 12 reglas de prioridad (presentes en la Tabla 4; R_j ($j = 1, \dots, 12$)). Minimizar la función f significa encontrar una nueva regla de prioridad cuya solución HEUR_PART_1 mejore los resultados obtenidos resolviendo HEUR_PART_1 con el cóctel de las otras 12 reglas de prioridad.

$$f = \min \left(\begin{array}{l} \min_{j=1, \dots, 12} \left(N^{\circ}Operarios_{R_j} + \frac{N^{\circ}Estaciones_{R_j}}{UB^M + 1} \right); \\ N^{\circ}Operarios_{new_R} + \frac{N^{\circ}Estaciones_{new_R}}{UB^M + 1} \end{array} \right); \quad (\text{Ec. 45})$$

En relación con el conjunto de entrenamiento T y el conjunto de validación V , se utilizan los mismos de la sección 5.6.1: un conjunto de entrenamiento T de 50 instancias, detallado en el Apéndice 1, y un conjunto de validación V de 250 instancias, descritas en la sección 5.4.1. En cuanto a la función h , en este caso se utilizan las mismas tres funciones h presentadas en Ec. 42-44 y, por tanto, se diseñan tres nuevas reglas de prioridad.

Nótese que en esta aplicación del procedimiento EAGH-CKTL, el objetivo es calibrar los parámetros Π de la función $h(R_j, \Pi)$ de forma que se minimice la solución HEUR_PART_1 obtenida con las 13 reglas de prioridad (las 12 reglas de prioridad ya conocidas más la nueva que está siendo diseñada). La nueva regla de prioridad puede contribuir a minimizar φ especialmente en aquellos casos en los que las 12 reglas de prioridad funcionen peor.

Para el calibrado de las funciones h de Ec. 42-44, se evaluaron diferentes hipertetraedros iniciales. En la Tabla 15 se muestran las longitudes iniciales de las aristas δ y los vértices iniciales V_0 que fueron testeados, con la misma estructura que la Tabla 13.

h	δ	V_0	f utilizando $h(R_j, \Pi)$
h_1	1	(1,0,0,0)	6367.23
	1	(0,1,0,0)	6365.93
	1	(0,0,1,0)	6367.08
	1	(0,0,0,1)	6366.91
	1	(1,1,1,1)	6339.62
	2	(1,1,1,1)	6347.20
	1	(1,1,0.5,0.5)	6355.27
	1	(0.5,0.5,1,1)	6362.33
	1	(1.25,1,0.5,0.5)	6359.95
	2	(1,1,0.5,0.5)	6354.09
h_2	1	(1,0,0,0)	6366.89

h	δ	V_0	f utilizando $h(R_j, \Pi)$
	1	(0,1,0,0)	6366.70
	1	(0,0,1,0)	6367.02
	1	(0,0,0,1)	6365.81
	1	(1,1,1,1)	6364.04
	2	(1,1,1,1)	6363.96
	2	(1,0.5,0.5,1.25)	6363.70
	2	(1,0.75,0.25,1.25)	6362.81
	1	(1,0.75,0.25,1.25)	6362.28
	1	(1,0.25,0.75,1.25)	6362.63
h_3	1	(1,0,0,0)	6365.82
	1	(0,1,0,0)	6365.25
	1	(0,0,1,0)	6366.76
	1	(0,0,0,1)	6365.54
	1	(1,1,1,1)	6352.06
	2	(1,1,1,1)	6359.23
	1	(1.25,1,0.5,0.5)	6358.99
	1	(1,1.25,0.5,0.5)	6357.35
	2	(1,1.25,0.5,0.5)	6356.88
2	(1,1.25,0.25,0.75)	6354.90	

Tabla 15. Hipertetraedros iniciales evaluados en la experiencia computacional previa de EAGH-CKTL.

Fuente: propia

Los hipertetraedros iniciales que dan lugar a funciones h con un mejor rendimiento son: para h_1 , $V_0 = (1,1,1,1,1)$ y $\delta = 1$; para h_2 , $V_0 = (1,0.75,0.25,1.25)$ y $\delta = 1$; para h_3 , $V_0 = (1,1,1,1)$ y $\delta = 1$.

Los resultados finales de aplicar el algoritmo N&M para calibrar los parámetros Π de las funciones h_1 , h_2 y h_3 mediante EAGH-CKTL se presentan en la Tabla 16. Los tiempos de cálculo necesarios para calibrar los parámetros Π fueron los siguientes: h_1 empleó 22.96 horas (y realizó 41 iteraciones), h_2 necesitó 40.75 horas (con 79 iteraciones) y h_3 utilizó 24.27 horas (y 45 iteraciones). Se denominan R16, R17 y R18 a las funciones h_1 , h_2 y h_3 resultantes, respectivamente. En la sección 5.6.3 se muestran los resultados de resolver las instancias del conjunto de validación V utilizando las nuevas reglas de prioridad R16, R17 y R18 (Paso 3).

Reglas de prioridad resultantes mediante el algoritmo N&M en EAGH-CKTL	
R16	$h_1 = 0.48 \cdot \frac{P_{i,1}}{L_i} + 1.39 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} - 1.66 \cdot \frac{L_i}{ ASC_i + 1} - 1.76 \cdot L_i + 0.85 \cdot E_i$
R17	$h_2 = 1.63 \cdot \frac{P_{i,1} + \sum_{h \in ASC_i} P_{h,1}}{CT} - 2.75 \cdot \frac{P_{i,1}}{CT} - 3.73 \cdot ASC_i + 0.74 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{CT}$
R18	$h_3 = 0.87 \cdot \frac{P_{i,1}}{L_i} - 0.78 \cdot \frac{L_i}{ ASC_i + 1} + 1.59 \cdot \frac{ ASC_i }{L_i - E_i} + 1.70 \cdot \frac{P_{i,1} + \sum_{h \in APR_i} P_{h,1}}{ ASC_i + 1}$

Tabla 16. Reglas de prioridad resultantes tras aplicar el algoritmo de Nelder y Mead en EAGH-CKTL. Fuente: propia

5.6.3. Experiencia computacional para comparar HEUR_PART, R13-R15 de EAGH y R16-R18 de EAGH-CKTL

Esta sección evalúa las seis nuevas reglas de prioridad diseñadas para resolver el SMALBP-DTT1 mediante HEUR_PART_1: R13, R14 y R15 (sección 5.6.1) y R16, R17 y R18 (sección 5.6.2). La efectividad de las nuevas reglas de prioridad se analiza comparando los resultados obtenidos con dichas reglas al resolver HEUR_PART_1, con los resultados obtenidos con cada una de las 12 reglas de prioridad usadas en HEUR_PART. Además, se comparan las nuevas reglas de prioridad considerándolas como parte de un cóctel de heurísticas formado también por R1-R12.

Las instancias utilizadas para los experimentos computacionales proceden de Otto *et al.* (2013). Se evalúan las mismas 250 instancias, que formaron el conjunto de validación V utilizado en EAGH y EAGH-CKTL (125 instancias con 50 tareas y 125 con 100 tareas). Para más detalle sobre los ejemplares utilizados y las características de la experiencia computacional, revisar la sección 5.4.1.

La Tabla 17 presenta una comparación entre las tres reglas de prioridad utilizadas en HEUR_PART que presentan mejores resultados (R1, R2 y R11; más detalles en la sección 5.4.2.2) y las seis nuevas reglas de prioridad ahora propuestas (R13-R18), sin incluirlas como parte del cóctel. La mejor regla de prioridad es la que encuentra una solución SMALBP-DTT1 con el menor número de trabajadores y, en caso de empate, el menor número de estaciones. La Tabla 17 indica el número medio de trabajadores y estaciones obtenido con cada regla de prioridad y el tiempo medio de cálculo. Los resultados obtenidos para cada regla de la Tabla 17 en las 250 instancias de Otto *et al.* (2013) se muestran en el Apéndice 3.

	Tres reglas con mejores resultados en HEUR_PART			Reglas diseñadas con EAGH considerando la heurística por sí sola			Reglas diseñadas con EAGH-CKTL teniendo en cuenta un cóctel de heurísticas		
	R1	R2	R11	R13	R14	R15	R16	R17	R18
Número medio de trabajadores	25.62	25.41	25.28	25.61	25.23	25.41	25.33	25.80	25.46
Número medio de estaciones	10.78	12.02	11.73	11.22	10.85	10.66	10.68	11.29	11.28
Tiempo medio de cálculo (s)	224	257	236	237	243	253	237	251	220

Tabla 17. Resultados de las nuevas reglas de prioridad en comparación con las mejores reglas de prioridad de HEUR_PART. Fuente: propia

La regla de prioridad R14 presenta el mejor rendimiento (con una media de 25.23 trabajadores y 10.85 estaciones) considerando todas las reglas de prioridad (las 12 reglas de prioridad utilizadas en HEUR_PART y las 6 nuevas propuestas); en particular, R14 mejora los resultados de R11 (con una media de 25.28 trabajadores y 11.73 estaciones),

la regla utilizada en HEUR_PART que había obtenido los mejores resultados. También cabe destacar R16, que muestra un buen rendimiento por sí sola (siendo la tercera mejor entre las 18 reglas de prioridad, con una media de 25.33 trabajadores y 10.68 estaciones) a pesar de haber sido diseñada con el objetivo de tener un buen rendimiento como parte de un cóctel de heurísticas. En general, las nuevas reglas de prioridad presentan buenos resultados al compararlas con las 12 reglas de prioridad utilizadas en HEUR_PART: no sólo las ya mencionada R14 (la mejor de todas) y R16 (3ª), sino también R15 (4ª), R18 (6ª) y R13 (7ª). La regla R17 es la única de las nuevas cuyos resultados no se encuentran entre los siete primeros (ocupa la posición 12ª). En términos del tiempo de cálculo necesario para resolver un ejemplar con las nuevas reglas de prioridad, se necesita una media de entre 220 y 257 segundos para cada regla para encontrar una solución SMALBP-DTT1 utilizando HEUR_PART_1.

A continuación, se evalúa el rendimiento de las reglas de prioridad R13-R18 considerando cada una de ellas como parte de un cóctel de heurísticas compuesto por dicha regla y las 12 reglas de prioridad utilizadas en HEUR_PART. La Tabla 18 muestra el número medio de trabajadores y de estaciones, y el tiempo medio de cálculo obtenido resolviendo HEUR_PART_1 con: las 12 reglas de prioridad R1-R12 (como se hace en la sección 5.4.2.1); y con 13 reglas de prioridad (R1-R12 y cada una de las aquí propuestas). En todos los casos, se considera la mejor solución (es decir, la que obtiene el menor número de trabajadores y, en caso de empate, el menor número de estaciones) entre todas las reglas de prioridad evaluadas.

	HEUR_PART (con R1-R12)	HEUR_PART (con R1-R12)					
		+R13	+R14	+R15	+R16	+R17	+R18
Número medio de trabajadores	25.09	25.06	25.04	25.04	24.98	25.07	25.03
Número medio de estaciones	10.01	9.99	9.94	10.06	10.07	10.04	9.98
Tiempo medio de cálculo (s)	2610	2847	2853	2863	2847	2861	2830

Tabla 18. Resultados obtenidos por HEUR_PART_1 considerando R1-R12 y la adición de las nuevas reglas de prioridad. Fuente: propia

Las seis nuevas reglas de prioridad, cuando se incluyen individualmente en un cóctel formado por R1-R12, mejoran los resultados obtenidos por dicho cóctel. En particular, la regla de prioridad R16 es aquella con la que añadiéndola al cóctel de heurísticas formado por R1-R12 se obtienen los mejores resultados (el número medio de trabajadores disminuye de 25.09 a 24.98).

Además, por un lado, la regla de prioridad R14, que tiene el mejor rendimiento por sí sola (Tabla 17), muestra una mejora menor que R16 cuando se evalúa como parte del cóctel de heurísticas R1-R12. Por otro lado, la regla de prioridad R18, que presenta el tercer peor rendimiento por sí sola entre las seis nuevas reglas de prioridad (Tabla 17), es la segunda

mejor regla si se considera como parte del cóctel de heurísticas R1-R12. Ambos resultados son esperables, ya que R13-R15 se diseñaron mediante EAGH para tener un buen rendimiento sin tener en cuenta los resultados de R1-R12, mientras que R16-R18 se diseñaron mediante EAGH-CKTL para tener un buen rendimiento como parte de un cóctel de heurísticas que contenía R1-R12. Por último, los tiempos de cálculo muestran que resolver HEUR_PART_1 con un cóctel de 13 reglas de prioridad requiere una media de entre 2830 y 2863 segundos.

5.7. Heurística basada en la resolución de problemas de partición combinados (MULTI_HEUR_PART)

En esta sección se presenta MULTI_HEUR_PART, un procedimiento heurístico para el SMALBP-DTT1 (minimizando el número de trabajadores y estaciones como objetivos primario y secundario, respectivamente) basado en la resolución de problemas de partición combinados. La experiencia computacional de MULTI_HEUR_PART se detalla en la sección 5.9.

MULTI_HEUR_PART consiste en resolver uno o varios problemas de partición adicionales, combinando las U-estaciones candidatas encontradas en el Paso 2 de HEUR_PART por distintas reglas de prioridad (a diferencia de HEUR_PART que sólo considera las U-estaciones obtenidas en el Paso 2 por cada regla de prioridad por separado).

Para resolver el problema de partición considerando las U-estaciones candidatas encontradas por distintas reglas de prioridad, se utiliza el siguiente modelo matemático:

Datos

N número de tareas ($i = 1, \dots, N$).

S número de U-estaciones candidatas ($s = 1, \dots, S$).

NW_s número de trabajadores en la U-estación s ($s = 1, \dots, S$).

IS_{si} 1 si en la U-estación s está incluida la tarea i , 0 en caso contrario:

$i = 1, \dots, N; s = 1, \dots, S$.

\overline{INC} grupos de U-estaciones que *no* tienen tareas en común pero que son infactibles por razones de precedencia.

R número de U-estaciones en cada grupo de \overline{INC} ($R = 2, \dots, N$).

$$\overline{INC} = \{(r_1, \dots, r_R) \mid r_1 \in \{1, \dots, S - (R - 1)\}, \dots, r_R \in \{r_1 + (R - 1), \dots, S\}, r_1, \dots, r_R \text{ infactibles}\}.$$

Dos U-estaciones ($R = 2$) son infactibles si en una de las U-estaciones hay al mismo tiempo tareas sucesoras y predecesoras con respecto a las tareas de la otra U-estación. Tres o más U-estaciones ($R = 3, \dots, J$) son infactibles si cada U-estación tiene relaciones de precedencia con sólo otras dos U-estaciones (más concretamente, tiene tareas predecesoras con respecto a las tareas de una U-estación y tareas sucesoras con respecto a las tareas de la otra U-estación) y no hay infactibilidades en ninguna combinación de $2, \dots, R - 1$ U-estaciones.

Variables

$x_s \in \{0,1\}$ 1 si se usa la U-estación candidata s , 0 en caso contrario: $s = 1, \dots, S$.

Función objetivo

La función objetivo minimiza el número total de trabajadores como primer objetivo y el número de U-estaciones seleccionadas como objetivo secundario:

$$[MIN] \sum_{s=1}^S NW_s \cdot x_s + \frac{1}{N+1} \cdot \sum_{s=1}^S x_s \quad (\text{Ec. 46})$$

Restricciones

$$\sum_{s=1}^S IS_{si} \cdot x_s = 1 \quad i = 1, \dots, N \quad (\text{Ec. 47})$$

$$\sum_{s=r_1}^{r_R} x_s \leq R - 1 \quad (r_1, \dots, r_R) \in \overline{INC} \quad (\text{Ec. 48})$$

Las restricciones (47) garantizan que la solución final incluya cada tarea una sola vez. Las restricciones (48) impiden asignar U-estaciones infactibles por razones de precedencia.

MULTI_HEUR_PART tiene en consideración dos parámetros adicionales:

- *PopSz*: número de reglas de prioridad que se consideran en la heurística MULTI_HEUR_PART.
- *CmbSz*: número de reglas de prioridad que se combinan en la resolución del problema de partición. Si *CmbSz* es inferior a *PopSz*, es necesario resolver varios problemas de partición con todas las posibles combinaciones de grupos de *CmbSz* reglas entre las *PopSz* reglas de prioridad. Por ejemplo, si *CmbSz* = 4 y *PopSz* =

6, se deben resolver tantos problemas de partición como combinaciones de 4 reglas de prioridad se obtengan usando 6 reglas. En cambio, si $CmbSz$ es igual a $PopSz$, se debe resolver un único problema de partición combinando $CmbSz = PopSz$ reglas.

La selección de las $PopSz$ reglas de prioridad entre todas las posibles a combinar en MULTI_HEUR_PART se realiza siguiendo los siguientes criterios:

1. Mejor solución obtenida con HEUR_PART_1.
2. En caso de empate, se utiliza un indicador (tpc_{TB}) relacionado con el grado de variedad entre las U-estaciones encontradas en el Paso 2 de HEUR_PART_1 por distintas reglas de prioridad.

Considerando las reglas de prioridad ya seleccionadas según el primer criterio y las que empatan, para cada combinación de $PopSz$ reglas de prioridad (incluyendo siempre las ya seleccionadas según el primer criterio) se calcula el indicador tpc_{TB} . Siendo:

$NUnS_t$ número de U-estaciones candidatas obtenidas en el Paso 2 de HEUR_PART_1 con la regla de prioridad t ($t = 1, \dots, PopSz$). Las reglas de prioridad t se ordenan de manera creciente según el número de U-estaciones encontradas.

$NUnS_{mix}$ número de U-estaciones diferentes resultantes de la combinación de $PopSz$ reglas de prioridad diferentes.

tpc_{TB} relación entre el número de U-estaciones repetidas en el conjunto de las reglas combinadas, respecto al número total de U-estaciones generadas por el conjunto de estaciones combinadas, exceptuando aquella regla que más combinaciones genera; $0 \leq tpc_{TB} \leq 1$. Es decir:

$$tpc_{TB} = \frac{\sum_{t=1}^{PopSz} NUnS_t - NUnS_{mix}}{\sum_{t=1}^{PopSz-1} NUnS_t} \cdot 100$$

Supóngase $PopSz = 5$, que las reglas R1, R2 y R3 ya fueron seleccionadas según el primer criterio y que ahora empatan las reglas R4, R5, R6 y R7. Así, se debe calcular el valor del indicador tpc_{TB} para cada una de las siguientes 6 combinaciones de 5 reglas: R1+R2+R3+R4+R5, R1+R2+R3+R4+R6, R1+R2+R3+R4+R7, R1+R2+R3+R5+R6, R1+R2+R3+R5+R7 y R1+R2+R3+R6+R7.

A continuación, se muestra un ejemplo de cálculo de tpc_{TB} . Considérese $PopSz = 3$ y las siguientes U-estaciones encontradas por tres reglas de prioridad:

Regla de prioridad 1 → U-estación 1: [tareas 1,2,3]; U-estación 2: [1,3]; U-estación 3: [2,4].

Regla de prioridad 2 → U-estación 1: [1,2,3]; U-estación 2: [1,4]; U-estación 3: [2,3]; U-estación 4: [2,3,4].

Regla de prioridad 3 → U-estación 1: [1,2]; U-estación 2: [1,3]; U-estación 3: [1,4]; U-estación 4: [2,4]; U-estación 5: [3,4].

$NUnst$ toma los siguientes valores: $NUnst_1 = 3$, $NUnst_2 = 4$, $NUnst_3 = 5$. Con la combinación de las tres reglas se generan 8 U-estaciones diferentes ([1,2], [1,2,3], [1,3], [1,4], [2,3], [2,3,4], [2,4], [3,4]); $NUnst_{mix} = 8$. Así pues, para la combinación de las reglas de prioridad 1, 2 y 3: $tpc_{TB} = (12 - 8)/7 = 57\%$.

Para ver la influencia del valor tpc_{TB} se realizó una experiencia computacional previa con los 50 ejemplares de Otto *et al.* (2013) (las instancias utilizadas se detallan en el Apéndice 1) usados hasta el momento en experiencias previas y como conjunto de calibrado. En la experiencia computacional, los posibles valores de tpc_{TB} se agrupan en cuatro intervalos: [0%-15%), [15%-35%), [35%-50%) y [50%-100%], en función de los resultados obtenidos. Para cada uno de estos intervalos, se determina el porcentaje de ejemplares en los que se mejoran los resultados al resolver el problema de partición usando las U-estaciones de las $PopSz$ reglas de prioridad, respecto a resolverlo utilizando las U-estaciones de cada regla por separado. En relación a $CmbSz$, en dichas pruebas computacionales se incluyen tanto casos dónde $CmbSz$ coincide con $PopSz$, como casos con un valor de $CmbSz$ inferior. Los resultados obtenidos se muestran en la Tabla 19.

tpc_{TB}	Porcentaje de casos en los que se mejoran los resultados obtenidos con el problema de partición usando las $PopSz$ reglas de prioridad respecto a resolverlo con cada regla de prioridad por separado
[0%-15%)	56.50%
[15%-35%)	43.46%
[35%-50%)	55.33%
$\geq 50\%$	50.00%

Tabla 19. Porcentaje de casos que mejoran los resultados al resolver un problema de partición combinado según tpc_{TB} . Fuente: propia

Finalmente, y en función de los resultados de la Tabla 19, se establece un orden de prioridad según el indicador tpc_{TB} entre las reglas de prioridad a incluir en $PopSz$ (véase Tabla 20).

Prioridad	tpc_{TB}
1	[0%-15%]
2	[35%-50%]
3	$\geq 50\%$
4	[15%-35%]

Tabla 20. Criterio de desempate según tpc_{TB} para seleccionar las reglas de prioridad a incluir en $PopSz$. Fuente: propia

- En caso de persistir empates, las reglas de prioridad a incluir en $PopSz$ se seleccionan de manera aleatoria.

A continuación, se muestra un ejemplo del proceso de selección de las $PopSz$ reglas de prioridad a utilizar en MULTI_HEUR_PART. Considérese $PopSz = 5$. La Tabla 21 muestra un ejemplo de los resultados obtenidos al resolver un ejemplar con HEUR_PART_1 con las 12 reglas de prioridad (R1-R12).

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
Trabajadores	12	12	12	12	12	12	12	13	13	13	14	14
Estaciones	3	3	5	4	5	6	5	4	5	7	5	5

Tabla 21. Ejemplo de resultados obtenidos con HEUR_PART_1 con las 12 reglas de prioridad. Fuente: propia

Como se puede observar en la Tabla 21, la reglas que obtienen mejores resultados son R1 y R2 (12 trabajadores; 3 estaciones), seguidas de R4 (12;4). En cuarto lugar, con 12 trabajadores y 5 estaciones, se encuentran R3, R5 y R7. Aplicando el primer criterio de selección se incluirán R1, R2 y R4, y sólo dos reglas entre R3, R5 y R7. Para seleccionar estas dos reglas de prioridad, se utiliza el segundo criterio de selección, que se basa en el valor del indicador tpc_{TB} . La Tabla 22 muestra los valores de tpc_{TB} según las distintas combinaciones de $PopSz = 5$ reglas de prioridad.

Combinaciones de $PopSz$ reglas	tpc_{TB}	
R1+R2+R4	+R3+R5	13%
	+R3+R7	56%
	+R5+R7	45%

Tabla 22. Valores de tpc_{TB} según las distintas combinaciones de $PopSz$ reglas de prioridad del ejemplo. Fuente: propia

Considerando los valores de tpc_{TB} indicados en la Tabla 22, y utilizando el orden de prioridad de la Tabla 20, se seleccionan las reglas R3 y R5. De esta forma, $PopSz$ contendrá las reglas R1, R2, R3, R4 y R5.

Los valores $PopSz$ y $CmbSz$ utilizados para los experimentos computacionales se detallan en la sección 5.9.

5.8. Heurística OPWK_PART

En esta sección se presenta OPWK_PART, un procedimiento heurístico que resuelve el SMALBP-DTT1 basándose en mejorar las soluciones obtenidas con HEUR_PART mediante intercambio y transferencia de tareas entre estaciones. La experiencia computacional de OPWK_PART se detalla en la sección 5.9.

OPWK_PART consiste en modificar las estaciones de una solución de HEUR_PART (generando intercambios y transferencias de tareas entre estaciones) y en resolver un problema de partición con restricciones. Igual que en los demás procedimientos presentados para el SMALBP-DTT1, el objetivo principal de OPWK_PART es minimizar el número de trabajadores y el objetivo secundario es minimizar el número de estaciones. Para ejemplificar el funcionamiento de este procedimiento, la solución obtenida en HEUR_PART (Figura 13) se utiliza en las secciones 5.8.1 y 5.8.2.

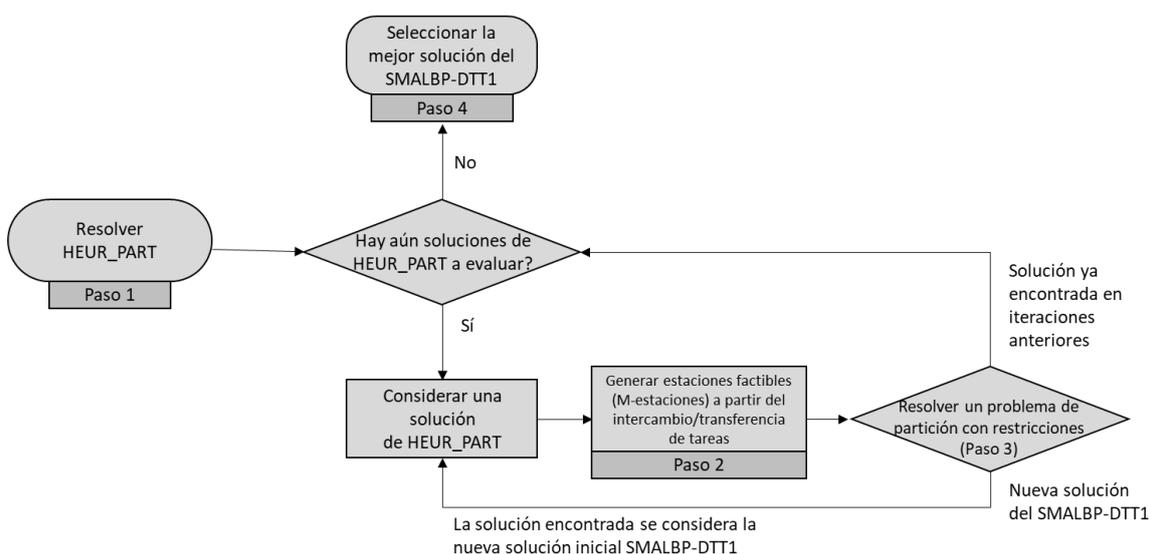


Figura 14. Diagrama de flujo del procedimiento OPWK_PART. Fuente: propia

El diagrama de flujo de OPWK_PART se presenta en la Figura 14. El procedimiento consta de los siguientes pasos que se detallan en las siguientes subsecciones:

Paso 1. Resolver HEUR_PART (sección 5.3).

Para cada una de las soluciones de HEUR_PART, se aplican los pasos 2 y 3.

Paso 2. Evaluar todos los posibles intercambios y transferencias de tareas entre las estaciones de la solución HEUR_PART, y seleccionar únicamente las estaciones modificadas que sean factibles (sección 5.8.1). En concreto, se evalúan cuatro tipos de modificaciones: intercambio de una sola tarea; transferencia de una

sola tarea; intercambio de una sola tarea seguido de transferencia de una sola tarea; y transferencia de una sola tarea seguida de intercambio de una sola tarea. Sean las estaciones j_1 y j_2 (sin tareas en común), un intercambio/transferencia de tareas entre esas dos estaciones (generando las estaciones j_1' y j_2') es factible si se respetan las relaciones de precedencia en j_1' y j_2' (sección 5.8.1.1), y el número de trabajadores no aumenta (es decir, la suma de los trabajadores de j_1' y j_2' no es superior a la suma de los trabajadores de j_1 y j_2 ; sección 5.8.1.2). Para evitar ambigüedades, una estación modificada se denominará a partir de ahora M-estación.

Paso 3. Resolver un problema de partición con restricciones mediante programación matemática con todas las M-estaciones factibles encontradas en el paso anterior (sección 5.8.2). El modelo matemático minimiza el número de trabajadores como objetivo principal y el número de M-estaciones como objetivo secundario. Si la solución encontrada al resolver el problema de partición con restricciones es diferente de todas las soluciones anteriores obtenidas, dicha solución se considera la nueva solución en curso de SMALBP-DTT1 y el proceso vuelve al Paso 2. En caso contrario, se considera la siguiente solución de HEUR_PART y el proceso reinicia al Paso 2.

Paso 4. Se selecciona la solución con el menor número de trabajadores de entre todas las soluciones encontradas en el Paso 3. En caso de empate, se selecciona la solución con el menor número de M-estaciones.

5.8.1. Modificación de estaciones mediante intercambio y transferencia de tareas (Generación de M-estaciones)

El Paso 2 consiste en modificar las estaciones de la solución de HEUR_PART (solución SMALBP-DTT1 en curso). Para ello se aplican cuatro tipos de modificaciones:

- **Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso:** sean j_1 y j_2 estaciones de la solución SMALBP-DTT1 en curso que no tienen tareas en común, con la tarea i_1 contenida en j_1 y la tarea i_2 contenida en j_2 ; un intercambio de una sola tarea consiste en permutar las tareas i_1 e i_2 entre j_1 y j_2 para generar j_1' (que contiene i_2 pero no i_1) y j_2' (que contiene i_1 pero no i_2). Se prueban todos los posibles intercambios de una sola tarea entre todas las estaciones de la solución SMALBP-DTT1 en curso.
- **Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso:** sean j_1 y j_2 estaciones de la solución SMALBP-DTT1 en curso que no tienen tareas en común y con la tarea i_1 contenida en j_1 ; una transferencia

de una sola tarea consiste en mover la tarea i_1 de j_1 a j_2 para generar j_1' (que no contiene i_1) y j_2' (que contiene i_1). Se estudian todas las posibles transferencias de una sola tarea entre todas las estaciones de la solución SMALBP-DTT1 en curso, incluso aquellas en las que j_1 contiene sólo la tarea i_1 .

- **Transferencia de una sola tarea entre las estaciones encontradas en “Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso”**. El proceso es idéntico al de transferir una sola tarea descrito en el párrafo anterior, con la única diferencia de que las estaciones iniciales son las encontradas con la modificación “Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso”, incluyendo también las estaciones de la solución en curso SMALBP-DTT1 que no generaron ningún intercambio factible.
- **Intercambio de una sola tarea entre las estaciones encontradas en “Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso”**. El proceso es idéntico al de intercambiar una sola tarea descrito anteriormente, con la única diferencia de que las estaciones iniciales son las encontradas con la modificación “Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT1 en curso”, incluyendo también las estaciones de la solución en curso SMALBP-DTT1 que no generaron ninguna transferencia factible.

Si un intercambio/transferencia de tareas entre dos estaciones se considera factible en términos de precedencia (sección 5.8.1.1) y no aumenta el número de trabajadores (sección 5.8.1.2), dichas dos estaciones modificadas (llamadas M-estaciones) se incluyen en la lista de M-estaciones candidatas para el problema de partición con restricciones (Paso 3). En caso de que, después de intentar todos los intercambios y transferencias de tareas posibles, sigue habiendo una estación que no se ha podido modificar ni una sola vez, dicha estación también se incluye en la lista de M-estaciones candidatas.

5.8.1.1. Factibilidad de pares de estaciones modificadas según precedencias

Considérense las estaciones j_1 y j_2 , un intercambio/transferencia de tareas entre esas dos estaciones genera j_1' y j_2' . Las estaciones j_1' y j_2' se consideran factibles entre sí en términos de relaciones de precedencia si se cumplen todas las condiciones siguientes:

- Si no hay tareas (de las no incluidas en la estación j_1') que sean al mismo tiempo sucesoras y predecesoras con respecto a las tareas de la estación j_1' .
- Si no hay tareas (de las no incluidas en la estación j_2') que sean al mismo tiempo sucesoras y predecesoras con respecto a las tareas de la estación j_2' .
- Si en la estación j_1' no hay al mismo tiempo tareas sucesoras y predecesoras con respecto a la estación j_2' .

- Si en la estación j_2' no hay al mismo tiempo tareas sucesoras y predecesoras con respecto a la estación j_1' .

Para la solución inicial de HEUR_PART (Figura 13), considerando la estación 2 (tareas 2, 3, 4 y 5) y la estación 3 (tareas 6 y 7), se genera una transferencia no factible moviendo la tarea 2 a la estación 3. Esta transferencia crearía la estación 2' (tareas 3, 4 y 5) y la estación 3' (tareas 2, 6 y 7). La estación 3' es infactible porque la tarea 5 es a la vez sucesora de la tarea 2 y predecesora de la tarea 6. Un ejemplo de intercambio infactible surge al cambiar la tarea 2 y la tarea 7. Este intercambio crearía la estación 2'' (tareas 3, 4, 5 y 7) y la estación 3'' (tareas 2 y 6). La estación 3'' es infactible por la misma razón explicada anteriormente para la estación 3'.

5.8.1.2. Factibilidad de pares de estaciones modificadas sin aumentar el número de trabajadores

Considérense las estaciones j_1' y j_2' , generadas por un intercambio/transferencia de tareas entre j_1 y j_2 . En esta sección se calcula el número de trabajadores de las estaciones j_1' y j_2' . Si la suma de los trabajadores de j_1' y j_2' no es mayor que la suma de los trabajadores de j_1 y j_2 , entonces j_1' y j_2' son factibles.

Para calcular el número de trabajadores de una estación se resuelve iterativamente el modelo matemático presentado en la sección 5.3.2.2. La primera iteración se resuelve considerando la cota inferior del número de trabajadores NW , teniendo en cuenta que la suma de los tiempos de proceso de todas las tareas no sea superior al producto del tiempo de ciclo CT por el número de trabajadores. Si el modelo matemático no encuentra una solución factible, el número de trabajadores NW se incrementa en una unidad y el modelo se resuelve de nuevo. El número de trabajadores NW no puede ser mayor a la cota superior UB^0 . Se establece un límite de tiempo de cálculo de $\beta 2$ segundos (el valor $\beta 2$ utilizado para la experiencia computacional se detalla en la sección 5.9).

Partiendo de la solución obtenida con HEUR_PART (Figura 13), la lista de M-estaciones candidatas que son factibles en términos de precedencias y no aumentan el número de trabajadores se muestra en la Tabla 23.

Tareas incluidas en la M-estación	Número de trabajadores
1	1
2 + 4 + 5	2
2 + 3 + 4 + 5	2
3 + 6 + 7	2
6+7	2

Tabla 23. M-estaciones candidatas a formar parte de la solución final OPWK_PART. Fuente: propia

5.8.2. Resolución de un problema de partición con restricciones

En el Paso 3, se utiliza un modelo matemático para resolver un problema de partición con restricciones considerando las M-estaciones. La lista de M-estaciones candidatas está compuesta por las M-estaciones encontradas en el Paso 2 en curso y las M-estaciones candidatas que se generaron en el Paso 2 para las $p1$ iteraciones anteriores. La propuesta de $p1$ evita tener una cantidad demasiado grande de M-estaciones candidatas, lo que conllevaría un aumento significativo de los tiempos computacionales para resolver el problema de partición; el valor de $p1$ utilizado para la experiencia computacional se detalla en la sección 5.9. Además, cada tarea individual también se incluye como una M-estación candidata. El objetivo es minimizar el número de trabajadores como objetivo principal y el número de M-estaciones como objetivo secundario. En caso de empate, se selecciona la solución que utiliza el menor número de M-estaciones incluidas en las $n1$ soluciones anteriores (el valor $n1$ utilizado para la experiencia computacional se detalla en la sección 5.9). Dicha condición de desempate se añade para promover la generación de soluciones con M-estaciones diversas y, por tanto, para tratar de evitar la condición final del Paso 3 (es decir, encontrar una solución idéntica a otra anterior).

Datos

N número de tareas ($i = 1, \dots, N$).

J número de M-estaciones candidatas ($j = 1, \dots, J$).

NW_j número de trabajadores en la M-estación j ($j = 1, \dots, J$).

$OldS_j$ 1 si la M-estación j ha sido seleccionada en las $n1$ soluciones previas del problema de partición, 0 en caso contrario ($j = 1, \dots, J$).

IS_{ji} 1 si en la M-estación j está incluida la tarea i , 0 en caso contrario:

$$i = 1, \dots, N; j = 1, \dots, J.$$

\overline{INC} grupos de M-estaciones que no tienen tareas en común pero que son infactibles por razones de precedencia.

R número de M-estaciones en cada grupo de \overline{INC} ($R = 2, \dots, N$).

$$\overline{INC} = \{(r_1, \dots, r_R) \mid r_1 \in \{1, \dots, J - (R - 1)\}, \dots, r_R \in \{r_1 + (R - 1), \dots, J\}, r_1, \dots, r_R \text{ infactibles}\}.$$

Dos M-estaciones ($R = 2$) son infactibles si en una de las M-estaciones hay al

mismo tiempo tareas sucesoras y predecesoras con respecto a las tareas de la otra M-estación. Tres o más M-estaciones ($R = 3, \dots, N$) son infactibles si cada M-estación tiene relaciones de precedencia con sólo otras dos M-estaciones (más concretamente, tiene tareas predecesoras con respecto a las tareas de una M-estación y tareas sucesoras con respecto a las tareas de la otra M-estación) y no hay infactibilidades en ninguna combinación de $2, \dots, R - 1$ M-estaciones.

Variables

$x_j \in \{0,1\}$ 1 si se usa la M-estación j , 0 en caso contrario: $j = 1, \dots, J$.

Función objetivo

La función objetivo minimiza número total de trabajadores como primer objetivo, el número total de M-estaciones seleccionadas como objetivo secundario y el número de M-estaciones ya utilizadas en las $n1$ soluciones anteriores del problema de partición como objetivo terciario:

$$[MIN] \sum_{j=1}^J NW_j \cdot x_j + \frac{1}{N+1} \cdot \left(\sum_{j=1}^J x_j + \frac{1}{N+1} \cdot \sum_{j=1}^J OldS_j \cdot x_j \right) \quad (\text{Ec. 49})$$

Restricciones

$$\sum_{j=1}^J IS_{ji} \cdot x_j = 1 \quad i = 1, \dots, N \quad (\text{Ec. 50})$$

$$\sum_{j=r_1}^{r_R} x_j \leq R - 1 \quad (r_1, \dots, r_R) \in \overline{INC} \quad (\text{Ec. 51})$$

Las restricciones (50) garantizan que cada tarea se incluya en la solución final una sola vez. Las restricciones (51) evitan que se asignen M-estaciones infactibles por razones de precedencia.

Para el ejemplo ilustrativo (Tabla 23), la solución obtenida en la primera iteración al resolver el problema de partición utiliza 5 trabajadores y 3 M-estaciones: la M-estación 1 que contiene la tarea 1, la M-estación 2 que contiene las tareas 2, 4 y 5 y la M-estación 3 que contiene las tareas 3, 6 y 7. La solución se representa en la Figura 9. Esta solución obtiene el mismo número de trabajadores y estaciones que la solución inicial de HEUR_PART, pero utiliza distintas estaciones ya que se aplica la regla de desempate que minimiza el número de M-estaciones utilizadas en las soluciones anteriores.

5.9. Experiencia computacional para comparar HEUR_PART con R16, MULTI_HEUR_PART y OPWK_PART

De las experiencias computacionales realizadas (secciones 5.4, 5.5.2 y 5.6.3), el procedimiento HEUR_PART (sección 5.3) mejorado con la incorporación de la regla de prioridad R16 diseñada mediante EAGH-CKTL (sección 5.6) es el que presenta, por el momento, los mejores resultados entre todos los procedimientos propuestos en esta tesis para resolver el SMALBP-DTT1.

En esta sección dicho procedimiento se compara con MULTI_HEUR_PART (sección 5.7) y OPWK_PART (sección 5.8). El objetivo de esta experiencia computacional es identificar el procedimiento que obtiene los mejores resultados y compararlo con procedimientos propuestos en la literatura.

En primer lugar, los procedimientos HEUR_PART con R16, MULTI_HEUR_PART y OPWK_PART se comparan en la sección 5.9.2.1, utilizando los ejemplares indicados en la sección 5.9.1.

Cabe especificar que en la literatura no existe ningún otro procedimiento que resuelva el SMALBP-DTT1 con la minimización del número de trabajadores como primer objetivo y el número de estaciones como segundo. Sin embargo, Sepahi y Jalali Naini (2014) proponen una heurística (denominada Max_S) que también resuelve el SMALBP-DTT1, pero con el objetivo de minimizar únicamente el número de estaciones. En la sección 5.9.2.2 se compara la heurística Max_S con OPWK_PART, el procedimiento que presenta los mejores resultados de la comparación realizada en la sección 5.9.2.1, teniendo en cuenta que los objetivos son diferentes.

A continuación, en el apartado 5.9.2.3, se compara el número de trabajadores y estaciones obtenidos utilizando OPWK_PART con el número óptimo del SALBP-1.

Por último, OPWK_PART se compara, en la sección 5.9.2.4, con GanttSA, un algoritmo de búsqueda basado en un SA propuesto en Kellegöz (2017) que resuelve el SMALBP con los mismos objetivos jerárquicos, pero considerando que los tiempos de las tareas son independientes del número de trabajadores en la estación (aunque OPWK_PART no ha sido diseñada bajo estas circunstancias).

5.9.1. Características de la experiencia computacional

En la experiencia computacional de las secciones 5.9.2.1-5.9.2.3 se utilizan los mismos 250 ejemplares de Otto *et al.* (2013), con las mismas características que los utilizados en las experiencias computacionales anteriores (para más detalle, ver la sección 5.4.1).

Para la comparación con GanttSA en la sección 5.9.2.4 se utilizan los ejemplares que usan los autores con dicho procedimiento: algunas de las instancias de la literatura de referencia, que pueden descargarse en www.assembly-line-balancing.de. Para comparar OPWK_PART con GanttSA, los tiempos de las tareas se consideran independientes (es decir, invariables) del número de trabajadores. Además, se consideran las mismas cotas superiores en el número de trabajadores por estación utilizados en Kellegöz (2017). Para ejemplares pequeños, el número máximo de trabajadores en la misma estación se fija en 2 ($UB^O = 2$) y 4 ($UB^O = 4$). Para ejemplares medianos y grandes, los experimentos computacionales se realizan considerando un máximo de 2 ($UB^O = 2$), 4 ($UB^O = 4$) y 6 ($UB^O = 6$) trabajadores en la misma estación.

Por un lado, en referencia a la heurística MULTI_HEUR_PART, se ha llevado a cabo una prueba computacional preliminar para determinar los valores de $PopSz$ y $CmbSz$. Para dicha prueba computacional preliminar se utilizaron los 50 ejemplares de Otto *et al.* (2013) ya utilizados en experiencias computacionales previas de calibración en esta tesis (las instancias se detallan en el Apéndice 1). La Tabla 24 muestra los resultados de la prueba computacional realizada. Para comparar los resultados obtenidos por cada combinación de los parámetros $PopSz$ y $CmbSz$, se utiliza la función objetivo del modelo matemático Ec. 1. Para cada combinación de parámetros testeada se muestra el sumatorio de los valores de la función objetivo y el tiempo medio de cálculo. La variante que obtiene mejores resultados considera $PopSz = 6$ reglas de prioridad y resuelve problemas de partición combinando $CmbSz = 4$ reglas. Aunque en principio con valores superiores de $PopSz$ se deberían obtener mejores resultados, esto no es así debido a que para ciertos ejemplares el número de U-estaciones candidatas del problema de partición es tan elevado que no es posible encontrar una solución factible dentro del tiempo máximo de cálculo de 3600 segundos.

	$CmbSz = 4$ $PopSz = 4$	$CmbSz = 4$ $PopSz = 6$	$CmbSz = 6$ $PopSz = 6$	$CmbSz = 8$ $PopSz = 8$	$PopSz = 12$
Sumatorio función objetivo	1246.82	1246.59	1246.96	1246.78	1247.01
Tiempo medio de cálculo (s)	2770	2985	2895	3010	3100

Tabla 24. Resultados de la experiencia computacional previa de MULTI_HEUR_PART. Fuente: propia

Por otro lado, en relación con la heurística OPWK_PART, se ha realizado una prueba computacional preliminar con los mismos 50 ejemplares de Otto *et al.* (2013) para determinar los siguientes parámetros: parámetro $p1$ (en el problema de partición se consideran las M-estaciones candidatas de la iteración en curso y las de las $p1$ iteraciones anteriores), el parámetro $n1$ (como criterio de desempate, se selecciona la solución que utiliza el menor número de M-estaciones incluidas en las $n1$ soluciones anteriores) y el parámetro $\beta2$ (tiempo de cálculo en segundos del número de trabajadores en una estación).

La Tabla 25 muestra los resultados de la prueba computacional. Para comparar los resultados obtenidos por cada combinación de los tres parámetros se utiliza la función objetivo definida en Ec. 1. Para cada combinación de parámetros, la Tabla 25 contiene el sumatorio de los valores de la función objetivo para los 50 ejemplares y , entre paréntesis, el tiempo medio de cálculo. Los mejores resultados se obtienen con: $p1 = 3$, $n1 = 2$ y $\beta2 = 5$ segundos.

	Incorporar las estaciones candidatas de las $p1$ iteraciones anteriores	Se selecciona la solución que utiliza el menor número de estaciones incluidas en las $n1$ soluciones anteriores		
		$n1 = 1$	$n1 = 2$	$n1 = 3$
$\beta2 = 5$ segundos	$p1 = 1$	1248.28 (2880 segundos)	1248.49 (2905 segundos)	1248.66 (2995 segundos)
	$p1 = 2$	1244.50 (3090 segundos)	1243.93 (3050 segundos)	1244.26 (3115 segundos)
	$p1 = 3$	1242.43 (3125 segundos)	1241.35 (3160 segundos)	1241.62 (3300 segundos)
$\beta2 = 10$ segundos	$p1 = 1$	1248.19 (3130 segundos)	1248.70 (3090 segundos)	1248.70 (3190 segundos)
	$p1 = 2$	1245.64 (3210 segundos)	1245.88 (3235 segundos)	1245.75 (3375 segundos)
	$p1 = 3$	1244.32 (3300 segundos)	1244.56 (3290 segundos)	1244.02 (3370 segundos)

Tabla 25. Resultados de la experiencia computacional previa de OPWK_PART. Fuente: propia

5.9.2. Análisis de los resultados

5.9.2.1. Comparación entre HEUR_PART con R16, MULTI_HEUR_PART y OPWK_PART

En esta sección el procedimiento HEUR_PART con R16 se compara con MULTI_HEUR_PART y con OPWK_PART.

Los resultados se muestran en la Tabla 26. La estructura de la Tabla 26 es la misma que la de la Tabla 8. El detalle de los resultados obtenidos por cada procedimiento en las 250 instancias de Otto *et al.* (2013) se muestran en el Apéndice 4.

Los resultados de la Tabla 26 muestran que OPWK_PART y MULTI_HEUR_PART consiguen mejorar los resultados obtenidos por HEUR_PART con R16. La heurística OPWK_PART es la que, en global, presenta mejores resultados (alcanza la mejor solución en el 89.2% de los casos, con una media de 24.83 trabajadores y 9.97 estaciones). En términos de tiempo de ejecución, OPWK_PART presenta mayores tiempos medios de cálculo, pero las diferencias respecto a los otros procedimientos (354 segundos más de media respecto a MULTI_HEUR_PART y 428 segundos más respecto a HEUR_PART con R16) pueden considerarse negligibles en la resolución real de un ALBP.

		HEUR_PAR con R16	MULTI_HEUR_PART	OPWK_PART
50 tareas	Mejor solución alcanzada	46.4%	56.0%	90.4%
	Mejor solución no alcanzada	53.6%	44.0%	9.6%
	Número medio de trabajadores	20.33	20.26	20.07
	Número medio de estaciones	8.54	8.46	8.23
	Gap medio con la cota inferior	4.82%	4.56%	3.43%
	Gap medio con la mejor solución	1.49%	1.24%	0.12%
	Tiempo medio de cálculo (s)	2350	2564	3170
100 tareas	Mejor solución alcanzada	84.0%	92.8%	88.0%
	Mejor solución no alcanzada	16.0%	7.2%	12.0%
	Número medio de trabajadores	29.63	29.58	29.60
	Número medio de estaciones	11.60	11.31	11.70
	Gap medio con la cota inferior	2.38%	2.28%	2.31%
	Gap medio con la mejor solución	0.18%	0.09%	0.11%
	Tiempo medio de cálculo (s)	3344	3297	3400
Global	Mejor solución alcanzada	65.2%	74.4%	89.2%
	Mejor solución no alcanzada	34.8%	25.6%	10.8%
	Número medio de trabajadores	24.98	24.92	24.83
	Número medio de estaciones	10.07	9.88	9.97
	Gap medio con la cota inferior	3.60%	3.42%	2.87%
	Gap medio con la mejor solución	0.84%	0.66%	0.11%
	Tiempo medio de cálculo (s)	2847	2931	3285

Tabla 26. Tabla comparativa entre HEUR_PART incorporando R16, MULTI_HEUR_PART y OPWK_PART.

Fuente: propia

5.9.2.2. Comparación entre OPWK_PART y Max_S

Como se ha mencionado anteriormente, en la literatura no se ha publicado ningún otro procedimiento que resuelva el SMALBP-DTT1 y que tenga como objetivo primario minimizar el número de trabajadores y como secundario minimizar el número de estaciones. Sin embargo, Sepahi y Jalali Naini (2014) proponen Max_S, una heurística que resuelve el SMALBP-DTT1 con el objetivo de minimizar únicamente el número de estaciones. Teniendo en cuenta que sus objetivos son diferentes, en esta sección Max_S se compara con OPWK_PART, el procedimiento que presenta los mejores resultados. La Tabla 27 y la Tabla 28 presentan los resultados de esta comparación.

La Tabla 27 y la Tabla 28 muestran que, en casi la totalidad de las instancias, OPWK_PART obtiene un número de trabajadores inferior respecto a Max_S: 99.2% para instancias de 50 tareas y 92.0% para instancias de 100 tareas.

En particular, en aproximadamente la mitad de las instancias (49.6% para instancias de 50 tareas y 52.8% para instancias de 100 tareas), OPWK_PART obtiene un menor número

de trabajadores (una media de 4.21 trabajadores menos para instancias de 50 tareas y 4.50 trabajadores menos para instancias de 100 tareas) y Max_S obtiene un menor número de estaciones (una media de 2.50 estaciones menos para instancias de 50 tareas y 4.76 estaciones menos para instancias de 100 tareas).

OPWK_PART obtiene un menor número de trabajadores con las mismas estaciones en el 29.6% de instancias de 50 tareas (con una media de 3.54 trabajadores menos) y en el 25.6% de instancias de 100 tareas (con una media de 3.66 trabajadores menos). Incluso, OPWK_PART presenta un menor número tanto de estaciones como de trabajadores en el 20.0% de instancias de 50 tareas (con una media de 3.36 trabajadores menos y 1.12 estaciones menos) y en el 13.6% de instancias de 100 tareas (con una media de 4.29 trabajadores menos y 1.41 estaciones menos).

Max_S consigue un número inferior de trabajadores sólo en el 1.6% de instancias de 100 tareas (0.8% con las mismas estaciones pero menos trabajadores y 0.8% con menos trabajadores y estaciones). En un 2.4% de instancias de 100 tareas Max_S y OPWK_PART obtienen el mismo número de trabajadores, pero Max_S consigue una solución con un número menor de estaciones.

En el 0.8% de instancias de 50 tareas y el 4.0% de instancias de 100 tareas ambos procedimientos obtienen las mismas soluciones.

Resultados de OPWK_PART comparados con Max_S (50 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	0%	0%	20.0% (25/125) 3.36 trabajadores menos 1.12 estaciones menos
Igual número de estaciones	0%	0.8% (1/125)	29.6% (37/125) 3.54 trabajadores menos
Número de estaciones superior	0%	0%	49.6% (62/125) 4.21 trabajadores menos 2.50 estaciones más

Tabla 27. Tabla comparativa para instancias de 50 tareas entre OPWK_PART y Max_S. Fuente: propia

Resultados de OPWK_PART comparados con Max_S (100 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	0%	0%	13.6% (17/125) 4.29 trabajadores menos 1.41 estaciones menos
Igual número de estaciones	0.8% (1/125) 1.00 trabajadores más	4.0% (5/125)	25.6% (32/125) 3.66 trabajadores menos

Resultados de OPWK_PART comparados con Max_S (100 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones superior	0.8% (1/125) 1 trabajadores más 1 estaciones más	2.4% (3/125) 1.33 estaciones más	52.8% (66/125) 4.50 trabajadores menos 4.76 estaciones más

Tabla 28. Tabla comparativa para instancias de 100 tareas entre OPWK_PART y Max_S. Fuente: propia

5.9.2.3. Comparación entre OPWK_PART y el número óptimo de trabajadores y estaciones del SALBP-1

A continuación, se muestra una comparación del número de trabajadores y estaciones obtenidos por OPWK_PART, respecto al número óptimo de trabajadores y estaciones del SALBP-1 especificado en Otto *et al.* (2013). Considerando que en cada estación del SALBP-1 hay un único trabajador, el número óptimo de trabajadores y estaciones coinciden.

En primer lugar, en la Tabla 29 se muestra una comparación del número de trabajadores obtenido por OPWK_PART y el número óptimo para el SALBP-1. Para cada valor del número de trabajadores adicionales en OPWK_PART respecto al óptimo SALBP-1, en la Tabla 29 se indica el porcentaje de ejemplares y la media de estaciones reducidas en OPWK_PART con respecto al óptimo del SALBP-1. Los resultados muestran que, en la mayoría de los casos, OPWK_PART obtiene soluciones con el mismo número de trabajadores y una media de 15.81 estaciones menos (66.4%) o con un solo trabajador adicional y una media de 11.88 estaciones menos (32.0%). Como se indica en la sección 4.1, una de las ventajas de las líneas MAL es la posibilidad de reducir el número de estaciones necesarias respecto a una línea del SALBP, disminuyendo así la longitud de la línea de ensamblaje y mejorando la utilización del espacio.

	Tareas	Número de trabajadores adicionales en OPWK_PART respecto al óptimo del SALBP-1		
		+0	+1	+2
Porcentaje de ejemplares	50	76.0%	23.2%	0.8%
	100	56.8%	40.8%	2.4%
	Global	66.4%	32.0%	1.6%
Media de estaciones reducidas en OPWK_PART con respecto al óptimo del SALBP-1	50	12.65	8.55	17.00
	100	20.03	13.76	34.33
	Global	15.81	11.88	30.00

Tabla 29. Tabla comparativa del número de trabajadores y estaciones obtenidos por OPWK_PART respecto al óptimo del SALBP-1. Fuente: propia

En segundo lugar, la Tabla 30 presenta una comparación entre el número de estaciones reducidas en OPWK_PART con respecto al número óptimo de estaciones del SALBP-1. Los resultados muestran que OPWK_PART consigue reducir el número de estaciones en

la totalidad de las instancias. En particular, la mayoría de las soluciones reducen de 15 a 19 estaciones (37.6%), de 5 a 9 estaciones (26.0%) o de 10 a 14 estaciones (16.0%) respecto la solución del SALBP-1.

Tareas	Número de estaciones reducidas en OPWK_PART con respecto al óptimo del SALBP-1								
	1-4	5-9	10-14	15-19	20-24	25-29	30-34	35-39	40-44
50	9.6%	36.0%	8.8%	40.8%	4.0%	0.8%	0%	0%	0%
100	2.4%	16.0%	23.2%	34.4%	0%	4.0%	10.4%	8.0%	1.6%
Global	6.0%	26.0%	16.0%	37.6%	2.0%	2.4%	5.2%	4.0%	0.8%

Tabla 30. Tabla comparativa del número de estaciones obtenidos por OPWK_PART respecto al óptimo del SALBP-1. Fuente: propia

5.9.2.4. Comparación entre OPWK_PART y GanttSA

En la Tabla 31 se presenta una comparación entre OPWK_PART y GanttSA. GanttSA es un algoritmo de búsqueda por SA propuesto en Kellegöz (2017) que resuelve el SMALBP-1 con los mismos objetivos jerárquicos aquí adoptados (minimizar el número de trabajadores como objetivo primario y el número de estaciones como secundario), pero considerando que los tiempos de las tareas son independientes (y, así, invariables) del número de trabajadores en la estación. A efectos comparativos, los tiempos de las tareas también se consideran independientes del número de trabajadores al resolver OPWK_PART (aunque dicho procedimiento no se ha diseñado específicamente considerando tiempos de tarea deterministas).

En la Tabla 31, para cada ejemplar, si OPWK_PART obtiene un mejor número de trabajadores (objetivo principal), las celdas del número de trabajadores y estaciones se colorean de verde (de rojo si GanttSA obtiene un mejor número de trabajadores). Si OPWK_PART mejora sólo el número de estaciones (objetivo secundario), se colorea de verde sólo la celda de las estaciones (de rojo si GanttSA mejora sólo las estaciones). Si las soluciones son iguales, las celdas permanecen blancas.

Como se observa en la Tabla 31, los resultados de OPWK_PART son mejores que los obtenidos por GanttSA. OPWK_PART supera a GanttSA en 19 instancias (27.9%), obtiene los mismos resultados en 48 ejemplares (70.6%) y sólo en 1 instancia obtiene resultados peores (1.5%). OPWK_PART alcanza una mejor solución en el 98.5% de los casos, frente a GanttSA que lo hace en el 72.1%.

Problema	CT	UB ^o	GanttSA (Kellegöz,2017)		OPWK_PART		
			Trabajadores	Estaciones	Trabajadores	Estaciones	Tiempo de cálculo (s)
Heskiaoff	138	2	8	5	8	5	15
		4	8	4	8	4	38
	256	2	4	3	4	3	11
		4	4	3	4	3	17

Problema	CT	UB ^o	GanttSA (Kellegöz,2017)		OPWK_PART		
			Trabajadores	Estaciones	Trabajadores	Estaciones	Tiempo de cálculo (s)
	342	2	3	2	3	2	15
		4	3	2	3	2	21
Sawyer	25	2	14	8	14	8	37
		4	14	8	14	8	78
	27	2	13	8	13	8	19
		4	13	8	13	8	68
	54	2	7	4	7	4	12
		4	7	4	7	4	23
	75	2	5	3	5	3	16
		4	5	3	5	3	24
Kilbridge	57	2	10	6	10	6	18
		4	10	5	10	5	36
		6	10	5	10	5	51
	110	2	6	3	6	3	67
		4	6	3	6	3	79
		6	6	3	6	3	81
	184	2	3	2	3	2	309
		4	3	2	3	2	320
		6	3	2	3	2	343
Tonge	176	2	21	12	21	12	65
		4	21	10	21	10	276
		6	21	10	21	10	686
	410	2	9	5	9	5	157
		4	9	4	9	3	411
		6	9	4	9	3	466
	527	2	7	4	7	4	295
		4	7	3	7	3	482
		6	7	3	7	3	519
Arcus1	5048	2	16	10	16	10	54
		4	16	10	16	10	234
		6	16	10	16	10	249
	7571	2	11	6	11	6	61
		4	11	6	11	6	129
		6	11	6	11	6	140
	10816	2	8	5	8	5	274
		4	8	5	8	4	387
		6	8	5	8	4	465
Arcus2	5755	2	27	16	27	15	134
		4	27	13	27	12	569
		6	27	13	27	12	1582
	10743	2	15	9	15	8	34
		4	15	8	15	7	133
		6	15	8	15	7	236
	17067	2	9	5	9	5	156
		4	9	5	9	4	250
		6	9	5	9	4	318
Barthol2	84	2	51	26	51	27	520
		4	51	16	51	16	2675
		6	51	14	51	14	3366
	106	2	41	21	41	21	452
		4	41	13	41	11	1993
		6	41	12	41	10	3600
	170	2	25	13	25	13	421
		4	25	8	25	8	1549
		6	25	8	25	6	3369
Barthold	403	2	14	7	14	7	331
		4	14	5	14	4	732
		6	14	5	14	4	1881
	513	2	11	6	11	6	147
		4	11	4	11	3	1975
		6	11	4	11	3	3360

Problema	CT	UB ^o	GanttSA (Kellegöz, 2017)		OPWK_PART		
			Trabajadores	Estaciones	Trabajadores	Estaciones	Tiempo de cálculo (s)
	805	2	7	4	7	4	573
		4	7	3	7	3	1130
		6	7	3	7	3	3280
Media		2	13.80	7.72	13.80	7.68	167.72
		4	13.80	6.20	13.80	5.84	545.16
		6	16.17	6.67	16.17	6.06	1332.89

Tabla 31. Tabla comparativa entre OPWK_PART y GanttSA. Fuente: propia

5.10. Conclusiones sobre los procedimientos de resolución del SMALBP-DTT1

En la sección 5 de la presente tesis doctoral se aborda la resolución de la variante SMALBP-DTT1. En primer lugar, se propone un modelo matemático (MILP-DTT1; sección 5.1) que corrige, evitando soluciones infactibles, y mejora los que se han publicado en la literatura hasta el momento actual. En segundo lugar, se desarrollan dos procedimientos basados en Relajar y Fijar (1RF y 2RF; sección 5.2). En tercer lugar, se presenta un procedimiento heurístico que incluye la resolución de un problema de partición con restricciones (HEUR_PART; sección 5.3).

Una primera experiencia computacional (sección 5.4) con el objetivo de comparar la eficacia del método exacto desarrollado (MILP-DTT1) respecto a los heurísticos (1RF, 2RF y HEUR_PART), muestra que HEUR_PART presenta un mejor comportamiento para resolver SMALBP-DTT1 que MILP-DTT1, 1RF y 2RF.

En la sección 5.5 se presentan un conjunto de variantes de HEUR_PART con el objetivo de mejorar los resultados obtenidos: HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE. Analizando los resultados de la experiencia computacional (sección 5.5.2), se observa que la variante HEUR_PART_SGL obtiene el mismo resultado que HEUR_PART con un tiempo medio de cálculo inferior.

A continuación, en la sección 5.6, se mejora el procedimiento HEUR_PART utilizando el procedimiento Empirically Adjusted Greedy Heuristics (EAGH), introducido en Corominas (2005).

Una de las contribuciones de esta tesis doctoral es la presentación de un nuevo procedimiento (denominado "EAGH-CKTL") para el diseño de heurísticas *greedy*, basado en el uso de EAGH combinado con el concepto de cóctel de heurísticas introducido en Corominas *et al.* (2012). El objetivo de EAGH-CKTL es diseñar heurísticas *greedy* mediante EAGH que presenten un buen rendimiento como parte de un cóctel de

heurísticas.

EAGH y EAGH-CKTL se utilizan para diseñar nuevas reglas de prioridad para resolver el SMALBP-DTT1 mediante HEUR_PART_1; recuérdese que el término "HEUR_PART_1" se refiere a la heurística que dada una única regla de prioridad encuentra una solución SMALBP-DTT1 siguiendo los pasos de HEUR_PART. En primer lugar, se generan tres nuevas reglas de prioridad mediante EAGH, R13, R14 y R15, que presentan un buen rendimiento por sí solas. Este enfoque de EAGH es el que se ha utilizado en la literatura hasta el momento. En segundo lugar, se diseñan tres reglas de prioridad adicionales mediante EAGH-CKTL: R16, R17 y R18. En este segundo caso, las reglas de prioridad se construyen para mostrar un buen rendimiento teniendo también en cuenta los resultados obtenidos por el cóctel que forman las 12 reglas de prioridad utilizadas en HEUR_PART.

Los experimentos computacionales realizados muestran la eficacia tanto de EAGH como de EAGH-CKTL, en el diseño de nuevas reglas de prioridad para resolver el SMALBP-DTT1 mediante HEUR_PART_1. Por un lado, la nueva regla de prioridad R14 (diseñada con EAGH) presenta un mejor rendimiento que cualquiera de las 12 reglas de prioridad utilizadas en HEUR_PART y las otras reglas de prioridad generadas con EAGH. Por otro lado, la nueva regla de prioridad R16 (diseñada con EAGH-CKTL) resulta ser la que mejores resultados alcanza. La adición de la regla de prioridad R16 al procedimiento HEUR_PART produce una notable mejora en los resultados obtenidos.

Seguidamente se presentan otros dos procedimientos que incluyen la resolución de un problema de partición con restricciones: MULTI_HEUR_PART (sección 5.7), que resuelve problemas de partición combinados, y OPWK_PART (sección 5.8), que se basa en modificar las estaciones mediante intercambio y transferencia de tareas.

En la sección 5.9 se realiza una experiencia computacional comparando HEUR_PART con la adición de la regla R16, MULTI_HEUR_PART y OPWK_PART. Los resultados muestran que OPWK_PART es el procedimiento que obtiene los mejores resultados. Además, se comprueba que OPWK_PART obtiene mejores resultados que cualquier otro procedimiento de resolución que se haya publicado en la literatura para esta variante hasta el momento, aunque el objetivo a optimizar no es exactamente el mismo. Finalmente se compara con GanttSA (sección 5.9.2.4), que considera que los tiempos de las tareas son independientes del número de trabajadores en la estación, y OPWK_PART obtiene mejores resultados, aunque ésta no se haya diseñado considerando, específicamente, tiempos de tarea deterministas.

6. Procedimientos de resolución para el SMALBP-DTT2

En esta tesis doctoral se presentan distintos procedimientos para resolver la variante SMALBP-2 con tiempos de proceso dependientes (SMALBP-DTT2). Recuérdese que en el SMALBP-DTT2 se fija una cota superior del número de estaciones de la solución final (UB^M). El objetivo principal es minimizar el tiempo de ciclo y el secundario es minimizar el número de trabajadores.

En primer lugar, se formula un modelo matemático (MILP-DTT2; sección 6.1). A continuación, se presentan dos procedimientos heurísticos basados en la resolución de un problema de partición con restricciones: HEUR_CT_PART (sección 6.2), que parte de una solución SALBP-2 a partir de la cual se generan uniones de estaciones; y OPCT_PART (sección 6.3), que parte de una solución SMALBP-DTT2 y genera nuevas soluciones mediante intercambio y transferencia de tareas. Además, se propone un procedimiento heurístico constructivo *greedy*, denominado GRDY_DTT2 (sección 6.3.1), que además se utiliza para generar la solución inicial SMALBP-DTT2 de OPCT_PART.

La eficacia de estos cuatro procedimientos (MILP-DTT2, HEUR_CT_PART, OPCT_PART y GRDY_DTT2) se evalúa en la experiencia computacional presentada en la sección 6.4.

En la literatura no hay ningún otro procedimiento que aborde el SMALBP-DTT2. Sin embargo, Roshani y Giglio (2017) resuelven SMALBP-2 con los mismos objetivos, utilizando dos algoritmos basados en SA: DSA e ISA. De todos los procedimientos propuestos en la literatura para el SMALBP-2, los procedimientos de Roshani y Giglio (2017) son los que mejores resultados han obtenido hasta el momento. En la sección 6.4.2.2, DSA e ISA se comparan con OPCT_PART, el procedimiento que presenta los mejores resultados entre todos los que han sido propuestos en esta tesis para el SMALBP-DTT2.

6.1. Modelo matemático (MILP-DTT2)

Actualmente, en la literatura no han sido formulados modelos matemáticos para el SMALBP-DTT2. En esta sección se formula por primera vez un modelo matemático para el SMALBP-DTT2, denominado "MILP-DTT2".

Los datos, las variables, la función objetivo y las restricciones se presentan en las secciones 6.1.1-6.1.4, respectivamente. La experiencia computacional del MILP-DTT2 se detalla en la sección 6.4.

6.1.1. Datos

- UB^M cota superior del número de estaciones; $j = 1, \dots, UB^M$.
- N número de tareas ($i = 1, \dots, N$). A diferencia del modelo MILP-DTT1 (sección 5.1), en éste la tarea N es una tarea real.
- UB^O cota superior del número de operarios por estación; $o = 1, \dots, UB^O$.
- P_{io} tiempo de proceso de la tarea i si hay o operarios en la estación en la que es procesada; $i = 1, \dots, N$, $o = 1, \dots, UB^O$ (siendo $P_{No} = 0 \forall o$).
- IPR_i conjunto de predecesoras inmediatas de la tarea i ; $i = 1, \dots, N$.
- APR_i conjunto de todas las predecesoras de la tarea i : $APR_i = IPR_i \cup \left(\bigcup_{h \in IPR_i} APR_h \right)$;
 $i = 1, \dots, N$.
- \overline{PR} conjunto de parejas de tareas que no tienen relaciones de precedencia entre ellas: $\overline{PR} = \{(h \in \{1, \dots, N-1\}, i \in \{h+1, \dots, N\}) \mid h \notin APR_i \wedge i \notin APR_h\}$.

6.1.2. Variables

- CT tiempo de ciclo.
- $t_{ij} \in \mathbb{R}^+$ tiempo de proceso de la tarea i si es asignada a la estación j ;
 $i = 1, \dots, N$; $j = 1, \dots, UB^M$.
- $x_{ijk} \in \{0,1\}$ 1 si la tarea i es asignada al operario k de la estación j , 0 en caso contrario:
 $i = 1, \dots, N$; $j = 1, \dots, UB^M$; $k = 1, \dots, UB^O$.
- $y_{jo} \in \{0,1\}$ 1 si hay o operarios en la estación j , 0 en caso contrario:
 $j = 1, \dots, UB^M$; $o = 1, \dots, UB^O$.
- $w_{jk} \in \{0,1\}$ 1 si se usa el operario k de la estación j , 0 en caso contrario:
 $j = 1, \dots, UB^M$; $k = 1, \dots, UB^O$.
- $st_i \in \mathbb{R}^+$ tiempo de inicio de la tarea i : $i = 1, \dots, N$. Este tiempo es local para cada estación, lo que significa que cada vez que una unidad de producto llega a una estación se considera como tiempo 0.
- $\theta_{hi} \in \{0,1\}$ variable auxiliar para asegurar que las tareas h y i no son realizadas de

manera simultánea por el mismo operador: $(h, i) \in \overline{PR}$.

6.1.3. Función objetivo

La función objetivo minimiza el tiempo de ciclo como primer objetivo y el número de trabajadores como segundo. El peso $\frac{1}{\min(UB^M \cdot UB^O + 1, N + 1)}$ garantiza que el segundo término, que representa el número de trabajadores, sea el criterio de desempate entre soluciones de un mismo tiempo de ciclo.

$$[\text{MIN}] CT + \frac{1}{\min(UB^M \cdot UB^O + 1, N + 1)} \cdot \sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} w_{jk} \quad (\text{Ec. 52})$$

6.1.4. Restricciones

$$\sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} x_{ijk} = 1 \quad i = 1, \dots, N \quad (\text{Ec. 53})$$

$$st_i + t_{ij} \leq CT + (P_{i,UB^O} - P_{i,1}) \cdot \left(1 - \sum_{k=1}^{UB^O} x_{ijk}\right) \quad i = 1, \dots, N; \quad (\text{Ec. 54})$$

$$j = 1, \dots, UB^M$$

$$\sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} j \cdot x_{hjk} \leq \sum_{j=1}^{UB^M} \sum_{k=1}^{UB^O} j \cdot x_{ijk} \quad i = 1, \dots, N; \quad (\text{Ec. 55})$$

$$h \in IPR_i$$

$$st_i \geq st_h + t_{hj} - \left(\sum_{i=1}^N P_{i,UB^O}\right) \cdot \left(2 - \sum_{k=1}^{UB^O} x_{hjk} - \sum_{k=1}^{UB^O} x_{ijk}\right) \quad i = 1, \dots, N; \quad (\text{Ec. 56})$$

$$h \in IPR_i;$$

$$j = 1, \dots, UB^M;$$

$$st_i \geq st_h + t_{hj} - \left(\sum_{i=1}^N P_{i,UB^O}\right) \cdot (3 - \theta_{hi} - x_{hjk} - x_{ijk}) \quad (h, i) \in \overline{PR}; \quad (\text{Ec. 57})$$

$$k = 1, \dots, UB^O;$$

$$j = 1, \dots, UB^M;$$

$$st_h \geq st_i + t_{ij} - \left(\sum_{i=1}^N P_{i,UB^O}\right) \cdot (3 - (1 - \theta_{hi}) - x_{hjk} - x_{ijk}) \quad (h, i) \in \overline{PR}; \quad (\text{Ec. 58})$$

$$k = 1, \dots, UB^O$$

$$j = 1, \dots, UB^M;$$

$$\sum_{i=1}^N x_{ijk} \leq N \cdot w_{jk} \quad j = 1, \dots, UB^M; \quad (\text{Ec. 59})$$

$$k = 1, \dots, UB^O$$

$$\sum_{o=1}^{UB^0} o \cdot y_{jo} = \sum_{k=1}^{UB^0} w_{jk} \quad j = 1, \dots, UB^M \quad (\text{Ec. 60})$$

$$w_{jk} \leq w_{j,k-1} \quad \begin{array}{l} j = 1, \dots, UB^M; \\ k = 2, \dots, UB^0 \end{array} \quad (\text{Ec. 61})$$

$$t_{ij} = \sum_{o=1}^{UB^0} P_{io} \cdot y_{jo} \quad \begin{array}{l} i = 1, \dots, N; \\ j = 1, \dots, UB^M \end{array} \quad (\text{Ec. 62})$$

$$\sum_{o=1}^{UB^0} y_{jo} \leq 1 \quad j = 1, \dots, UB^M \quad (\text{Ec. 63})$$

$$\sum_{i=1}^N x_{ijk} \geq w_{jk} \quad \begin{array}{l} j = 1, \dots, UB^M; \\ k = 1, \dots, UB^0 \end{array} \quad (\text{Ec. 64})$$

$$\sum_{i=1}^N x_{ijk} \leq \sum_{i=1}^N x_{ij,k-1} \quad \begin{array}{l} j = 1, \dots, UB^M; \\ k = 2, \dots, UB^0 \end{array} \quad (\text{Ec. 65})$$

$$\sum_{o=1}^{UB^0} y_{jo} \leq \sum_{o=1}^{UB^0} y_{j-1,o} \quad j = 2, \dots, UB^M \quad (\text{Ec. 66})$$

Las restricciones (53) garantizan que cada tarea se asigne a un solo trabajador en una estación. Las restricciones (54) aseguran que la carga de trabajo de cada estación no supere el tiempo de ciclo. Las restricciones (55) y (56) garantizan que se cumplan las relaciones de precedencia. Las restricciones (57) y (58) establecen que si dos tareas han sido asignadas al mismo trabajador no puedan ser procesadas simultáneamente. Las restricciones (59) impiden asignar tareas a trabajadores que no estén asignados a una estación. Las restricciones (60) establecen una relación entre w_{jk} y y_{jo} . Las restricciones (61) hacen que los trabajadores, en cada estación, sean utilizados en orden creciente de sus índices. Las restricciones (62) fijan el tiempo de proceso de cada tarea en función del número de trabajadores de la estación. Las restricciones (63) garantizan la congruencia de las variables y_{jo} . Las restricciones (64) evitan que algunas soluciones utilicen trabajadores sin asignarles tareas. Las restricciones (65) rompen simetrías al garantizar que el número de tareas asignadas a un trabajador no sea superior al número de tareas asignadas a un trabajador con un índice k inferior; y las restricciones (66) rompen simetrías al evitar llenar una estación si hay una vacía con un índice j inferior.

6.2. Heurística HEUR_CT_PART

En esta sección se presenta HEUR_CT_PART, un procedimiento heurístico basado en la resolución de un problema de partición con restricciones. La experiencia computacional de HEUR_CT_PART se detalla en la sección 6.4.

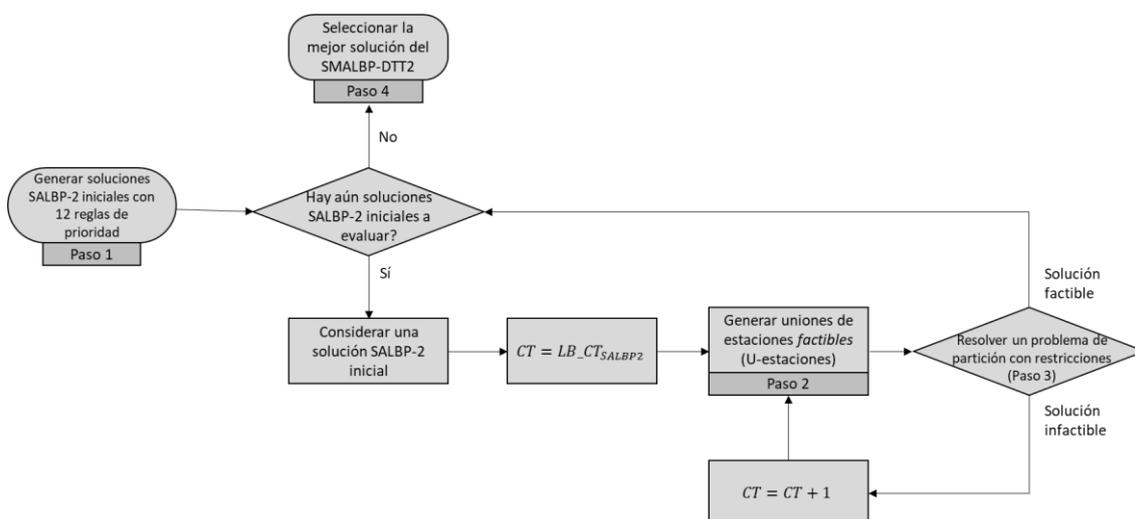


Figura 15. Diagrama de flujo del procedimiento HEUR_CT_PART. Fuente: propia

La Figura 15 presenta el diagrama de flujo de HEUR_CT_PART, que consta de los siguientes pasos que se describirán en las siguientes subsecciones:

Paso 1. Generar un conjunto de soluciones iniciales resolviendo SALBP-2 con 12 procedimientos heurísticos (sección 6.2.1). Para cada procedimiento heurístico, el SALBP-1 se resuelve de forma iterativa. La primera iteración utiliza, como tiempo de ciclo, el valor de una cota inferior (LB_{CT}). La heurística termina si el número de estaciones encontradas no es mayor que el producto de la cota superior del número de estaciones (UB^M) por la cota superior del número de trabajadores en cada estación MALBP (UB^0). En caso contrario, el tiempo de ciclo se incrementa en una unidad y se realiza una nueva iteración. Se hace referencia al tiempo de ciclo de cada solución SALBP-2 inicial como LB_CT_{SALBP2} .

Para cada una de las soluciones del paso 1, se aplican los pasos 2 y 3.

Paso 2. Considerando el tiempo de ciclo encontrado en el Paso 1 para una solución SALBP-2 ($CT = LB_CT_{SALBP2}$), evaluar todas las posibles uniones de estaciones entre ellas y seleccionar sólo aquellas uniones que sean factibles (sección 6.2.2). Una unión de estaciones puede concebirse como una estación única en la que se procesan todas sus tareas. La estación resultante de esta unión de estaciones,

también se denominará a partir de ahora U-estación. Cada U-estación se considera factible si se respetan las relaciones de precedencia (sección 6.2.2.1) y si el número de trabajadores necesarios en la U-estación no es mayor a la cota superior UB^O (sección 6.2.2.2).

Paso 3. Resolver un problema de partición con restricciones, utilizando programación matemática, con todas las U-estaciones factibles encontradas en el paso anterior (apartado 6.2.3). El modelo matemático considera un máximo de UB^M U-estaciones seleccionadas y genera una solución que minimiza el tiempo de ciclo como objetivo principal y, como objetivo secundario, el número de trabajadores. En caso de no encontrar una solución factible para el problema de partición con restricciones, el tiempo de ciclo se incrementa en una unidad ($CT = CT + 1$) y el proceso vuelve al Paso 2.

Paso 4. La solución con el menor tiempo de ciclo y, en caso de empate, con el menor número de trabajadores, se selecciona de entre todas las soluciones encontradas en el Paso 3.

Para ilustrar el funcionamiento de este procedimiento heurístico, en las secciones 6.2.1, 6.2.2 y 6.2.3 se utiliza el ejemplo ilustrativo presentado en la Figura 16 (se considera un máximo de $UB^O = 3$ operarios por estación y se fija una cota superior del número de estaciones de $UB^M = 3$). La Figura 16 muestra el grafo de precedencias entre tareas y, delimitado por los símbolos “<>”, el tiempo necesario para realizar cada tarea según el número de operarios (de 1 a 3) en la estación.

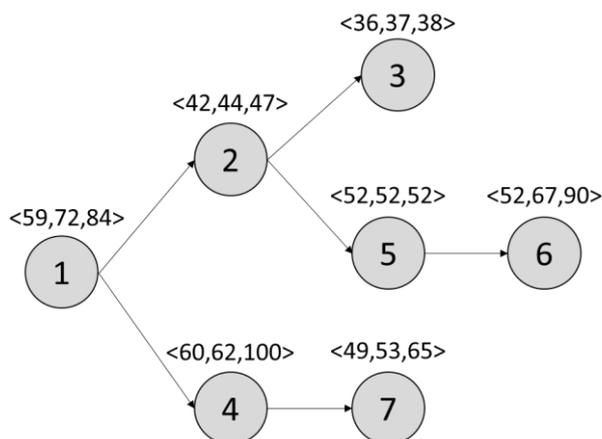


Figura 16. Grafo de precedencias y tiempos de proceso del ejemplo ilustrativo del SMALBP-DDT2. Fuente: propia

6.2.1. Generación de soluciones iniciales del SALBP-2

Para generar una solución inicial SALBP-2 se resuelve iterativamente el SALBP-1 para varios tiempos de ciclo, con el fin de determinar el menor tiempo de ciclo para el que se obtiene una solución factible con un número máximo de estaciones. En particular, el número de estaciones de la solución SALBP-2 no debe ser mayor al producto de la cota superior del número de estaciones MALBP (UB^M) por la cota superior del número de trabajadores en cada estación MALBP (UB^O).

La primera iteración resuelve SALBP-1 utilizando una cota inferior de tiempo de ciclo (LB_{CT}), que se calcula a través de la Ec. 67.

$$LB_{CT} = \max \left(\max_{i=1, \dots, N} P_{i,1}, \frac{\sum_{i=1}^N P_{i,1}}{UB^M \cdot UB^O} \right) \quad (\text{Ec. 67})$$

Para resolver el SALBP-1 con el tiempo de ciclo dado, se utiliza el mismo procedimiento de la sección 5.3.1 que, recuérdese, consistía en un procedimiento constructivo greedy orientado a estaciones (Scholl y Voß, 1996). En primer lugar, se abre una estación. Iterativamente, se asigna la mejor tarea candidata (siguiendo una regla de prioridad) a la estación abierta en ese momento. Una tarea es candidata si: no ha sido asignada previamente a una estación, todas sus tareas precedentes han sido asignadas y puede ser ejecutada sin exceder el tiempo de ciclo. En caso de que no haya más tareas candidatas disponibles, pero aún queden tareas por asignar, se cierra la estación abierta y se abre una nueva estación. El proceso finaliza cuando se han asignado todas las tareas.

En caso de que el número de estaciones encontradas resolviendo SALBP-1 no sea mayor que el producto de UB^M por UB^O , el procedimiento finaliza (se ha encontrado una solución inicial para SALBP-2). En caso contrario, el tiempo de ciclo se incrementa en una unidad y se resuelve de nuevo el SALBP-1. Las soluciones SALBP-2 iniciales aquí obtenidas pueden tener un número de estaciones mayor a la cota superior (UB^M) y, así ser infactibles, pero podrán convertirse en soluciones factibles uniendo sus estaciones, como se expone en los pasos siguientes.

Se generan 12 soluciones iniciales, resultantes de utilizar las 12 reglas de prioridad presentadas en la Tabla 4. De entre todas las soluciones iniciales, no se considera más de una solución entre las equivalentes (dos soluciones son equivalentes si la única diferencia entre ellas es el orden de las estaciones).

Se toma, como ejemplo para los siguientes apartados, la solución obtenida utilizando la octava regla de prioridad para el ejemplo ilustrativo (Figura 16). Considerando un $LB_{CT} = 60$, la primera iteración obtiene una solución (mostrada en la Figura 17) con 7 estaciones, que no supera el producto de $UB^M = 3$ por $UB^O = 3$.

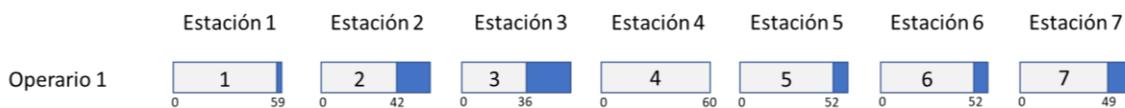
Solución usando la 8ª regla de prioridad

Figura 17. Solución SALBP-2 del ejemplo ilustrativo con la octava regla de prioridad. Fuente: propia

6.2.2. Generación de uniones de estaciones (U-estaciones)

El siguiente paso consiste en generar uniones de estaciones, las llamadas U-estaciones, que no excedan la cota superior del número de trabajadores en cada estación (UB^0), dado un tiempo de ciclo. De todas las posibles uniones entre 2 y UB^0 estaciones de la solución SALBP-2 (Paso 1), el procedimiento HEUR_CT_PART considera sólo aquellas que son factibles en términos de relaciones de precedencia (sección 6.2.2.1).

Como se indicó anteriormente, inicialmente se trabaja con el tiempo de ciclo hallado en el Paso 1 para la solución SALBP-2 ($CT = LB_{CT_{SALBP2}}$). Para cada unión candidata (que sea factible en términos de precedencia; sección 6.2.2.1), se utiliza un modelo matemático para evaluar la posibilidad de unir las estaciones con el menor número de trabajadores (sección 6.2.2.2). Si el procedimiento encuentra una solución factible para el tiempo de ciclo dado sin superar los UB^0 trabajadores, dicha U-estación se incluye en la lista de U-estaciones candidatas a formar parte de la solución final.

Con todas las U-estaciones factibles encontradas, HEUR_CT_PART resuelve un problema de partición con restricciones (Paso 3; explicado en la sección 6.2.3). Si el problema de partición no encuentra una solución factible con el tiempo de ciclo dado, dicho tiempo de ciclo se incrementa en una unidad ($CT = CT + 1$) y nuevamente se vuelve a este Paso 2 para generar más U-estaciones (que no eran factibles con el tiempo de ciclo anterior). Aquellas U-estaciones que ya eran factibles con un tiempo de ciclo menor se mantienen en la lista de candidatas para formar parte de la solución final (a calcular en el Paso 3).

6.2.2.1. Factibilidad de la unión de estaciones según precedencias

Una U-estación no es factible si en una sola estación (de las no incluidas en esta unión) hay, al mismo tiempo, tareas sucesoras y predecesoras con respecto a las tareas de la U-estación evaluada. Es decir, una U-estación es factible cuando no hay ninguna otra estación (fuera de la unión evaluada) con tareas sucesoras y predecesoras con respecto a la U-estación evaluada. Para consultar un ejemplo, véase sección 5.3.2.1.

6.2.2.2. Factibilidad de la unión de estaciones sin superar la concentración máxima de trabajadores

Para las U-estaciones factibles según precedencias, se ejecuta iterativamente un modelo matemático que resuelve un SMALBP con una estación, un tiempo de ciclo fijo y un número dado de trabajadores. El modelo matemático utilizado es el presentado en la sección 5.3.2.2. El proceso iterativo consiste en resolver el modelo matemático aumentando el número de trabajadores (NW) en una unidad hasta encontrar una solución factible o hasta alcanzar la cota superior del número de trabajadores en cada estación (UB^0).

El número de trabajadores NW de la primera iteración es igual a su cota inferior (para más detalles ver sección 5.3.2.2). Si el modelo matemático no encuentra una solución factible, el número de trabajadores NW se incrementa en una unidad y el modelo se resuelve de nuevo. El número de trabajadores NW no debe ser mayor a la cota superior del número de trabajadores en cada estación (UB^0). Solo si el procedimiento encuentra una solución factible, la U-estación se incluye en la lista de U-estaciones candidatas a formar parte de la solución final. Para la resolución de los modelos matemáticos se establece un límite de tiempo de computación, de β_3 segundos; el valor β_3 utilizado para los experimentos computacionales se detalla en la sección 6.4.

Para la solución obtenida considerando la octava regla de prioridad (Figura 17), la Tabla 32 muestra, para cada tiempo de ciclo, la lista de U-estaciones que son factibles en términos de relaciones de precedencia y que encuentran una solución factible sin superar los $UB^0 = 3$ trabajadores. Recuérdese que, para cada tiempo de ciclo, hasta que se consiga una solución factible dada una solución SALBP-2 inicial, se ejecutan secuencialmente la generación de U-estaciones (Paso 2) y la resolución de un problema de partición (Paso 3). La Tabla 32 también informa, en la última columna, sobre si el problema de partición (Paso 3) encuentra una solución factible para cada tiempo de ciclo. Por ejemplo, para $CT = 60$, se generan 4 U-estaciones factibles. Una de ellas es la unión de las estaciones 2 y 7, creando una U-estación con dos trabajadores que procesan las tareas 2 y 7. Para $CT = 61$, no se encuentran nuevas U-estaciones factibles. Para $CT = 62$, se obtienen 3 nuevas U-estaciones factibles (además de las 4 U-estaciones anteriores encontradas con $CT = 60$); pero no se encuentra ninguna solución factible de SMALBP-DTT2. En la sección 6.2.3 se proporcionan más detalles sobre la solución del problema de partición con $CT = 96$.

Tiempo de ciclo (CT)	U-estaciones factibles	Tareas incluidas en la U-estación	Número de trabajadores en la U-estación	¿Solución factible del problema de partición con restricciones? (Paso 3)
60	Estaciones 2 + 7	2 + 7	2	No
	Estaciones 3 + 5	3 + 5	2	
	Estaciones 3 + 7	3 + 7	2	
	Estaciones 5 + 7	5 + 7	2	

Tiempo de ciclo (CT)	U-estaciones factibles	Tareas incluidas en la U-estación	Número de trabajadores en la U-estación	¿Solución factible del problema de partición con restricciones? (Paso 3)
62	Estaciones 2 + 4	2 + 4	2	No
	Estaciones 3 + 4	3 + 4	2	
	Estaciones 4 + 5	4 + 5	2	
65	Estaciones 3 + 5 + 7	3 + 5 + 7	3	No
67	Estaciones 3 + 6	3 + 6	2	No
	Estaciones 4 + 6	4 + 6	2	
	Estaciones 6 + 7	6 + 7	2	
78	Estaciones 2 + 3	2 + 3	1	No
81	Estaciones 2 + 3 + 4	2 + 3 + 4	2	No
	Estaciones 2 + 3 + 7	2 + 3 + 7	2	
89	Estaciones 3 + 4 + 5	3 + 4 + 5	2	No
90	Estaciones 3 + 6 + 7	3 + 6 + 7	2	No
94	Estaciones 2 + 5	2 + 5	1	No
96	Estaciones 2 + 3 + 5	2 + 3 + 5	2	Sí
	Estaciones 2 + 4 + 5	2 + 4 + 5	2	
	Estaciones 2 + 5 + 7	2 + 5 + 7	2	

Tabla 32. U-estaciones factibles que son candidatas a estar en la solución final de HEUR_CT_PART en función del tiempo de ciclo. Fuente: propia

6.2.3. Resolución de un problema de partición con restricciones

Con la lista de U-estaciones candidatas para un tiempo de ciclo determinado, incluyendo además como U-estaciones candidatas cada una de las estaciones de la solución inicial del SALBP-2 de forma individual, se propone un modelo matemático para resolver un problema de partición con restricciones entre U-estaciones candidatas. Si el problema de partición con restricciones encuentra una solución factible, el Paso 3 finaliza. En caso contrario, el tiempo de ciclo se incrementa en una unidad ($CT = CT + 1$) y el procedimiento vuelve al Paso 2 para intentar generar U-estaciones candidatas adicionales.

A continuación, se presenta el modelo matemático para resolver un problema de partición con restricciones. El modelo busca una solución que minimice el tiempo de ciclo como objetivo principal y el número de trabajadores como objetivo secundario.

Datos

- UB^M cota superior del número de U-estaciones que se pueden seleccionar.
- J número de estaciones obtenidas en la resolución del SALBP-2 ($j = 1, \dots, J$).
- S número de U-estaciones candidatas ($s = 1, \dots, S$).
- CT_s tiempo de ciclo de la U-estación s ($s = 1, \dots, S$).
- NW_s número de trabajadores en la U-estación s ($s = 1, \dots, S$).
- IS_{sj} 1 si en la U-estación s está incluida la estación j , 0 en caso contrario:

$$j = 1, \dots, J; s = 1, \dots, S.$$

\overline{INC} grupos de U-estaciones que no tienen estaciones en común pero que son infactibles por razones de precedencia.

R número de U-estaciones en cada grupo de \overline{INC} ($R = 2, \dots, J$).

$$\overline{INC} = \{(r_1, \dots, r_R) \mid r_1 \in \{1, \dots, S - (R - 1)\}, \dots, r_R \in \{r_1 + (R - 1), \dots, S\}, r_1, \dots, r_R \text{ infactibles}\}.$$

Dos U-estaciones ($R = 2$) son infactibles si en una de las U-estaciones hay, al mismo tiempo, tareas sucesoras y predecesoras con respecto a las tareas de la otra U-estación. Tres o más U-estaciones ($R = 3, \dots, J$) son infactibles si cada U-estación tiene relaciones de precedencia con sólo otras dos U-estaciones (más concretamente, tiene tareas predecesoras con respecto a las tareas de una U-estación y tareas sucesoras con respecto a las tareas de la otra U-estación) y no hay infactibilidades en ninguna combinación de $2, \dots, R - 1$ U-estaciones.

Variables

$x_s \in \{0,1\}$ 1 si se usa la U-estación candidata s , 0 en caso contrario: $s = 1, \dots, S$.

CT_{max} tiempo de ciclo máximo entre las U-estaciones seleccionadas.

Función objetivo

La función objetivo minimiza el tiempo de ciclo como primer objetivo y el número de trabajadores como objetivo secundario:

$$[MIN] CT_{max} + \frac{1}{\max_{s=1, \dots, S} (NW_s) \cdot UB^M + 1} \cdot \sum_{s=1}^S NW_s \cdot x_s \quad (\text{Ec. 68})$$

Restricciones

$$\sum_{s=1}^S IS_{sj} \cdot x_s = 1 \quad j = 1, \dots, J \quad (\text{Ec. 69})$$

$$\sum_{s=r_1}^{r_R} x_s \leq R - 1 \quad (r_1, \dots, r_R) \in \overline{INC} \quad (\text{Ec. 70})$$

$$CT_{max} \geq CT_s \cdot x_s \quad s = 1, \dots, S \quad (\text{Ec. 71})$$

$$\sum_{s=1}^S x_s \leq UB^M \quad (\text{Ec. 72})$$

Las restricciones (69) garantizan que la solución final incluya cada una de las estaciones obtenidas en la resolución del SALBP-2, una sola vez. Las restricciones (70) impiden asignar U-estaciones infactibles por razones de precedencia. Las restricciones (71) obtienen el valor de la variable CT_{max} como el tiempo de ciclo máximo de las U-estaciones seleccionadas. La restricción (72) garantiza que el número de U-estaciones seleccionadas no es mayor que la cota superior UB^M .

Para el ejemplo ilustrativo (Tabla 32), el modelo matemático se resuelve inicialmente con $CT = 60$, considerando 11 U-estaciones (7 U-estaciones de la solución inicial SALBP-2 y las 4 U-estaciones encontradas en el Paso 2 para $CT = 60$) y no encuentra una solución factible. La segunda iteración, con $CT = 61$, no genera U-estaciones adicionales. La tercera iteración, con $CT = 62$, considera 14 U-estaciones (las 11 U-estaciones anteriores más las 3 nuevas encontradas en el Paso 2 para $CT = 62$) pero el modelo matemático anterior tampoco obtiene una solución factible. El procedimiento continúa aumentando el tiempo de ciclo hasta $CT = 96$. Para $CT = 96$, con 27 U-estaciones (7 U-estaciones de la solución inicial SALBP-2 y 20 U-estaciones encontradas en el Paso 2, incluidas en la Tabla 32), se obtiene una solución factible con el modelo matemático anterior. La solución obtenida utiliza 5 trabajadores y 3 U-estaciones (Figura 18): tarea 1 (estación 1), tareas 2, 4 y 5 (unión de las estaciones 2, 4 y 5) y tareas 3, 6 y 7 (unión de las estaciones 3, 6 y 7).



Figura 18. Solución óptima para el ejemplo ilustrativo que resuelve el SMALBP-DTT2. Fuente: propia

6.3. Heurística OPCT_PART

En esta sección se presenta OPCT_PART, un procedimiento heurístico que resuelve el SMALBP-DTT2. OPCT_PART consiste en modificar las estaciones de una solución inicial SMALBP-DTT2, generando intercambios y transferencias de tareas entre estaciones, y en resolver un problema de partición con restricciones. Las soluciones iniciales SMALBP-DTT2 se generan utilizando un procedimiento heurístico constructivo *greedy* denominado "GRDY_DTT2". Para ejemplificar el funcionamiento de este procedimiento, el ejemplo

ilustrativo de la Figura 16 también se utiliza en las secciones 6.3.1, 6.3.2 y 6.3.3. La experiencia computacional de OPCT_PART se detalla en la sección 6.4.

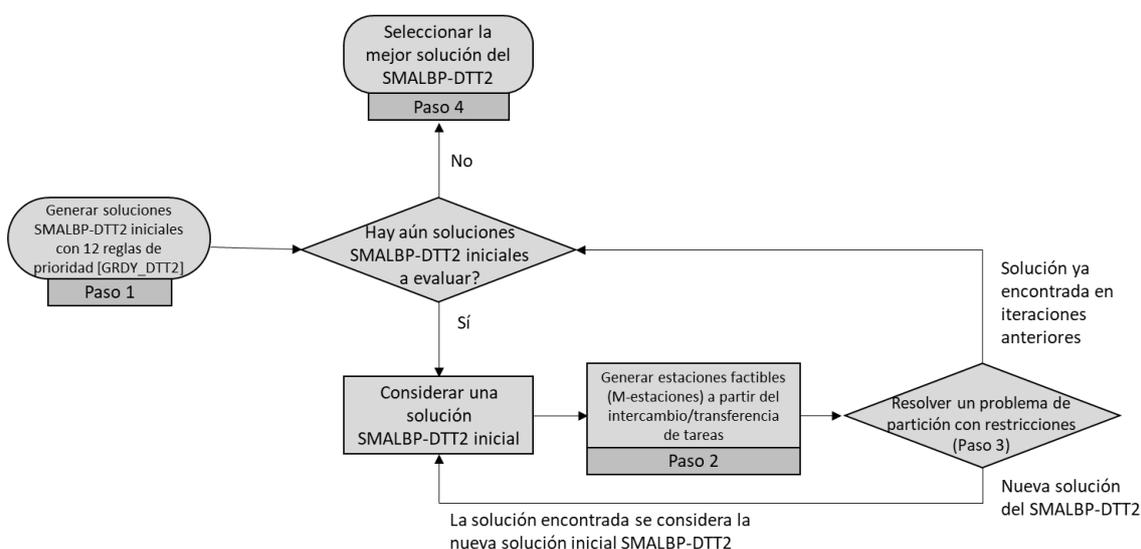


Figura 19. Diagrama de flujo del procedimiento OPCT_PART. Fuente: propia

El diagrama de flujo de OPCT_PART se presenta en la Figura 19. El procedimiento consta de los siguientes pasos que se detallan en las siguientes subsecciones:

Paso 1. Generar un conjunto de soluciones iniciales del SMALBP-DDT2 con la heurística GRDY_DTT2 (sección 6.3.1). Dado un conjunto de reglas de prioridad, para cada una de ellas GRDY_DTT2 resuelve iterativamente SMALBP-DDT1 considerando una cota superior en el número de trabajadores en cada estación (UB^0).

Para cada una de las soluciones del paso anterior, se aplican los pasos 2 y 3.

Paso 2. Evaluar todos los posibles intercambios y transferencias de tareas entre las estaciones de la solución SMALBP-DDT2 en curso, y seleccionar únicamente las estaciones modificadas que sean factibles (sección 6.3.2). En concreto, se evalúan cuatro tipos de modificaciones: intercambio de una sola tarea, transferencia de una sola tarea, intercambio de una sola tarea seguido de transferencia de una sola tarea y transferencia de una sola tarea seguida de intercambio de una sola tarea. Sean las estaciones j_1 y j_2 (sin tareas en común), un intercambio/transferencia de tareas entre esas dos estaciones (generando las estaciones j_1' y j_2') es factible si se respetan las relaciones de precedencia en j_1' y j_2' (sección 6.3.2.1) y el tiempo de ciclo no aumenta (es decir, el tiempo de ciclo máximo entre j_1' y j_2' no es superior al tiempo de ciclo máximo entre j_1 y j_2 ; sección 6.3.2.2). Para evitar ambigüedades, una estación modificada se denominará a

partir de ahora M-estación.

Paso 3. Resolver un problema de partición con restricciones mediante programación matemática con todas las M-estaciones factibles encontradas en el paso anterior (sección 6.3.3). El modelo matemático considera un máximo de UB^M M-estaciones seleccionadas, con la condición adicional de que cada tarea debe incluirse en una y sólo una M-estación. El modelo matemático minimiza el tiempo de ciclo como objetivo principal y el número de trabajadores como objetivo secundario. En caso de empate, se selecciona la solución que utilice el menor número de M-estaciones ya incluidas en las soluciones anteriores. Si la solución encontrada mediante el problema de partición con restricciones es diferente de todas las soluciones anteriores obtenidas, dicha solución se considera la nueva solución en curso de SMALBP-DTT2 y el proceso vuelve al Paso 2. En caso contrario, se considera la siguiente solución del Paso 1 y el proceso vuelve al Paso 2.

Paso 4. Se selecciona la solución con el menor tiempo de ciclo de entre todas las soluciones encontradas en el Paso 3. En caso de empate, se selecciona la solución con el menor número de trabajadores.

6.3.1. Generación de soluciones iniciales del SMALBP-DTT2 (GRDY_DTT2)

Las soluciones iniciales del SMALBP-DTT2 se generan utilizando GRDY_DTT2, un procedimiento heurístico constructivo *greedy* propuesto en esta tesis doctoral para resolver el SMALBP-DTT2.

GRDY_DTT2 se basa en resolver iterativamente el SMALBP-DTT1 para varios tiempos de ciclo, con el fin de determinar el tiempo de ciclo más pequeño para el que existe una solución factible con un número de estaciones igual o inferior a la cota superior UB^M . La cota inferior del tiempo de ciclo (LB_{CT}) utilizada en la primera iteración del SMALBP-DTT1 se calcula mediante la Ec. 67. Si el número de estaciones encontradas resolviendo SMALBP-DTT1 no es superior a UB^M , el procedimiento finaliza (se ha encontrado una solución inicial SMALBP-DTT2). En caso contrario, el tiempo de ciclo se incrementa en una unidad y se vuelve a resolver el SMALBP-DTT1.

Para resolver el SMALBP-DTT1 con un tiempo de ciclo dado, se propone un procedimiento constructivo *greedy* orientado a estaciones denominado "GRDY_DTT1". En primer lugar, se abre una estación. A continuación, el procedimiento calcula, mediante el procedimiento de asignación que se describe en el párrafo siguiente, cuántas tareas pueden asignarse a dicha estación para cada número posible de trabajadores (es decir, desde un trabajador hasta UB^0) y selecciona el menor número de trabajadores que pueda procesar el mayor

número de tareas. En caso de que aún queden tareas por asignar, se abre una nueva estación. El procedimiento constructivo *greedy* finaliza cuando se han asignado todas las tareas. Para evaluar la eficacia del nuevo procedimiento diseñado para resolver el SMALBP-DTT1, en el Apéndice 5 GRDY_DTT1 se compara con OPWK_PART, teniendo en cuenta que los objetivos son diferentes, y con Max_S.

Considerando un número fijo de trabajadores (que determinan los tiempos de proceso de las tareas), el procedimiento para calcular cuántas y qué tareas pueden asignarse a una estación consiste en asignar iterativamente la mejor tarea candidata (siguiendo una regla de prioridad) al trabajador que pueda procesarla antes (considerando que deben respetarse las relaciones de precedencia). En caso de empate, la tarea se asigna al trabajador con menor tiempo ocioso entre su tarea anterior y la tarea candidata. Una tarea es candidata si no ha sido asignada previamente, todas sus tareas precedentes han sido asignadas y puede ser ejecutada sin exceder el tiempo de ciclo. El proceso de asignación finaliza cuando se han asignado todas las tareas.

Las 12 reglas de prioridad para la ordenación de las tareas presentadas en Scholl y Voß (1996) también se utilizan en este procedimiento, de forma que se obtienen 12 soluciones iniciales de SMALBP-DTT2. Estas reglas de prioridad se muestran en la Tabla 4. Nuevamente, no se toma en consideración más de una solución entre las equivalentes (cuya única diferencia está en el orden de las estaciones).

Considérese el ejemplo ilustrativo (Figura 16), utilizando la octava regla de prioridad y con un valor $LB_{CT} = 60$. La Tabla 33 muestra cómo se resuelve iterativamente el SMALBP-DTT1 hasta encontrar una solución con no más de $UB^M = 3$ estaciones. En este caso, con $CT = 100$, se obtiene una solución SMALBP-DTT2 con 3 estaciones (Figura 20).

Tiempo de ciclo (CT)	Solución SMALBP-DTT2							¿Solución con no más de $UB^M = 3$?
		Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	
60-61	Operario 1	1	2	3	4	6	7	No, 6 estaciones encontradas
	Operario 2			5				
	Operario 3							
62-64	Operario 1	1	2	3	6	7		No, 5 estaciones encontradas
	Operario 2		4	5				
	Operario 3							
65-77	Operario 1	1	2	3	6			No, 4 estaciones encontradas
	Operario 2		4	5				
	Operario 3			7				
78-80	Operario 1	1	2, 3	4	6			No, 4 estaciones encontradas
	Operario 2			5	7			
	Operario 3							
81-99	Operario 1	1	2, 3	5	6			No, 4 estaciones encontradas
	Operario 2		4	7				
	Operario 3							
100	Operario 1	1	2, 3	6				Sí, 3 estaciones encontradas

Tiempo de ciclo (CT)	Solución SMALBP-DTT2						¿Solución con no más de $UB^M = 3$?
	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	
Operario 2		4	7				
Operario 3		5					

Tabla 33. Soluciones SMALBP-DTT2 encontradas aplicando el primer paso del procedimiento OPCT_PART.

Fuente: propia

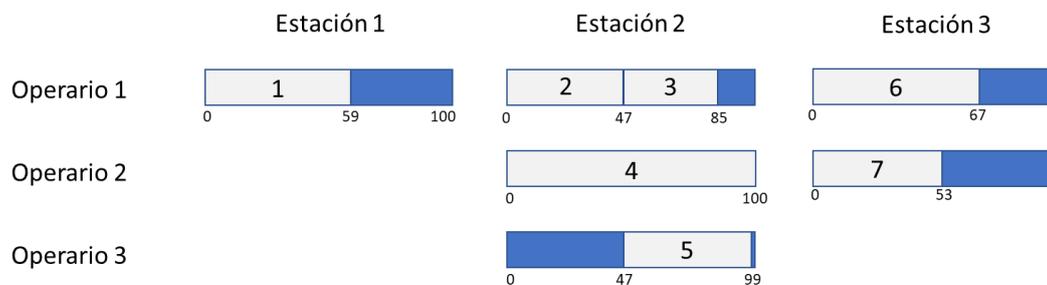


Figura 20. Asignación de tareas a estaciones para la solución inicial SMALBP-DTT2 de OPCT_PART.

Fuente: propia

6.3.2. Modificación de estaciones mediante intercambio y transferencia de tareas (Generación de M-estaciones)

El Paso 2 de OPCT_PART es igual al Paso 2 de OPWK_PART (sección 5.8.1) excepto en una de las condiciones para considerar factible un intercambio/transferencia de tareas entre dos estaciones: en OPWK_PART el número de trabajadores no puede aumentar (sección 5.8.1.2), mientras que en OPCT_PART no puede aumentar el tiempo de ciclo (sección 6.3.2.2).

El Paso 2 de OPCT_PART consiste en modificar las estaciones de la solución SMALBP-DTT2 inicial. Para ello se aplican cuatro tipos de modificaciones:

- **Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso:** sean j_1 y j_2 estaciones de la solución SMALBP-DTT2 en curso que no tienen tareas en común, con la tarea i_1 contenida en j_1 y la tarea i_2 contenida en j_2 ; un intercambio de una sola tarea consiste en permutar las tareas i_1 e i_2 entre j_1 y j_2 para generar j_1' (que contiene i_2 pero no i_1) y j_2' (que contiene i_1 pero no i_2). Se prueban todos los posibles intercambios de una sola tarea entre todas las estaciones de la solución SMALBP-DTT2 en curso.
- **Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso:** sean j_1 y j_2 estaciones de la solución SMALBP-DTT2 en curso que no tienen tareas en común y con la tarea i_1 contenida en j_1 ; una transferencia de una sola tarea consiste en mover la tarea i_1 de j_1 a j_2 para generar j_1' (que no

contiene i_1) y j_2' (que contiene i_1). Se estudian todas las posibles transferencias de una sola tarea entre todas las estaciones de la solución SMALBP-DTT2 inicial, incluso aquellas en las que j_1 contiene sólo la tarea i_1 .

- **Transferencia de una sola tarea entre las estaciones encontradas en “Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso”.** El proceso es idéntico al de transferir una sola tarea descrito en el párrafo anterior, con la única diferencia de que las estaciones iniciales son las encontradas con la modificación “Intercambio de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso”, incluyendo también las estaciones de la solución en curso SMALBP-DTT2 que no generaron ningún intercambio factible.
- **Intercambio de una sola tarea entre las estaciones encontradas en “Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso”.** El proceso es idéntico al de intercambiar una sola tarea descrito anteriormente, con la única diferencia de que las estaciones iniciales son las encontradas con la modificación “Transferencia de una sola tarea entre estaciones de la solución SMALBP-DTT2 en curso”, incluyendo también las estaciones de la solución en curso SMALBP-DTT2 que no generaron ninguna transferencia factible.

Si un intercambio/transferencia de tareas entre dos estaciones se considera factible en términos de precedencia (sección 6.3.2.1) y no aumenta el tiempo de ciclo (sección 6.3.2.2), dichas dos estaciones modificadas (llamadas M-estaciones) se incluyen en la lista de M-estaciones candidatas para el problema de partición con restricciones (Paso 3). En caso de que, después de intentar todos los intercambios y transferencias de tareas posibles, sigue habiendo una estación que no se ha podido modificar ni una sola vez, dicha estación también se incluye en la lista de M-estaciones candidatas.

6.3.2.1. Factibilidad de pares de estaciones modificadas según precedencias

Considérense las estaciones j_1 y j_2 , un intercambio/transferencia de tareas entre esas dos estaciones genera j_1' y j_2' . Las estaciones j_1' y j_2' se consideran factibles entre sí en términos de relaciones de precedencia si se cumplen todas las condiciones siguientes (para consultar un ejemplo, véase sección 5.8.1.1):

- Si no hay tareas (de las no incluidas en la estación j_1') que sean al mismo tiempo sucesoras y predecesoras con respecto a las tareas de la estación j_1' .
- Si no hay tareas (de las no incluidas en la estación j_2') que sean al mismo tiempo sucesoras y predecesoras con respecto a las tareas de la estación j_2' .
- Si en la estación j_1' no hay al mismo tiempo tareas sucesoras y predecesoras con respecto a la estación j_2' .

- Si en la estación j_2' no hay al mismo tiempo tareas sucesoras y predecesoras con respecto a la estación j_1' .

6.3.2.2. Factibilidad de pares de estaciones modificadas sin exceder el tiempo de ciclo ni la concentración máxima de trabajadores

Considérense las estaciones j_1' y j_2' , generadas por un intercambio/transferencia de tareas entre j_1 y j_2 . En esta sección se calcula el tiempo de ciclo y el número de trabajadores de las estaciones j_1' y j_2' . Si el tiempo de ciclo máximo entre j_1' y j_2' no es mayor que el tiempo de ciclo máximo entre j_1 y j_2 , entonces j_1' y j_2' son factibles.

Dado un conjunto de tareas asignadas a una estación con múltiples operarios, el procedimiento presentado en los dos párrafos siguientes se utiliza para determinar el menor tiempo de ciclo para el que existe una asignación factible con un número de trabajadores igual o inferior a la cota superior UB^0 . La cota inferior del tiempo de ciclo (LB_{CT_MAL}) se calcula mediante la Ec. 73 (donde N_{sttn} es el número de tareas asignadas a la estación evaluada).

$$LB_{CT_MAL} = \max \left(\max_{i=1, \dots, N_{sttn}} \left(P_{i,1} + \sum_{h \in ASC_i} P_{h,1} \right), \frac{\sum_{i=1}^{N_{sttn}} P_{i,1}}{UB^0} \right) \quad (\text{Ec. 73})$$

Dado un tiempo de ciclo, se ejecuta iterativamente el modelo matemático que resuelve un SMALBP con una estación y un número fijo de trabajadores, y que es introducido en la sección 5.3.2.2. El proceso iterativo consiste en resolver el modelo matemático aumentando el número de trabajadores (NW) en una unidad hasta encontrar una solución factible, o hasta alcanzar la cota superior del número de trabajadores en cada estación (UB^0).

Si se alcanza una asignación factible con UB^0 o menos trabajadores, se ha encontrado el tiempo de ciclo de la estación. En caso contrario, el tiempo de ciclo (que inicialmente equivale a la cota inferior) se incrementa en una unidad y se ejecuta de nuevo el procedimiento descrito en el párrafo anterior. Si el nuevo tiempo de ciclo es superior al tiempo de ciclo máximo entre j_1 y j_2 , el procedimiento finaliza sin éxito. Se establece un límite de tiempo de cálculo de β_4 segundos (el valor β_4 utilizado para la experiencia computacional se detalla en la sección 6.4).

Para la solución inicial del SMALBP-DTT2 utilizando la octava regla de prioridad (Figura 20), la lista de M-estaciones candidatas que son factibles en términos de precedencias y no aumentan el tiempo de ciclo se muestra en la Tabla 34.

Tareas incluidas en la M-estación	Tiempo de ciclo (CT)	Número de trabajadores
1	59	1
2 + 4 + 5	96	2
2 + 3 + 4 + 5	100	3
3 + 6 + 7	90	2
6+7	67	2

Tabla 34. M-estaciones candidatas a formar parte de la solución final OPCT_PART. Fuente: propia

6.3.3. Resolución de un problema de partición con restricciones

En el Paso 3, se utiliza un modelo matemático para resolver un problema de partición con restricciones considerando las M-estaciones. La lista de M-estaciones candidatas está compuesta por las M-estaciones encontradas en el Paso 2 en curso y las M-estaciones candidatas que se generaron en el Paso 2 para las $p2$ iteraciones anteriores. La propuesta de $p2$ evita tener una cantidad demasiado grande de M-estaciones candidatas, lo que conllevaría un aumento significativo de los tiempos computacionales para resolver el problema de partición; el valor de $p2$ utilizado para la experiencia computacional se detalla en la sección 6.4. Además, cada tarea individual también se incluye como una M-estación candidata. El objetivo es minimizar el tiempo de ciclo como objetivo principal y el número de trabajadores como objetivo secundario, considerando un máximo de UB^M M-estaciones seleccionadas. En caso de empate, se selecciona la solución que utiliza el menor número de M-estaciones incluidas en las $n2$ soluciones SMALBP-DTT2 anteriores (el valor $n2$ utilizado para la experiencia computacional se detalla en la sección 6.4). Dicha condición de desempate se añade para promover la generación de soluciones con M-estaciones distintas y, por tanto, para tratar de evitar la condición final del Paso 3 (es decir, encontrar una solución idéntica a otra anterior).

Datos

- UB^M cota superior del número de M-estaciones que se pueden seleccionar.
- N número de tareas ($i = 1, \dots, N$).
- J número de M-estaciones candidatas ($j = 1, \dots, J$).
- CT_j tiempo de ciclo de la M-estación j ($j = 1, \dots, J$).
- NW_j número de trabajadores en la M-estación j ($j = 1, \dots, J$).
- $OldS_j$ 1 si la M-estación j ha sido seleccionada en las $n2$ soluciones previas del problema de partición, 0 en caso contrario ($j = 1, \dots, J$).
- IS_{ji} 1 si en la M-estación j está incluida la tarea i , 0 en caso contrario:

$$i = 1, \dots, N; j = 1, \dots, J.$$

\overline{INC} grupos de M-estaciones que no tienen tareas en común pero que son infactibles por razones de precedencia.

R número de M-estaciones en cada grupo de \overline{INC} ($R = 2, \dots, N$).

$$\overline{INC} = \{(r_1, \dots, r_R) \mid r_1 \in \{1, \dots, J - (R - 1)\}, \dots, r_R \in \{r_1 + (R - 1), \dots, J\}, r_1, \dots, r_R \text{ infactibles}\}.$$

Dos M-estaciones ($R = 2$) son infactibles si en una de las M-estaciones hay al mismo tiempo tareas sucesoras y predecesoras con respecto a las tareas de la otra M-estación. Tres o más M-estaciones ($R = 3, \dots, N$) son infactibles si cada M-estación tiene relaciones de precedencia con sólo otras dos M-estaciones (más concretamente, tiene tareas predecesoras con respecto a las tareas de una M-estación y tareas sucesoras con respecto a las tareas de la otra M-estación) y no hay infactibilidades en ninguna combinación de $2, \dots, R - 1$ M-estaciones.

Variables

$x_j \in \{0,1\}$ 1 si se usa la M-estación j , 0 en caso contrario: $j = 1, \dots, J$.

CT_{max} tiempo de ciclo máximo entre las M-estaciones seleccionadas.

Función objetivo

La función objetivo minimiza el tiempo de ciclo como primer objetivo, el número total de trabajadores como objetivo secundario y el número de M-estaciones utilizadas en las $n2$ soluciones anteriores del problema de partición como objetivo terciario:

$$[MIN] CT_{max} + \frac{1}{\max_{j=1, \dots, J} (NW_j) \cdot UB^M + 1} \cdot \left(\sum_{j=1}^J NW_j \cdot x_j + \frac{1}{UB^M + 1} \cdot \sum_{j=1}^J OldS_j \cdot x_j \right) \quad (\text{Ec. 74})$$

Restricciones

$$\sum_{j=1}^J IS_{ji} \cdot x_j = 1 \quad i = 1, \dots, N \quad (\text{Ec. 75})$$

$$\sum_{j=r_1}^{r_R} x_j \leq R - 1 \quad (r_1, \dots, r_R) \in \overline{INC} \quad (\text{Ec. 76})$$

$$CT_{max} \geq CT_j \cdot x_j \quad j = 1, \dots, J \quad (\text{Ec. 77})$$

$$\sum_{j=1}^J x_j \leq UB^M \quad (\text{Ec. 78})$$

Las restricciones (75) garantizan que cada tarea se incluya en la solución final una sola vez. Las restricciones (76) evitan que se asignen M-estaciones infactibles por razones de precedencia. Las restricciones (77) obtienen el valor de la variable CT_{max} como el tiempo de ciclo máximo de las M-estaciones seleccionadas. La restricción (78) garantiza que el número de M-estaciones seleccionadas no supere la cota superior UB^M .

Para el ejemplo ilustrativo (Tabla 34), la solución obtenida utiliza 5 trabajadores y 3 M-estaciones (como se muestra en la Figura 18).

6.4. Experiencia computacional para comparar MILP-DDT2, HEUR_CT_PART, GRDY-DDT2 y OPCT_PART

En esta sección se evalúan los diferentes procedimientos diseñados para resolver el SMALBP-DDT2. Las características de la experiencia computacional realizada se describen en la sección 6.4.1. El modelo matemático MILP-DDT2 (sección 6.1), HEUR_CT_PART (sección 6.2), GRDY-DDT2 (sección 6.3.1) y OPCT_PART (sección 6.3) se comparan en la sección 6.4.2.1.

Cabe especificar que en la literatura no hay ningún otro procedimiento que aborde el SMALBP-DDT2. Sin embargo, Roshani y Giglio (2017) resuelven el SMALBP-2 con los mismos objetivos (minimización del tiempo de ciclo como primer objetivo y del número de trabajadores como segundo, considerando un número máximo de estaciones) utilizando dos algoritmos basados en SA: DSA e ISA. De todos los procedimientos propuestos en la literatura para el SMALBP-2 con un número fijo de estaciones, los procedimientos de Roshani y Giglio (2017) son los que mejores resultados han obtenido hasta el momento. En la sección 6.4.2.2, DSA e ISA se comparan con OPCT_PART, el procedimiento que presenta mejores resultados en la sección 6.4.2.1, considerando que los tiempos de tarea son independientes (y, así, invariables) del número de trabajadores en la estación (aunque OPCT_PART no se ha diseñado específicamente considerando tiempos de tarea deterministas).

6.4.1. Características de la experiencia computacional

De forma análoga a las experiencias computacionales precedentes, todos los

procedimientos para resolver las instancias han sido ejecutados en un ordenador con un procesador Intel® Core™ i5 CPU 650 a 3.33 GHz; por otro lado, el tiempo de computación para resolver cada instancia se ha fijado en un máximo de 3600 segundos.

En relación con HEUR_CT_PART y OPCT_PART, se realizó una prueba computacional inicial para determinar los siguientes parámetros: el parámetro β_3 para HEUR_CT_PART, y los parámetros β_3 , β_4 , p_2 y n_2 , para OPCT_PART. Para las pruebas computacionales preliminares se utilizaron los 50 ejemplares de Otto *et al.* (2013) anteriormente ya usados en otras pruebas preliminares (las instancias utilizadas se detallan en el Apéndice 1).

Por un lado, la Tabla 35 muestra los resultados de la prueba computacional previa con HEUR_CT_PART. Para comparar los resultados obtenidos por cada posible valor del parámetro β_3 se utiliza la función objetivo definida en Ec. 52. Los mejores resultados se obtienen con $\beta_3 = 5$ segundos.

	$\beta_3 = 5$ segundos	$\beta_3 = 10$ segundos	$\beta_3 = 30$ segundos
Sumatorio función objetivo	14536.33	14537.01	14561.56
Tiempo medio de cálculo (s)	1995	2260	2705

Tabla 35. Resultados de la experiencia computacional previa de HEUR_CT_PART. Fuente: propia

Por otro lado, la Tabla 36 contiene los resultados de la prueba computacional previa con OPCT_PART. Para comparar los resultados obtenidos por cada combinación de parámetros también se utiliza la función objetivo definida en Ec. 52. Para cada combinación de parámetros, la Tabla 36 expone el sumatorio de los valores de la función objetivo para los 50 ejemplares y, entre paréntesis, el tiempo medio de cálculo. Los mejores resultados, considerando el sumatorio de la función objetivo y el tiempo de cálculo, se obtienen con: $\beta_3 = 5$ segundos, $\beta_4 = 20$ segundos, $p_2 = 2$ y $n_2 = 2$.

	Incorporar las estaciones candidatas de las p_2 iteraciones anteriores	Se selecciona la solución que utiliza el menor número de estaciones incluidas en las n_2 soluciones anteriores		
		$n_2 = 1$	$n_2 = 2$	$n_2 = 3$
$\beta_3 = 5$ segundos $\beta_4 = 20$ segundos	$p_2 = 1$	14520.05 (2295 segundos)	14464.15 (2355 segundos)	14464.15 (2355 segundos)
	$p_2 = 2$	14159.23 (2340 segundos)	14125.05 (2360 segundos)	14127.95 (2350 segundos)
	$p_2 = 3$	14345.70 (2335 segundos)	14304.30 (2385 segundos)	14304.30 (2350 segundos)
$\beta_3 = 10$ segundos $\beta_4 = 20$ segundos	$p_2 = 1$	14520.05 (2520 segundos)	14464.00 (2505 segundos)	14460.61 (2550 segundos)
	$p_2 = 2$	14159.23 (2570 segundos)	14125.05 (2610 segundos)	14126.95 (2680 segundos)
	$p_2 = 3$	14345.70 (2605 segundos)	14285.00 (2595 segundos)	14287.66 (2600 segundos)
$\beta_3 = 5$ segundos $\beta_4 = 30$ segundos	$p_2 = 1$	14534.31 (2665 segundos)	14480.77 (2635 segundos)	14493.83 (2670 segundos)

	Incorporar las estaciones candidatas de las $p2$ iteraciones anteriores	Se selecciona la solución que utiliza el menor número de estaciones incluidas en las $n2$ soluciones anteriores		
		$n2 = 1$	$n2 = 2$	$n2 = 3$
	$p2 = 2$	14167.28 (2705 segundos)	14150.42 (2770 segundos)	14143.01 (2900 segundos)
	$p2 = 3$	14416.29 (2825 segundos)	14380.91 (2905 segundos)	14388.44 (2880 segundos)
$\beta3 = 10$ segundos $\beta4 = 30$ segundos	$p2 = 1$	14531.94 (2790 segundos)	14480.77 (2865 segundos)	14493.83 (2905 segundos)
	$p2 = 2$	14165.80 (2865 segundos)	14145.16 (2990 segundos)	14143.01 (3040 segundos)
	$p2 = 3$	14409.53 (2980 segundos)	14373.66 (3100 segundos)	14388.44 (3095 segundos)

Tabla 36. Resultados de la experiencia computacional previa de OPCT_PART. Fuente: propia

Las instancias utilizadas en la sección 6.4.2.1 para comparar los procedimientos propuestos proceden de Otto *et al.* (2013). Se utiliza un total de 250 instancias (125 instancias con 50 tareas y 125 con 100 tareas), las mismas que son utilizadas en las experiencias computacionales para el SMALBP-DTT1 (para más detalle sobre la selección de las 250 instancias, ver la sección 5.4.1). Para adaptar las instancias de modo que consideren tiempos de tareas dependientes del número de trabajadores en la estación, se sigue el enfoque utilizado para las experiencias computacionales del SMALBP-DTT1: los tiempos de proceso se incrementan en una unidad por cada trabajador adicional en la estación ($P_{i,o+1} = P_{i,o} + 1$) y se establece un máximo de 4 trabajadores en la misma estación ($UB^0 = 4$). El número máximo de estaciones de cada instancia (UB^M) se fija al número de estaciones hallado por el procedimiento de HEUR_PART (Apéndice 2).

Finalmente, para la comparación con DSA e ISA, se utilizan los ejemplares que se resuelven en Roshani y Giglio (2017): algunas de las instancias de la literatura de referencia, que pueden descargarse en www.assembly-line-balancing.de. Para comparar OPCT_PART con DSA e ISA, los tiempos de las tareas se consideran independientes (es decir, invariables) del número de trabajadores. La cota superior del número de trabajadores en cada estación se establece en 2 ($UB^0 = 2$), análogamente a Roshani y Giglio (2017).

6.4.2. Análisis de los resultados

6.4.2.1. Comparación de los procedimientos propuestos

La comparación entre el modelo matemático MILP-DTT2 (sección 6.1), HEUR_CT_PART (sección 6.2), GRDY_DTT2 (sección 6.3.1) y OPCT_PART (sección 6.3) se presenta en la Tabla 37. La mejor solución es la que obtiene el menor tiempo de ciclo y, en caso de empate, el menor número de trabajadores.

La Tabla 37 muestra el porcentaje de ejemplares en los que cada procedimiento alcanza

(y no alcanza) la mejor solución y el mejor tiempo de ciclo encontrados por cualquiera de los procedimientos propuestos, y la cota inferior (LB_{CT}) del tiempo de ciclo. Además, se expone el número de soluciones factibles encontradas, la media de la cota inferior del tiempo de ciclo, la media del tiempo de ciclo y del número de trabajadores obtenidos, la diferencia (*gap*) media del tiempo de ciclo respecto a la cota inferior y a la mejor solución de cada ejemplar, y la media del tiempo de cálculo.

La cota inferior del tiempo de ciclo (LB_{CT}) de cada instancia se calcula con la Ec. 67. Cabe comentar que debido a que en el SMALBP-DTT2 los tiempos de las tareas dependen del número de trabajadores en la estación, en algunos casos la cota inferior del tiempo de ciclo puede ser inalcanzable. El *gap* del tiempo de ciclo de una instancia respecto a la cota inferior y a la mejor solución se calcula utilizando la Ec. 79. Los resultados obtenidos para cada procedimiento en las 250 instancias de Otto *et al.* (2013) se muestran en el Apéndice 6.

$$\text{Gap}_{CT} = \frac{CT_{\text{solución evaluada}} - CT_{\text{cota inferior/mejor solución}}}{CT_{\text{cota inferior/mejor solución}}} \quad (\text{Ec. 79})$$

		MILP-DTT2	HEUR_CT_PART	GRDY_DTT2	OPCT_PART
50 tareas	Mejor solución alcanzada	60.8%	36.8%	23.2%	84.8%
	Mejor solución no alcanzada	39.2%	63.2%	76.8%	15.2%
	Mejor tiempo de ciclo alcanzado	92.0%	80.0%	87.2%	98.4%
	LB_{CT} alcanzado	82.4%	73.6%	83.2%	87.2%
	Número de soluciones factibles	125 (100%)	125 (100%)	125 (100%)	125 (100%)
	LB_{CT} medio	652.14			
	Tiempo de ciclo medio	676.17	662.05	660.31	659.63
	Número medio de trabajadores	33.02	34.25	34.97	33.38
	Gap medio CT con LB_{CT}	4.71%	2.74%	2.49%	2.29%
	Gap medio CT con mejor solución	2.44%	0.43%	0.18%	0.02%
Tiempo medio de cálculo (s)	3510	2130	7	2970	
100 tareas	Mejor solución alcanzada	0%	19.2%	52.0%	95.2%
	Mejor solución no alcanzada	100%	80.8%	48.0%	4.8%
	Mejor tiempo de ciclo alcanzado	0%	56.8%	97.6%	100%
	LB_{CT} alcanzado	0%	44.8%	83.2%	85.6%
	Número de soluciones factibles	26 (20.8%)	114 (91.2%)	125 (100%)	125 (100%)
	LB_{CT} medio*		569.14		
	Tiempo de ciclo medio*		580.46	570.75	570.71
	Número medio de trabajadores*		52.63	52.61	51.46
	Gap medio CT con LB_{CT} *		2.02%	0.49%	0.48%
Gap medio CT con mejor solución*	1.52%		0.01%	0%	

	MILP-DDT2	HEUR_CT_PART	GRDY_DDT2	OPCT_PART
Tiempo medio de cálculo (s)	3600	3030	11	3270

Tabla 37. Resultados obtenidos por los tres procedimientos propuestos (* considerando sólo los 114 ejemplares en los que HEUR_CT_PART, GRDY_DDT2 y OPCT_PART encuentran una solución factible). Fuente: propia

Analizando los resultados de las instancias de 50 tareas, se observa que OPCT_PART obtiene la mejor solución en el 84.8% de las instancias y el mejor tiempo de ciclo en el 98.4% de ellas. MILP-DDT2 es el procedimiento que presenta el segundo mejor porcentaje de instancias que alcanzan la mejor solución (60.8%), obtenido el mejor tiempo de ciclo en el 92.0% de ellas. En relación a HEUR_CT_PART, alcanza la mejor solución en un porcentaje menor de ejemplares (36.8%) y, entre los cuatro procedimientos, es el que presenta el porcentaje más bajo de ejemplares que obtienen el mejor tiempo de ciclo (80.0%). Comparando OPCT_PART con GRDY_DDT2 (recuérdese que GRDY_DDT2 se utiliza en OPCT_PART para encontrar la solución inicial de SMALBP-DDT2), GRDY_DDT2 alcanza la mejor solución en el 23.2% de las instancias (en particular, OPCT_PART mejora los resultados obtenidos por GRDY_DDT2 en 84 de las 125 instancias) aunque obtiene el mejor valor del tiempo de ciclo en un 87.2% de los ejemplares. OPCT_PART obtiene también una media de tiempo de ciclo inferior (659.63 unidades de tiempo). En relación con la cota inferior del tiempo de ciclo LB_{CT} (con una media de 652.14 unidades de tiempo), que con tiempos de tarea dependientes puede ser inalcanzable en algunos ejemplares, OPCT_PART obtiene el mismo tiempo de ciclo en el 87.2% de los casos. En cuanto al *gap* del tiempo de ciclo medio con la mejor solución de cada instancia y con LB_{CT} , OPCT_PART presenta mejores resultados que los otros procedimientos propuestos. En términos de tiempo de ejecución, GRDY_DDT2 obtiene el tiempo medio de cálculo más bajo. Sin embargo, las diferencias de tiempo de cálculo entre GRDY_DDT2 y OPCT_PART no suelen ser significativas en las líneas de montaje de la vida real.

En relación con los resultados de las instancias de 100 tareas, OPCT_PART obtiene las mejores soluciones en el 95.2% de las instancias y el mejor tiempo de ciclo en todas ellas. Comparando OPCT_PART con GRDY_DDT2, GRDY_DDT2 obtiene la mejor solución en el 52.0% de las instancias y OPCT_PART mejora los resultados obtenidos por GRDY_DDT2 en 57 de las 125 instancias. En cuanto al modelo matemático, sólo se encuentra una solución en el 20.8% de las instancias y en ninguna de ellas se alcanza la mejor solución. Por lo tanto, el modelo matemático se excluye para las comparaciones posteriores. HEUR_CT_PART solo obtiene solución factible en el 91.2% de las instancias. La Tabla 37 muestra la comparación entre HEUR_CT_PART, GRDY_DDT2 y OPCT_PART teniendo únicamente en cuenta las 114 instancias en las que los tres procedimientos encuentran una solución factible. Los resultados muestran que OPCT_PART obtiene una media del tiempo de ciclo inferior (570.71 unidades de tiempo)

y que su media está muy próxima a la de la cota inferior (569.14 unidades de tiempo). De hecho, la diferencia entre el tiempo de ciclo medio de OPCT_PART y la cota inferior LB_{CT} es de sólo el 0.48%. Por último, GRDY_DTT2 tiene un tiempo medio de cálculo inferior al de OPCT_PART pero, como ya se ha dicho, estas diferencias no suelen ser significativas en las líneas de montaje de la vida real.

6.4.2.2. Comparación entre OPCT_PART, DSA e ISA

Como se ha mencionado anteriormente, en la literatura no se ha publicado ningún otro procedimiento que resuelva el SMALBP-DDT2. Sin embargo, Roshani y Giglio (2017) abordan el SMALBP-2 con los mismos objetivos jerárquicos (minimizar el tiempo de ciclo como objetivo primario, con el número de trabajadores como secundario) y tiempos de proceso deterministas (y, así, invariables del número de trabajadores en la estación). Los autores proponen dos algoritmos basados en SA: DSA e ISA. En la Tabla 38 se compara el mejor resultado obtenido por DSA e ISA con OPCT_PART. A efectos de comparación, los tiempos de las tareas también se consideran independientes del número de trabajadores al resolver OPCT_PART (aunque dicho procedimiento no se ha diseñado específicamente considerando tiempos de tarea deterministas).

En la Tabla 38, para cada ejemplar, si OPCT_PART obtiene un mejor tiempo de ciclo (objetivo principal), las celdas del tiempo de ciclo y el número de trabajadores se colorean de verde (de rojo si la mejor solución entre DSA e ISA obtiene un tiempo de ciclo mejor). Si OPCT_PART mejora sólo el número de trabajadores (objetivo secundario), se colorea de verde sólo la celda de los trabajadores (de rojo si la mejor solución entre DSA e ISA obtiene un número de trabajadores mejor). Si las soluciones son iguales, las celdas permanecen blancas.

Como se observa en la Tabla 38, los resultados de OPCT_PART son mejores que los obtenidos por la mejor solución entre DSA e ISA. OPCT_PART supera a la mejor solución entre DSA e ISA en 24 instancias (35.8%), obtiene los mismos resultados en 39 instancias (58.2%) y sólo en 4 instancias obtiene peores resultados (6.0%). OPCT_PART alcanza una mejor solución en el 94.0% de los casos (frente a la mejor solución entre DSA e ISA que lo hace en el 64.2%) y un mejor tiempo de ciclo en el 98.5% de los casos (por el 65.7% de la mejor solución entre DSA e ISA).

Problema	Número máximo de estaciones (UB^M)	DSA + ISA (Roshani and Giglio, 2017)		OPCT_PART		
		Tiempo de ciclo	Trabajadores	Tiempo de ciclo	Trabajadores	Tiempo de cálculo (s)
Merten	3	6	6	6	6	2.2
	2	11	3	11	3	5.7
	1	17	2	17	2	0.2
Bowman	5	17	5	17	5	0.9
	4	17	6	17	6	1.6
	3	26	4	26	4	2.1

Problema	Número máximo de estaciones (UB^M)	DSA + ISA (Roshani and Giglio, 2017)		OPCT_PART		
		Tiempo de ciclo	Trabajadores	Tiempo de ciclo	Trabajadores	Tiempo de cálculo (s)
	2	28	3	28	3	0.7
	1	55	2	55	2	0.3
Jaeschke	6	6	8	6	8	1.7
	5	8	6	8	6	2.9
	4	9	5	9	5	4.1
	3	10	5	10	5	2.4
	2	14	4	14	4	1.5
	1	28	2	28	2	0.7
Jackson	6	7	8	7	8	3.4
	5	8	7	8	7	12.6
	4	9	6	9	6	14.5
	3	11	5	11	5	9.9
	2	15	4	15	4	1.2
	1	28	2	28	2	0.5
Mansoor	3	45	5	45	5	80.8
	2	59	4	59	4	58.2
	1	116	2	116	2	26.5
Mitchell	8	13	9	13	10	87.6
	7	14	8	14	8	64.9
	6	16	7	16	8	32.3
	5	16	7	16	7	78.5
	4	20	7	20	7	29.2
	3	27	4	27	4	22.4
	2	39	3	39	4	10.9
Heskia	6	108	10	108	10	378.8
	5	124	9	124	9	614.8
	4	139	8	139	8	801.4
	3	183	6	183	6	679.3
	2	274	4	274	4	582.8
Sawyer	8	25	14	25	14	325.7
	7	28	13	28	12	298.0
	6	31	12	30	12	128.6
	5	40	9	39	9	315.2
	4	42	8	41	8	133.9
	3	57	6	56	6	80.9
	2	83	4	81	4	45.7
Kilbridge	6	55	11	55	11	696.9
	5	60	10	58	10	451.8
	4	75	8	73	8	626.6
	3	98	6	96	6	796.7
	2	146	4	144	4	690.7
Tonge	13	158	25	156	25	1356.0
	12	177	23	156	23	1782.1
	10	197	20	189	20	2105.8
	9	225	18	216	18	1874.7
	8	240	16	238	16	1906.6
	7	280	14	266	14	2210.2
Arcus1	6	311	12	310	12	1224.7
	15	3691	23	3691	23	2690.8
	14	3765	22	3691	21	2016.7
	12	4320	21	4131	21	1279.1
	10	4811	18	4811	18	1354.1
	8	6194	15	5905	15	1252.9
Arcus2	6	7688	12	7604	11	1004.0
	17	5689	30	5689	30	3148.1
	15	6145	28	5904	29	3209.3
	13	7235	24	6948	25	3094.7
	12	7467	24	7467	24	3057.5
	10	9726	18	9617	20	2961.6
	8	10889	16	10891	15	2503.4

Problema	Número máximo de estaciones (UB^M)	DSA + ISA (Roshani and Giglio, 2017)		OPCT_PART		
		Tiempo de ciclo	Trabajadores	Tiempo de ciclo	Trabajadores	Tiempo de cálculo (s)
	6	13350	12	13350	12	2457.7
Media		1415.24	10.18	1395.21	10.22	756.69
Mejor solución alcanzada		64.2%		94.0%		
Mejor tiempo de ciclo alcanzado		65.7%		98.5%		

Tabla 38. Tabla comparativa entre OPCT_PART y la mejor solución entre DSA e ISA. Fuente: propia

6.5. Conclusiones procedimientos de resolución SMALBP-DTT2

En la sección 6 de la presente tesis doctoral se aborda la resolución de la variante SMALBP-DTT2. En primer lugar, se propone un modelo matemático denominado MILP-DTT2 (sección 6.1). En segundo lugar, se desarrollan dos procedimientos heurísticos basados en la resolución de un problema de partición con restricciones: HEUR_CT_PART (sección 6.2), que parte de una solución SALBP-2 a partir de la cual se generan uniones de estaciones; y OPCT_PART (sección 6.3), que parte de una solución SMALBP-DTT2 y genera nuevas soluciones mediante intercambio y transferencia de tareas. Asimismo, se presenta un procedimiento heurístico constructivo *greedy* llamado GRDY_DTT2 (sección 6.3.1), el cual se utiliza para encontrar una solución inicial SMALBP-DTT2 de OPCT_PART.

Los procedimientos aquí propuestos son los primeros en la literatura que resuelven el SMALBP-DTT2. Los resultados computacionales muestran que OPCT_PART presenta un mejor comportamiento para resolver SMALBP-DTT2 que los otros procedimientos diseñados.

El procedimiento OPCT_PART también se compara con DSA e ISA, procedimientos propuestos por Roshani y Giglio (2017) para SMALBP-2 con los mismos objetivos jerárquicos, aunque con los tiempos de las tareas independientes del número de trabajadores en la estación. DSA e ISA son los procedimientos que obtienen los mejores resultados para el SMALBP-2 hasta el momento. Considerando que los tiempos de las tareas son invariables, OPCT_PART presenta mejores resultados que DSA e ISA (aunque OPCT_PART no se ha diseñado específicamente considerando tiempos de tarea deterministas).

7. Conclusiones finales, futuras vías de investigación y contribuciones

7.1. Conclusiones

Esta tesis doctoral estudia el problema de equilibrado de líneas de montaje con múltiples operarios en la estación (*Multi-manned Assembly Line Balancing Problem*, MALBP). En particular, se investiga una variante del MALBP que permite representar situaciones más cercanas a la realidad: aquellas en las que los operarios de una misma estación pueden interferir entre ellos, provocando un incremento en los tiempos de proceso de las tareas (*Multi-manned Assembly Line Balancing Problem with Dependent Task Times*, MALBP-DTT).

Analizando el estado del arte del MALBP se concluye que sólo un reducido número de artículos estudian las posibles interferencias entre trabajadores y, en particular, la posibilidad de que los tiempos de tarea dependan del número de trabajadores en la estación.

En esta tesis doctoral se desarrollan procedimientos de resolución exactos y heurísticos para las variantes del SMALBP-DTT1 (minimizar el número de trabajadores y estaciones, a partir de un tiempo de ciclo fijo) y del SMALBP-DTT2 (minimizar el tiempo de ciclo y el número de trabajadores, a partir de un número fijo de estaciones).

Por un lado, para el SMALBP-DTT1 se propone:

- MILP-DTT1 (sección 5.1), un modelo matemático que corrige, evitando soluciones infactibles, y mejora los que se han publicado en la literatura hasta el momento.
- 1RF y 2RF (sección 5.2), dos procedimientos basados en la matheurística de Relajar y Fijar.
- HEUR_PART (sección 5.3), un procedimiento heurístico basado en la resolución de un problema de partición con restricciones. HEUR_PART presenta un mejor comportamiento para resolver el SMALBP-DTT1 que MILP-DTT1, 1RF y 2RF (sección 5.4).
- Conjunto de variantes de HEUR_PART (sección 5.5): HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE. La variante HEUR_PART_SGL obtiene los mismos resultados que HEUR_PART con un tiempo de cálculo inferior.

- Aplicación de Empirically Adjusted Greedy Heuristics (EAGH; sección 5.6). Asimismo, se presenta un nuevo procedimiento (denominado "EAGH-CKTL") para diseñar heurísticas *greedy* basado en el uso del procedimiento EAGH combinado con el concepto de cóctel de heurísticas. EAGH y EAGH-CKTL se utilizan para generar nuevas reglas de prioridad para la resolución del SMALBP-DTT1 mediante HEUR_PART. En esta tesis doctoral se muestra la eficacia tanto de EAGH como de EAGH-CKTL en el diseño de nuevas reglas de prioridad.
- MULTI_HEUR_PART (sección 5.7), un procedimiento heurístico basado en la resolución de problemas de partición combinados.
- OPWK_PART (sección 5.8), un procedimiento que combina la resolución de problemas de partición con la generación de estaciones vecinas mediante intercambio y transferencia de tareas.
- GRDY_DTT1 (sección 6.3.1), un procedimiento heurístico constructivo *greedy* orientado a estaciones que se utiliza para resolver iterativamente el SMALBP-DTT1 en GRDY_DTT2.

OPWK_PART obtiene mejores resultados que cualquier otro de los procedimientos de resolución propuestos en esta tesis. OPWK_PART también mejora cualquiera de los procedimientos publicados en la literatura para esta variante hasta el momento, aunque el objetivo a optimizar no es exactamente el mismo.

Con relación al SMALBP-DTT2, se propone:

- MILP-DTT2 (sección 6.1), el primer modelo matemático que se formula para el SMALBP-DTT2.
- HEUR_CT_PART (sección 6.2), un procedimiento heurístico basado en la resolución de un problema de partición con restricciones, que parte de una solución SALBP-2 a partir de la cual se generan uniones de estaciones.
- OPCT_PART (sección 6.3), un procedimiento heurístico basado en la resolución de un problema de partición con restricciones, que partiendo de una solución SMALBP-DTT2 genera nuevas estaciones vecinas mediante intercambio y transferencia de tareas.
- GRDY_DTT2 (sección 6.3.1), un procedimiento heurístico constructivo *greedy* que se utiliza para encontrar la solución inicial SMALBP-DTT2 de OPCT_PART.

En la literatura no han sido publicados procedimientos que resuelvan el SMALBP-DTT2.

Así pues, los aquí propuestos son los primeros que tratan dicho problema. Los resultados computacionales muestran que OPCT_PART presenta mejores resultados que los otros procedimientos diseñados.

La eficacia de OPCT_PART también se evalúa comparándola con los procedimientos propuestos por Roshani y Giglio (2017) para SMALBP-2 con tiempos de proceso deterministas (los cuales presentan los mejores resultados para el SMALBP-2 hasta el momento). OPCT_PART, aunque no fue específicamente concebida para resolver el SMALBP-2 con tiempos invariables, obtiene mejores resultados que los procedimientos de Roshani y Giglio (2017).

En conclusión, esta tesis doctoral alcanza el objetivo de diseñar procedimientos que puedan ser de interés para la comunidad científica y que mejoren los que han sido publicados en la literatura hasta el momento para la variante del problema MALBP con tiempos de proceso dependientes (MALBP-DTT).

7.2. Futuras vías de investigación

La variante del MALBP con tiempos de proceso dependientes del número de operarios en la estación (MALBP-DTT) es una alternativa que permite representar situaciones más cercanas a la realidad donde las interferencias entre operarios se ven representadas. A continuación, se proponen futuras vías de investigación en cuanto a las técnicas de resolución y referentes al problema abordado.

Como futuras vías de investigación de la variante del MALBP-DTT pueden estudiarse otros enfoques para su resolución: tanto procedimientos heurísticos (por ejemplo, metaheurísticas como ACO, GA, PSO, SA o TS) como exactos (desarrollando, por ejemplo, un B&B).

Otra posibilidad consistiría en profundizar en el uso del procedimiento EAGH-CKTL (que se basa en el uso de EAGH combinado con el concepto de cóctel de heurísticas) para diseñar nuevas heurísticas *greedy* que resuelvan otras variantes del problema general de equilibrado de líneas de montaje o, incluso, otros tipos de problemas.

Por otro lado, también se pueden modificar alguna característica del problema a resolver. Una posibilidad sería considerar que los tiempos de proceso también dependan de otros factores, como por ejemplo del hecho de procesar de manera simultánea tareas que interfieren entre ellas. En un vehículo, se puede dar el caso de que dos operarios realicen tareas en partes distantes, con lo que puede no haber interferencias, o que trabajen en una zona próxima, con lo que podrían considerarse interferencias entre grupos/parejas de tareas realizadas simultáneamente. En este caso, también podría considerarse que los

aumentos del tiempo de proceso podrían depender del tiempo en que dichas tareas se procesan de forma simultánea.

Otras posibilidades serían, por ejemplo, incorporar la diferenciación entre trabajadores cualificados y no cualificados, o considerar la presencia de tareas que deban ser realizadas por más de un operario.

7.3. Contribuciones

Las publicaciones científicas derivadas de esta tesis doctoral son las siguientes:

- Andreu-Casas, E., García-Villoria, A. & Pastor, R. (2022). Multi-manned assembly line balancing problem with dependent task times: a heuristic based on solving a partition problem with constraints. *European Journal of Operational Research*, 302(1), 96-116. <https://doi.org/10.1016/j.ejor.2021.12.002>
- Andreu-Casas, E., García-Villoria, A. & Pastor, R. (2023a). Solving the multi-manned assembly line balancing problem with dependent task times by means of Empirically Adjusted Greedy Heuristics (EAGH). *DYNA Management*, 11(1). <https://doi.org/10.6036/MN10905>
- Andreu-Casas, E., García-Villoria, A. & Pastor, R. (2023b). Nuevas heurísticas empíricamente ajustadas para la resolución del problema de equilibrado de líneas de montaje con múltiples operarios y tiempos de tarea dependientes. Publicación en DYNA aprobada el 12/09/2023.
- Andreu-Casas, E., García-Villoria, A. & Pastor, R. (2023c). Multi-manned assembly line balancing problem with dependent task times: minimising cycle time. (segunda revisión en curso en *Computer and Operations Research*)

8. Referencias

- Andrés, C., Miralles, C. & Pastor, R. (2008). Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. *European Journal of Operational Research*, 187, 1212-1223.
- Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32, 909-932.
- Becker, C. & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3), 694-715.
- Becker, C. & Scholl, A. (2009). Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. *European Journal of Operational Research*, 199(2), 359-374.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), 238-252.
- Boysen, N., Schulze, P. & Scholl, A. (2022). Assembly line balancing: What happened in the last fifteen years?. *European Journal of Operational Research*, 301(3), 797-814.
- Capacho, L., Pastor, R., Dolgui, A. & Gunshinskaya, O. (2009). An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem. *Journal of Heuristics*, 15, 109-132.
- Cevikcan, E., Durmusoglu, M. B. & Unal, M. E. (2009). A team-oriented design methodology for mixed model assembly systems. *Computers & Industrial Engineering*, 56(2), 359-374.
- Chang, H. J. & Chang, T. M. (2010). Simultaneous perspective-based mixed-model assembly line balancing problem. *Tamkang Journal of Science and Engineering*, 12, 327-336.
- Chen, Y. (2017). A hybrid algorithm for allocating tasks, operators, and workstations in multi-manned assembly lines. *Journal of Manufacturing Systems*, 42, 196-209.
- Chen, Y., Cheng, C. & Li, J. (2018). Resource-constrained assembly line balancing problems with multi-manned workstations. *Journal of Manufacturing Systems*, 48, 107-119.
- Chutima, P. (2022). A comprehensive review of robotic assembly line balancing problem. *Journal of Intelligent Manufacturing*, 33, 1–34.
- Clarke, G. & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581.
- Corday, R. (2014). The evolution of assembly lines: A brief history. *ROBOTS Association*,

consultado el 8 octubre 2022, <https://robohub.org/the-evolution-of-assembly-lines-a-brief-history/>

Corominas, A. (2005). Empirically adjusted greedy algorithms (EAGH): a new approach to solving combinatorial optimisation problems. *Polytechnic University of Catalonia working paper IOC-DT-P-2005-22*.

Corominas, A., García-Villoria, A. & Pastor, R. (2010). Fine-tuning a parametric Clarke and Wright heuristic by means of EAGH (empirically adjusted greedy heuristics). *Journal of the Operational Research Society*, 61(8), 1309-1314.

Corominas, A., García-Villoria, A. & Pastor, R. (2012). Cocktail of heuristics for solving hard problems. *DYNA*, 87(3), 275-278.

Corominas, A., García-Villoria, A. & Pastor, R. (2014). Improving parametric Clarke and Wright algorithms by means of iterative empirically adjusted greedy heuristics. *SORT*, 38(1), 3-12.

Corominas, A. & Pastor, R. (2011). Designing greedy algorithms for the flow-shop problem by means of Empirically Adjusted Greedy Heuristics (EAGH). *Journal of the Operational Research Society*, 62(9), 1704-1710.

Delice, Y., Aydoğan, E. K., Özcan, U. & İlkay, M. S. (2017). Balancing two-sided U-type assembly lines using modified particle swarm optimization algorithm. *4OR*, 15(1), 37-66.

Dillenberger, C., Escudero, L. F., Wollensak, A. & Zhang, W. (1994). On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, 75, 275-286.

Dimitriadis, S. G. (2006). Assembly line balancing and group working: a heuristic procedure for workers' groups operating on the same product and workstation. *Computers & Operations Research*, 33(9), 2757-2774.

Ding, F-Y., Zhu, J. & Sun, H. (2006). Comparing two weighted approaches for sequencing mixed-model assembly lines with multiple objectives. *International Journal of Production Economics*, 102, 108-131.

Esfandyari, A. & Roshani, A. (2020). Assembly line balancing problem with skilled and unskilled workers: The advantages of considering multi-manned workstations. *Journal of Industrial and Systems Engineering*, 13, 66-77.

Fattahi, P., Roshani, A. & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *The international journal of advanced manufacturing technology*, 53, 363-378.

García-Villoria, A., Corominas, A., Nadal, A. & Pastor, R. (2018). Solving the accessibility windows assembly line problem level 1 (AWALBP-L1) with precedence constraints.

European Journal of Operational Research, 271, 882-895.

Ghosh, S. & Gagnon, R.J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 27, 637-670.

Giglio, D., Paolucci, M., Roshani, A. & Tonelli, F. (2017). Multi-manned assembly line balancing problem with skilled workers: a new mathematical formulation. *International Federation of Automatic Control*, 50(1), 1211-1216.

Hoffmann, T. R. (1963). Assembly line balancing with a precedence matrix. *Management Science*, 9, 551-562.

Make, M. R. A., Rashid, M. F. F. A. & Razali, M. M. (2017). A review of two-sided assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 89(5-8), 1743–1763.

Kazemi, A. & Sedighi, A. (2013a). A cost-oriented model for multi-manned assembly line balancing problem. *Journal of Optimization in Industrial Engineering*, 2(13), 13-25.

Kazemi, A. & Sedighi, A. (2013b). A cost-oriented model for balancing mixed-model assembly lines with multi-manned workstations. *International Journal of Services and Operations Management*, 16(3), 289-309.

Kellegöz, T. (2017). Assembly line balancing problems with multi-manned stations: a new mathematical formulation and Gantt based heuristic method. *Annals of Operations Management*, 253(1), 377-404.

Kellegöz, T. & Toklu, B. (2012). An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations. *Computers and Operations Research*, 39(12), 3344-3360.

Kellegöz, T. & Toklu, B. (2015). A priority rule-based constructive heuristic and an improvement method for balancing assembly lines with parallel multi-manned workstations. *International Journal of Production Research*, 53(3), 736-756.

Kim, Y. K., Song, W. S. & Kim, J. H. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computer and Operations Research*, 36, 853-865.

Lagarias, J. C., Reeds, J. A., Wright, M. H. and Wright, P. E. (1998) Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9, 112-147.

Lopes, T. C., Pastre, G. V., Michels, A. S. & Magatao, L. (2020). Flexible multi-manned assembly line balancing problem: Model, heuristic procedure, and lower bounds for line length minimization. *Omega*, 95, 102063.

Lusa, A. (2008). A survey of the literature on the multiple or parallel assembly line balancing problem. *European Journal of Industrial Engineering*, 2, 50-72.

Martino, L. & Pastor, R. (2010). Heuristic procedures for solving the general assembly line balancing problem with setups. *International Journal of Production Research*, 48, 1787-1804.

Mertens, P. (1967). Fließbandabshmmung mit dem Verfahren der begrenzten Enumeration nach Müller-Merbach. *Ablauf-und Planungsforschung*, 8, 429-433.

Michels, A. S., Lopes, T. C. & Magatao, L. (2020). An exact method with decomposition techniques and combinatorial Benders' cuts for the type-2 multi-manned assembly line balancing problem. *Operations Research Perspectives*, 7, 100163.

Michels, A. S., Lopes, T. C., Sikora, C. G. S. & Magatao, L. (2019). A Benders' decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem. *European Journal of Operational Research*, 278, 768-808.

Nelder, J. A. & Mead, R. (1965). A simplex method for function minimisation. *The Computer Journal*, 7(4), 308-313.

Otto, A. & Battaïa, O. (2017). Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: A survey. *Computers & Industrial Engineering*, 111, 467-480.

Otto, A., Otto, C. & Scholl, A. (2013). Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. *European Journal of Operational Research*, 228, 33-45.

Pastor, R., Andrés, C., Duran, A. & Pérez, M. (2002). Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the tasks dispersion. *Journal of the Operational Research Society*, 53, 1317-1323.

Qian, X. & Fan, Q. (2011). Solving multi-manned assembly line balancing problem by heuristic-mixed genetic algorithm. *International Conference on Information Management, Innovation Management and Industrial Engineering*, 3, 320-323.

Roshani, A. & Ghazi Nezami, F. (2017). Mixed model multi-manned assembly line balancing problem: a mathematical model and a simulated annealing approach. *Assembly Automation*, 37(1), 34-50.

Roshani, A. & Giglio, D. (2015a). A mathematical programming formulation for cost-oriented multi-manned assembly line balancing problem. *International Federation of Automatic Control*, 48(3), 2293-2298.

Roshani, A. & Giglio, D. (2015b). A simulated annealing approach for multi-manned assembly line balancing problem type II. *International Federation of Automatic Control*, 48(3), 2299-2304.

- Roshani, A. & Giglio, D. (2017). Simulated annealing algorithms for the multi-manned assembly line balancing problem: minimising cycle time. *International Journal of Production Research*, 55(10), 2731-2751.
- Roshani, A. & Giglio, D. (2020). A Tabu Search Algorithm for the Cost-Oriented Multi-Manned Assembly Line Balancing Problem. *International Journal of Industrial Engineering & Production Research*, 31(2), 189-202.
- Roshani, A. & Roshani, A. (2012). Multi-manned assembly line balancing problem: minimizing cycle time. *Proceedings of the IIE Asian Conference*, 612-620.
- Roshani, A., Roshani, A., Roshani, A., Salehi, M. & Esfandyari, A. (2013). A simulated annealing algorithm for multi-manned assembly line balancing problem. *Journal of Manufacturing Systems*, 32(1), 238-247.
- Sahin, M. & Kellegöz, T. (2019a). Balancing multi-manned assembly lines with walking workers: problem definition, mathematical formulation, and an electromagnetic field optimisation algorithm. *International Journal of Production Research*, 1-19.
- Sahin, M. & Kellegöz, T. (2019b). A new mixed-integer linear programming formulation and particle swarm optimization based hybrid heuristic for the problem of resource investment and balancing of the assembly line with multi-manned workstations. *Computers & Industrial Engineering*, 133, 107-120.
- Sahin, M. & Kellegöz, T. (2022). Benders' decomposition based exact solution method for multi-manned assembly line balancing problem with walking workers. *Annals of Operations Research*, 321, 507–540.
- Scholl, A. & Boysen, N. (2009). Designing parallel assembly lines with split workplaces: Model and optimization procedure. *International Journal of Production Economics*, 119 (1), 90-100.
- Scholl, A., Boysen, N. & Fliedner, M. (2008). The sequence-dependent assembly line balancing problem. *Operations Research Spectrum*, 30, 579-609.
- Scholl, A. & Voß, S. (1996). Simple assembly line balancing-Heuristic approaches. *Journal of Heuristics*, 2, 217–244.
- Sepahi, A. & Jalali Naini, S. G. (2014). Multi-manned assembly line balancing problem with variable task times. *European Journal of Academic Essays*, 1(4), 68-75.
- Yang, W. & Cheng, W. (2020). A mathematical model and a simulated annealing algorithm for balancing multi-manned assembly line balancing problem with sequence-dependent setup time. *Mathematical problems in Engineering*, 8510253.
- Yang, H., Lee, J.-H., Lee, S. H., Lee, S. G., Kim, H. R. & Kim, H.-J. (2022). A Multi-Manned Assembly Line Worker Assignment and Balancing Problem With Positional Constraints.

IEEE Robotics and Automation Letters, 7(3), 7786-7793.

Yazdanparast, V. & Hajhosseini, H. (2011). Multi-manned production lines with labour concentration. *Australian Journal of Basic and Applied Sciences*, 5, 839-846.

Yilmaz, H. & Yilmaz, M. (2015). Multi-manned assembly line balancing problem with balanced load density. *Assembly Automation*, 35(1), 137-142.

Yilmaz, H. & Yilmaz, M. (2016a). A multi-manned assembly line balancing problem with classified teams: a new approach. *Assembly Automation*, 36(1), 51-59.

Yilmaz, H. & Yilmaz, M. (2016b). Note to: a mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *International Journal of Advanced Manufacturing Technology*, 89, 1935–1939.

Zamzam, N. & Elakkad, A. (2021). Time and space multi-manned assembly line balancing problem using genetic algorithm. *Journal of Industrial Engineering and Management*, 14(4), 733-749.

Zamzam, N., Sadek, Y., Afia, N. & El-Kharbotly, A. (2015). Multi-manned assembly line balancing using genetic algorithm. *International Journal of Engineering Research & Technology*, 4(12), 56-61.

Zhang, Z., Tang, Q. & Chica, M. (2020). Multi-manned assembly line balancing with time and space constraints: A MILP model and memetic ant colony system. *Computers & Industrial Engineering*, 150, 106862.

Zhang, Z., Tang, Q., Li, Z. & Han, D. (2021). An efficient migrating birds optimization algorithm with idle time reduction for Type-I multi-manned assembly line balancing problem. *Journal of Systems Engineering and Electronics*, 32(2), 286-296.

Apéndice

Apéndice 1 – Ejemplares de Otto *et al.* (2013) de calibración y entrenamiento

El conjunto de ejemplares de calibración y entrenamiento está compuesto por las siguientes instancias de Otto *et al.* (2013):

- 25 instancias de 50 tareas; ejemplares números: 13, 34, 46, 61, 65, 94, 102, 108, 111, 118, 121, 145, 149, 181, 199, 218, 224, 243, 252, 262, 266, 268, 271, 281, 294.
- 25 instancias de 100 tareas; ejemplares números: 11, 30, 79, 82, 89, 92, 95, 102, 118, 148, 155, 170, 171, 185, 216, 226, 232, 240, 242, 254, 274, 295, 297, 504, 509.

Para cada uno de los ejemplares se considera que:

- Los tiempos de proceso de las tareas se incrementan en una unidad de tiempo por cada trabajador adicional en la estación.
- Cada estación puede tener un máximo de $UB^O = 4$ trabajadores.
- El tiempo de resolución máximo es de 3600 segundos.
- Adicionalmente, para las experiencias computacionales del SMALBP-DTT2 se considera que el número máximo de estaciones UB^M es igual al número de estaciones encontradas con la heurística HEUR_PART.

Estas 50 instancias se utilizan con los siguientes objetivos:

- Para las experiencias computacionales previas de los dos procedimientos heurísticos de Relajar y Fijar (sección 5.4.1), HEUR_PART (sección 5.4.1), MULTI_HEUR_PART (secciones 5.8 y 5.9.1) y OPWK_PART (sección 5.9.1) que resuelven el SMALBP-DTT1.
- Como conjunto de entrenamiento T para diseñar las reglas de prioridad mediante EAGH (sección 5.6.1) y EAGH-CKTL (sección 5.6.2).
- Para las experiencias computacionales previas de los procedimientos heurísticos HEUR_CT_PART (sección 6.4.1) y OPCT_PART (sección 6.4.1) que resuelven el SMALBP-DTT2.

Apéndice 2 – Resultados de la experiencia computacional de MILP-DTT1, 1RF, 2RF, HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE

Los resultados de los experimentos computacionales de MILP-DTT1, 1RF, 2RF, HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE, al resolver las 250 instancias de Otto *et al.* (2013), se representan en la Tabla 39 (siendo "Trb" el número de trabajadores y "Est" el número de estaciones).

Como se puede comprobar en la Tabla 39, los procedimientos HEUR_PART y HEUR_PART_SGL, obtienen los mismos resultados para todos los ejemplares. La misma circunstancia se presenta con HEUR_PART_OTB y HEUR_PART_OTB_SGL.

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_SGL		HEUR_PART_OTB y HEUR_PART_OTB_SGL		HEUR_PART_SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
50	12	7	7	7	7	7	7	6	5	6	5	7	2
50	153	8	3	8	8	8	8	8	3	8	3	8	3
50	231	8	8	8	8	8	8	7	4	7	4	8	3
50	244	8	3	8	8	8	8	7	6	7	6	8	3
50	167	8	4	8	8	8	8	8	2	8	2	8	2
50	170	8	8	8	8	8	8	8	2	8	2	8	2
50	173	8	8	8	8	8	8	8	2	8	2	8	2
50	232	8	4	8	8	8	8	8	3	8	3	8	3
50	236	8	3	8	8	8	8	8	3	8	3	8	3
50	239	8	3	8	8	8	8	8	3	8	3	8	4
50	249	8	3	8	8	8	8	8	4	8	4	8	4
50	90	8	4	8	8	8	8	8	3	8	3	8	3
50	92	7	7	7	7	7	7	7	3	7	3	7	3
50	97	7	7	7	7	7	7	7	5	7	5	7	5
50	79	9	3	9	9	9	9	8	5	8	5	9	3
50	168	9	9	9	9	9	9	9	3	9	3	9	3
50	202	10	4	10	10	10	10	10	3	10	3	10	4
50	204	11	11	11	11	11	11	11	3	11	3	11	3
50	288	11	11	11	11	11	11	11	5	11	5	11	5
50	298	11	11	11	11	11	11	11	5	11	6	11	6
50	75	12	12	12	12	12	11	12	3	12	3	12	3
50	139	12	12	12	12	12	12	12	5	12	5	12	5
50	150	11	11	11	11	11	11	11	5	11	5	11	5
50	206	12	12	12	12	12	12	12	4	12	4	12	4
50	221	11	11	11	11	11	11	11	4	11	4	11	4
50	223	11	11	11	11	11	11	11	5	11	5	11	5
50	286	12	6	12	12	12	12	12	6	12	6	12	6
50	289	12	7	12	12	12	12	12	5	12	5	12	5
50	216	12	12	12	12	12	12	12	4	12	6	12	6
50	278	13	13	13	13	13	13	13	5	13	5	13	5

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_ SGL		HEUR_PART _OTB y HEUR_PART _OTB_SGL		HEUR_PART _SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
50	283	13	5	13	13	13	13	13	5	13	5	13	5
50	287	13	13	13	13	13	13	13	6	13	6	13	6
50	66	12	12	12	12	12	12	12	4	12	4	12	4
50	128	13	7	13	13	13	13	13	6	13	6	13	6
50	132	13	7	13	13	13	13	13	6	13	6	13	6
50	140	13	7	13	13	13	13	13	6	13	6	13	6
50	143	12	12	12	12	12	12	12	7	12	7	12	7
50	276	13	7	13	13	13	13	13	6	13	6	13	6
50	53	13	13	13	13	13	13	13	4	13	4	13	4
50	60	13	13	13	13	13	12	13	4	13	4	13	4
50	67	13	13	13	13	13	13	13	5	13	5	13	5
50	69	12	12	12	12	12	12	12	5	12	5	12	5
50	146	13	13	13	13	13	13	13	7	13	7	13	7
50	208	14	14	14	14	14	14	13	5	13	5	14	4
50	135	14	14	14	14	14	14	14	5	14	5	14	5
50	141	13	13	13	13	13	13	13	7	13	7	13	7
50	144	13	13	13	13	13	13	13	7	13	7	13	7
50	147	13	13	13	13	13	13	13	7	13	7	13	7
50	210	13	13	13	13	13	13	13	5	13	5	13	5
50	217	14	14	14	14	14	14	14	4	14	4	14	4
50	277	13	13	13	13	13	13	13	7	13	7	13	7
50	62	13	13	13	13	13	13	13	5	13	5	13	5
50	71	13	13	13	13	13	13	13	5	13	5	13	5
50	127	14	14	14	14	14	14	14	7	14	8	14	8
50	134	14	14	14	14	14	14	14	7	14	7	14	7
50	295	16	16	16	16	16	16	16	10	16	10	16	10
50	119	27	27	27	27	27	27	26	14	27	12	27	12
50	179	28	28	28	28	28	28	27	9	27	9	28	9
50	185	28	28	28	28	28	28	27	12	27	12	28	11
50	186	27	27	27	27	27	27	26	13	26	13	27	12
50	189	27	27	27	27	27	27	26	13	26	13	27	8
50	269	29	29	29	29	29	29	28	12	28	12	29	10
50	180	27	27	27	27	27	27	27	8	27	9	27	9
50	40	26	26	26	26	26	26	26	11	26	11	26	11
50	196	29	29	29	29	29	29	28	13	29	11	29	11
50	265	28	28	28	28	28	28	28	12	28	12	28	12
50	272	28	28	28	28	28	28	28	12	28	12	28	12
50	112	27	27	27	27	27	27	27	10	27	10	27	10
50	114	28	28	28	28	28	28	28	12	28	12	28	12
50	120	28	28	28	28	28	28	28	13	28	13	28	13
50	176	28	28	28	28	28	28	28	9	28	11	28	11
50	26	27	27	27	27	27	27	27	11	27	13	27	13
50	264	27	27	27	27	27	27	27	14	27	14	27	14
50	275	28	28	28	28	28	28	28	16	28	16	28	16
50	107	29	29	29	29	29	29	29	12	29	13	29	13
50	113	29	29	29	29	29	29	29	12	29	12	29	12
50	183	29	29	29	29	29	29	29	10	29	11	29	11
50	193	30	30	30	30	30	30	30	11	30	11	30	11
50	195	29	29	29	29	29	29	29	10	29	10	29	10
50	198	28	28	28	28	28	28	28	12	28	12	28	12

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_SGL		HEUR_PART_OTB y HEUR_PART_OTB_SGL		HEUR_PART_SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
50	270	29	29	29	29	29	29	29	13	29	13	29	13
50	47	28	28	28	28	28	28	28	14	28	14	28	14
50	29	29	29	29	29	29	29	29	12	29	12	29	12
50	109	30	30	30	30	30	30	30	11	30	11	30	11
50	123	32	32	32	32	32	32	32	11	32	12	32	12
50	184	38	38	38	38	38	38	38	13	38	16	38	16
50	42	24	24	24	24	24	24	24	10	24	14	24	14
50	105	25	25	25	25	25	25	25	9	25	9	25	9
50	188	25	25	25	25	25	25	25	7	25	7	25	7
50	44	27	27	27	27	27	27	26	7	26	7	27	7
50	45	27	27	27	27	27	27	26	11	26	13	27	9
50	49	26	26	26	26	26	26	25	9	25	10	26	8
50	33	25	25	25	25	25	25	25	9	25	10	25	10
50	200	26	26	26	26	26	26	26	9	26	12	26	12
50	41	27	27	27	27	27	27	26	11	27	13	27	13
50	43	27	27	27	27	27	27	26	11	26	11	27	8
50	32	26	26	26	26	26	26	26	11	26	11	26	11
50	187	27	27	27	27	27	27	27	9	27	9	27	9
50	261	29	29	29	29	29	29	28	13	28	13	29	14
50	104	28	28	28	28	28	28	28	11	28	11	28	11
50	273	29	29	29	29	29	29	29	12	29	12	29	12
50	117	28	28	28	28	28	28	28	11	28	13	28	13
50	50	28	28	28	28	28	28	27	9	27	9	28	8
50	110	28	28	28	28	28	28	28	10	28	10	28	10
50	182	27	27	27	27	27	27	27	11	27	11	27	11
50	115	30	30	30	30	30	30	28	11	28	11	30	10
50	30	28	28	28	28	28	28	27	11	27	11	28	10
50	251	28	28	28	28	28	28	28	14	28	14	28	14
50	258	29	29	29	29	29	29	29	13	29	13	29	13
50	177	29	29	29	29	29	29	28	9	28	9	29	8
50	178	29	29	29	29	29	29	28	11	28	11	29	8
50	192	29	29	29	29	29	29	28	13	28	13	29	11
50	48	28	28	28	28	28	28	28	9	28	13	28	13
50	106	28	28	28	28	28	28	28	13	28	13	28	13
50	191	29	29	29	29	29	29	29	11	29	11	29	11
50	31	28	28	28	28	28	28	28	7	28	7	28	7
50	267	31	10	31	31	31	31	30	12	30	12	31	11
50	253	29	29	29	29	29	29	29	15	29	15	29	15
50	255	31	31	31	31	31	31	31	13	31	13	31	13
50	194	29	29	29	29	29	29	29	10	29	12	29	12
50	197	30	30	30	30	30	30	29	10	29	12	30	8
50	28	29	29	29	29	29	29	28	12	28	12	29	9
50	260	29	29	29	29	29	29	29	17	29	17	29	17
50	122	31	31	31	31	31	31	30	13	30	13	31	13
50	124	30	30	30	30	30	30	30	14	30	14	30	14
100	257	13	13	13	13	13	9	13	4	13	4	13	4
100	114	13	13	13	13	13	13	13	9	13	9	13	9
100	181	13	13	13	13	13	13	13	7	13	7	13	7
100	190	13	13	13	13	13	13	13	10	13	10	13	10
100	187	14	14	14	14	14	14	14	4	14	4	14	4

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_SGL		HEUR_PART_OTB y HEUR_PART_OTB_SGL		HEUR_PART_SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
100	192	14	14	14	14	14	14	14	5	14	5	14	5
100	266	14	14	14	14	14	10	14	5	14	5	14	5
100	27	14	14	14	14	14	14	14	5	14	5	14	5
100	273	14	14	14	14	14	14	14	5	14	5	14	5
100	112	14	14	14	14	14	14	14	5	14	5	14	5
100	198	14	14	14	14	14	14	14	5	14	5	14	5
100	271	14	14	14	14	14	14	14	5	14	5	14	5
100	252	14	14	14	14	14	14	14	6	14	6	14	6
100	29	14	14	14	14	14	14	14	9	14	9	14	9
100	40	15	15	15	15	15	15	14	7	14	7	15	5
100	186	14	14	14	14	14	14	14	8	14	8	14	8
100	258	15	15	15	14	15	15	15	5	15	5	15	5
100	28	15	15	15	15	15	15	15	4	15	4	15	4
100	43	15	15	15	15	15	15	15	5	15	5	15	5
100	108	15	15	15	15	15	15	15	5	15	5	15	5
100	26	15	15	15	15	15	15	15	4	15	4	15	4
100	36	15	15	15	15	15	15	15	4	15	4	15	4
100	259	15	15	15	15	15	15	15	8	15	8	15	8
100	101	15	15	15	15	15	15	15	9	15	11	15	11
100	117	16	16	16	16	16	16	16	6	16	6	16	6
100	124	16	16	16	16	16	16	16	6	16	6	16	6
100	116	16	16	16	16	16	16	16	6	16	6	16	6
100	248	20	20	20	20	20	20	20	8	20	8	20	8
100	18	20	20	20	20	20	20	20	5	20	5	20	5
100	77	20	20	20	20	20	20	20	8	20	8	20	9
100	90	21	21	21	21	21	21	21	8	21	8	21	8
100	14	21	21	21	21	21	21	21	6	21	6	21	6
100	81	21	21	21	21	21	21	21	7	21	8	21	8
100	169	21	21	21	21	21	21	21	8	21	9	21	9
100	21	21	21	21	21	21	21	21	7	21	8	21	8
100	151	22	22	22	22	22	22	22	6	22	6	22	6
100	17	22	22	22	22	22	22	22	7	22	7	22	7
100	244	22	22	22	22	22	22	22	9	22	9	22	9
100	168	22	22	22	22	22	22	22	6	22	6	22	6
100	249	22	22	22	22	22	22	22	8	22	8	22	8
100	78	22	22	22	22	22	22	22	8	22	8	22	8
100	160	22	22	22	22	22	22	22	8	22	8	22	8
100	5	22	22	22	22	22	22	22	6	22	7	22	7
100	6	22	22	22	22	22	22	22	7	22	7	22	7
100	87	22	22	22	22	22	22	22	10	22	10	22	10
100	152	22	22	22	22	22	22	22	7	22	12	22	12
100	25	23	23	23	23	23	23	22	9	22	9	23	6
100	83	22	22	22	22	22	22	22	8	22	8	22	8
100	99	22	22	22	22	22	22	22	7	22	7	22	7
100	162	23	23	23	23	23	23	23	6	23	7	23	7
100	228	23	23	23	23	23	23	23	9	23	9	23	9
100	247	23	23	23	23	23	23	23	8	23	8	23	8
100	80	23	23	23	23	23	23	23	7	23	7	23	7
100	161	23	23	23	23	23	23	23	6	23	6	23	6
100	174	23	23	23	23	23	23	23	7	23	7	23	7

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_SGL		HEUR_PART_OTB y HEUR_PART_OTB_SGL		HEUR_PART_SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
100	241	23	23	23	23	23	23	23	8	23	10	23	10
100	94	23	23	23	23	23	23	23	8	23	8	23	8
100	156	23	23	23	23	23	23	23	7	23	7	23	7
100	234	23	23	23	23	23	23	23	13	23	13	23	13
100	164	23	23	23	23	23	23	23	7	23	9	23	9
100	76	23	23	23	23	23	23	23	8	23	9	23	9
100	86	23	23	23	23	23	23	23	10	23	10	23	10
100	19	23	23	23	23	23	23	23	7	23	7	23	7
100	1	24	24	24	24	24	24	24	6	24	6	24	6
100	230	24	24	24	24	24	24	24	12	24	13	24	13
100	88	24	24	24	24	24	24	24	8	24	8	24	8
100	243	24	24	24	24	24	24	24	8	24	8	24	8
100	9	24	24	24	24	24	24	24	6	24	6	24	6
100	172	24	24	24	24	24	24	24	8	24	8	24	8
100	23	24	24	24	24	24	24	24	8	24	10	24	10
100	250	25	25	25	25	25	25	24	10	24	10	25	9
100	4	24	24	24	24	24	24	24	9	24	9	24	9
100	8	24	24	24	24	24	24	24	8	24	9	24	9
100	166	24	24	24	24	24	24	24	13	24	13	24	13
100	13	24	24	24	24	24	24	24	7	24	11	24	11
100	245	24	24	24	24	24	24	24	9	24	9	24	9
100	165	25	25	25	25	25	25	25	7	25	9	25	9
100	22	25	25	25	25	25	25	25	7	25	8	25	8
100	85	25	25	25	25	25	25	25	9	25	9	25	9
100	163	25	25	25	25	25	25	25	8	25	8	25	8
100	91	25	25	25	25	25	25	25	10	25	10	25	10
100	173	25	25	25	25	25	25	25	8	25	8	25	8
100	100	26	26	26	26	26	26	26	8	26	8	26	8
100	12	26	26	26	26	26	26	26	7	26	7	26	7
100	157	26	26	26	26	26	26	26	8	26	9	26	9
100	7	26	26	26	26	26	26	26	8	26	8	26	8
100	397	26	26	26	26	26	26	26	11	26	11	26	11
100	235	27	27	27	27	27	27	27	10	27	10	27	10
100	246	27	27	27	27	27	27	27	11	27	11	27	11
100	227	27	27	27	27	27	27	27	11	27	11	27	11
100	84	27	27	27	27	27	27	27	10	27	10	27	10
100	175	27	27	27	27	27	27	27	8	27	8	27	8
100	93	28	28	28	28	28	28	28	9	28	9	28	9
100	144	50	50	50	50	50	50	50	17	50	22	50	22
100	215	50	50	50	50	50	50	50	24	50	24	50	24
100	204	52	52	52	52	52	52	52	18	52	26	52	26
100	141	52	52	52	52	52	52	52	18	52	23	52	23
100	206	52	52	52	52	52	52	52	20	52	27	52	27
100	146	53	53	53	53	53	53	53	24	53	24	53	24
100	223	53	53	53	53	53	53	53	17	53	26	53	26
100	210	53	53	53	53	53	53	53	25	53	25	53	25
100	436	54	54	54	54	54	54	54	19	54	19	54	19
100	139	54	54	54	54	54	54	54	18	54	21	54	21
100	442	55	55	55	55	55	55	53	29	55	24	55	24
100	291	56	56	56	56	56	56	55	23	55	23	56	22

Número de tareas	Instancia	MILP-DTT1		1RF		2RF		HEUR_PART y HEUR_PART_SGL		HEUR_PART_OTB y HEUR_PART_OTB_SGL		HEUR_PART_SE	
		Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est	Trb	Est
100	293	55	55	55	55	55	55	55	24	55	24	55	24
100	137	56	56	56	56	56	56	55	26	55	28	56	22
100	143	55	55	55	55	55	55	55	22	55	22	55	22
100	214	55	55	55	55	55	55	55	27	55	29	55	29
100	300	56	56	56	56	56	56	56	24	56	24	56	24
100	449	57	57	57	57	57	57	56	24	56	24	57	23
100	279	56	56	56	56	56	56	56	29	56	30	56	30
100	444	57	57	57	57	57	57	55	25	57	24	57	24
100	285	58	58	58	58	58	58	57	22	57	22	58	21
100	290	57	57	57	57	57	57	56	21	57	25	57	25
100	288	59	59	59	59	59	59	58	33	58	33	59	23
100	432	59	59	59	59	59	59	58	21	58	27	59	22
100	448	59	59	59	59	59	59	57	26	58	24	59	23
100	286	59	59	59	59	59	59	59	23	59	27	59	27
100	434	60	60	60	60	60	60	59	19	60	24	60	24
100	278	60	60	60	60	60	60	60	22	60	30	60	30
100	298	61	61	61	61	61	61	61	26	61	26	61	26
100	282	61	61	61	61	61	61	61	33	61	33	61	33
100	147	61	61	61	61	61	61	60	23	61	22	61	22
100	289	63	63	63	63	63	63	63	31	63	31	63	31

Tabla 39. Resultados de los experimentos computacionales con instancias de Otto *et al.* (2013) para MILP-DTT1, 1RF, 2RF, HEUR_PART, HEUR_PART_SGL, HEUR_PART_OTB, HEUR_PART_OTB_SGL y HEUR_PART_SE. Fuente: propia

Apéndice 3 – Resultados de la experiencia computacional de EAGH y EAGH-CKTL

Los resultados de los experimentos computacionales con las 250 instancias de Otto *et al.* (2013) para las seis reglas de prioridad propuestas al utilizar EAGH y EAGH-CKTL (R13-R18), y las tres reglas de prioridad utilizadas en HEUR_PART que presentan mejores resultados (R1, R2 y R11), se representan en la Tabla 40 (siendo "Tr" el número de trabajadores y "Es" el número de estaciones).

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
50	12	7	3	7	3	7	3	6	4	6	5	7	2	6	5	6	5	6	5
50	153	8	3	8	3	8	3	8	3	8	3	8	3	8	3	8	2	8	4
50	231	8	4	8	5	8	4	8	4	7	4	8	4	8	4	8	3	8	4
50	244	8	5	7	6	7	6	7	5	8	5	7	6	7	6	7	6	7	6
50	167	8	3	8	3	8	3	8	3	8	3	8	3	8	3	8	3	8	3
50	170	8	2	8	3	8	3	8	3	8	3	8	2	8	2	8	3	8	3
50	173	8	3	8	3	8	3	8	2	8	2	8	2	8	3	8	3	8	3
50	232	8	4	8	3	8	3	8	4	8	4	8	4	8	5	8	3	8	5
50	236	8	4	8	4	8	4	8	4	8	4	8	3	8	4	8	5	8	4

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
50	239	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8	5	8	5
50	249	8	5	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8	4
50	90	8	3	8	4	8	4	8	5	8	4	8	3	8	4	8	3	8	3
50	92	7	4	7	4	7	4	8	3	7	4	7	4	7	4	8	3	8	3
50	97	8	4	8	4	8	4	8	4	7	5	8	4	7	5	7	5	7	5
50	79	9	5	9	4	9	4	9	3	9	4	9	3	9	3	9	4	9	4
50	168	9	4	9	3	9	3	9	3	9	3	9	4	9	3	9	3	9	3
50	202	10	5	10	5	10	4	10	5	10	4	10	5	10	5	10	6	10	4
50	204	11	4	11	5	11	5	11	5	11	4	11	4	11	5	11	7	11	4
50	288	11	5	11	6	11	6	11	7	11	6	11	6	11	7	11	7	11	6
50	298	11	5	11	6	12	5	12	8	11	7	11	5	12	6	12	7	12	7
50	75	12	6	12	7	12	7	12	7	12	5	12	4	12	6	12	5	12	5
50	139	12	5	12	7	12	7	12	6	12	5	12	5	12	5	12	7	12	7
50	150	11	5	11	6	11	7	11	6	11	5	12	5	11	5	11	5	11	5
50	206	12	5	12	5	12	4	12	5	12	4	12	5	12	4	12	5	12	7
50	221	12	3	12	7	12	7	12	4	12	3	11	4	12	4	12	5	12	5
50	223	12	4	12	5	12	5	12	5	12	5	12	4	11	5	11	5	11	5
50	286	12	6	12	7	12	7	12	7	12	6	12	6	12	6	13	8	12	7
50	289	12	5	12	5	12	5	12	7	12	6	12	5	12	5	12	6	12	5
50	216	12	5	13	4	12	6	12	5	12	6	12	5	12	4	12	4	12	4
50	278	13	6	13	6	13	6	13	7	13	6	13	6	13	8	13	7	13	8
50	283	13	6	13	10	13	10	13	8	13	7	13	8	13	7	13	10	13	8
50	287	13	7	13	8	13	9	13	8	13	9	13	7	13	7	14	7	13	7
50	66	12	4	13	7	13	6	12	5	12	5	12	5	12	5	13	8	13	6
50	128	13	6	13	7	13	7	13	8	13	7	13	6	13	8	13	9	13	6
50	132	13	7	13	6	13	6	13	7	13	7	13	6	13	7	13	7	13	6
50	140	13	7	13	7	13	7	13	6	13	8	13	7	13	8	13	8	13	8
50	143	13	7	12	7	12	8	13	6	12	6	12	6	13	7	13	7	13	6
50	276	13	6	13	8	13	8	13	8	13	7	13	6	13	7	13	8	13	7
50	53	13	4	13	6	13	6	13	6	13	5	13	4	13	5	14	6	13	5
50	60	13	4	13	7	13	7	13	6	13	4	13	4	13	5	13	7	13	7
50	67	13	5	13	7	13	7	12	8	13	5	13	5	13	5	13	5	13	5
50	69	12	5	13	6	13	6	13	6	12	5	12	5	12	6	13	7	13	7
50	146	13	8	14	7	14	7	14	7	13	8	13	8	14	7	14	10	14	8
50	208	14	5	14	4	14	4	14	5	13	7	13	5	14	5	14	6	14	5
50	135	14	6	14	6	14	6	14	7	14	6	14	5	14	5	15	7	14	6
50	141	13	7	14	7	14	7	14	8	13	7	13	7	13	7	16	6	14	7
50	144	13	7	13	7	14	7	14	7	13	7	13	7	13	7	13	7	13	7
50	147	13	7	14	8	13	7	14	9	13	7	13	7	13	7	15	9	14	7
50	210	13	7	13	5	14	7	13	7	13	6	13	7	13	7	15	6	13	7
50	217	14	5	14	5	14	7	14	5	14	6	14	5	14	4	14	8	14	9
50	277	14	7	13	7	13	7	13	7	13	7	14	7	13	7	13	7	13	7
50	62	13	5	13	6	13	5	13	6	13	5	13	5	13	5	13	5	13	5
50	71	13	5	13	6	13	6	13	5	13	6	13	5	13	5	14	7	13	5
50	127	14	8	14	9	14	9	15	8	14	8	14	8	14	8	15	8	15	9
50	134	14	7	15	9	15	8	15	8	14	7	15	7	14	9	15	6	15	8
50	295	17	8	16	10	16	10	16	9	16	10	16	9	16	10	16	10	16	10
50	119	28	12	27	12	27	12	27	12	26	13	27	12	26	14	26	14	26	14
50	179	28	11	28	10	27	9	29	11	27	9	28	11	29	12	31	13	29	11
50	185	29	13	29	16	27	12	28	11	29	9	29	12	30	9	28	12	29	11
50	186	28	12	28	12	26	13	29	14	28	11	28	12	29	11	29	14	29	11
50	189	26	13	27	12	27	13	28	12	26	10	27	9	27	8	27	13	28	13

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
50	269	29	11	28	14	28	14	28	14	28	12	29	11	28	12	28	12	28	12
50	180	27	10	27	10	27	9	26	11	27	11	27	14	27	9	27	11	28	14
50	40	27	8	27	12	26	11	28	13	26	11	27	8	27	9	27	11	27	14
50	196	28	15	30	15	29	11	30	10	29	11	29	12	29	13	29	12	29	10
50	265	28	13	28	14	28	13	29	13	27	16	28	13	28	12	29	14	29	13
50	272	29	14	28	16	28	17	28	13	28	14	29	14	29	13	29	16	28	14
50	112	29	11	28	12	27	10	28	11	28	11	29	11	29	11	28	13	28	10
50	114	30	11	29	13	28	12	29	11	30	10	29	11	30	11	30	10	29	11
50	120	29	12	28	13	28	14	27	15	28	13	29	12	27	14	28	13	28	13
50	176	28	11	28	10	28	12	29	13	28	11	28	8	28	9	29	12	29	14
50	26	29	10	27	11	27	13	30	11	29	14	29	10	29	11	28	13	29	11
50	264	29	17	27	14	27	15	29	10	29	17	29	16	29	15	28	15	29	14
50	275	30	13	30	20	28	16	31	17	30	14	29	14	29	15	28	15	30	19
50	107	30	14	29	15	29	13	29	13	29	15	29	13	29	15	29	12	30	14
50	113	30	14	29	12	29	12	30	11	30	15	30	14	30	14	30	11	29	11
50	183	29	10	30	14	29	11	29	14	29	10	29	12	29	10	29	10	29	10
50	193	30	12	30	16	30	11	31	10	30	12	30	12	30	10	29	12	31	11
50	195	29	12	29	13	29	10	29	9	29	10	29	11	29	10	28	14	29	10
50	198	29	10	28	12	28	12	29	16	29	12	29	10	29	12	29	11	29	13
50	270	29	16	29	13	29	15	29	14	29	15	28	15	29	13	29	13	28	14
50	47	28	14	29	13	29	12	29	11	28	12	28	14	28	15	29	14	30	10
50	29	30	12	29	12	29	15	30	13	30	13	30	14	30	14	29	12	29	13
50	109	30	11	30	13	30	11	30	12	30	11	30	11	30	11	31	11	30	12
50	123	32	12	32	12	32	13	33	13	32	14	32	13	32	15	33	14	33	11
50	184	39	17	38	17	38	16	38	18	38	17	39	17	39	16	39	18	39	19
50	42	25	10	24	14	24	10	25	11	25	8	25	8	25	10	25	13	25	10
50	105	26	8	25	11	25	11	26	11	25	13	26	8	26	9	26	12	25	11
50	188	25	9	25	7	25	10	25	11	25	7	25	10	25	9	26	12	25	10
50	44	26	11	26	9	26	10	25	10	26	8	26	11	26	9	26	10	27	8
50	45	27	10	26	12	26	13	26	12	26	10	26	12	26	11	26	11	26	11
50	49	26	9	25	9	25	10	26	12	26	11	26	9	26	12	27	11	26	10
50	33	25	10	26	11	25	10	26	12	25	9	25	11	25	11	27	14	26	13
50	200	26	9	26	11	26	12	26	8	26	10	26	9	26	9	26	9	26	9
50	41	27	13	27	12	27	13	27	12	27	12	27	12	27	11	28	12	28	9
50	43	26	11	27	11	26	11	26	10	27	11	26	11	26	13	26	11	26	11
50	32	27	11	26	11	26	14	27	9	26	11	27	9	26	11	26	11	26	11
50	187	27	9	27	13	27	9	29	9	27	9	27	9	28	9	28	13	28	10
50	261	29	15	29	16	28	15	31	17	28	15	28	15	29	15	30	16	29	16
50	104	30	12	29	15	28	11	30	14	28	11	30	12	29	12	30	11	29	14
50	273	29	15	29	12	29	13	28	15	29	14	28	15	27	18	29	14	31	13
50	117	29	13	28	12	28	13	28	12	27	14	28	13	27	13	28	13	28	16
50	50	28	14	27	10	27	9	28	11	28	13	28	9	28	10	28	10	29	9
50	110	28	10	28	12	28	12	28	11	27	12	28	10	28	10	28	11	28	10
50	182	28	10	27	13	27	11	28	14	28	9	29	11	27	11	28	14	28	14
50	115	30	12	29	15	28	11	28	12	28	11	29	12	28	11	28	11	28	11
50	30	29	14	27	11	28	13	29	12	27	14	28	12	29	9	29	11	28	10
50	251	31	15	28	14	29	13	31	14	29	14	27	15	27	15	29	14	29	12
50	258	29	15	30	12	30	15	30	16	30	13	29	14	29	15	31	14	29	15
50	177	28	9	29	12	28	12	28	11	28	10	28	9	29	12	29	11	28	10
50	178	29	8	28	11	29	12	28	13	28	11	29	10	28	9	29	11	29	12
50	192	29	16	28	16	28	13	29	13	28	16	29	16	27	13	28	13	29	12
50	48	28	10	28	13	28	11	28	11	28	10	29	9	28	9	28	9	28	9

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
50	106	29	10	28	13	29	11	30	13	28	10	29	10	30	11	30	14	29	12
50	191	29	11	29	12	30	10	30	16	29	12	30	10	29	16	31	14	30	11
50	31	28	12	28	12	28	12	28	12	28	12	28	12	28	14	28	9	28	13
50	267	30	12	31	13	31	12	31	14	30	12	29	12	31	12	31	13	30	14
50	253	29	16	30	18	29	15	31	16	30	14	29	16	28	16	30	13	29	16
50	255	33	16	31	16	31	15	31	15	30	16	31	13	30	15	31	13	31	13
50	194	29	14	29	12	29	13	28	16	28	11	29	14	29	12	30	16	29	12
50	197	30	14	30	11	29	12	28	14	29	10	30	8	28	14	29	10	29	10
50	28	29	11	30	14	28	12	29	11	28	10	29	11	29	10	28	12	28	10
50	260	31	11	30	13	29	17	31	12	29	12	31	11	31	11	30	12	30	11
50	122	31	14	30	14	30	13	30	12	30	14	31	14	29	14	30	13	29	14
50	124	31	13	30	14	31	13	32	13	30	13	31	13	29	14	31	14	31	12
100	257	13	6	13	6	13	6	13	7	13	5	13	6	13	5	13	8	13	8
100	114	13	9	14	6	14	5	14	4	13	8	14	6	13	8	14	5	14	5
100	181	13	7	13	9	13	7	13	7	13	7	13	7	14	5	14	6	14	5
100	190	14	7	13	10	13	10	14	5	13	10	13	10	14	5	14	6	14	6
100	187	14	5	14	8	14	8	14	6	14	4	14	5	14	6	14	7	14	6
100	192	14	8	14	8	14	8	14	6	14	5	14	5	14	6	14	7	14	8
100	266	14	7	14	10	14	8	14	8	14	7	14	7	14	6	14	8	14	10
100	27	14	8	14	8	14	9	14	5	14	6	14	5	14	5	14	5	14	7
100	273	14	7	14	5	14	5	14	10	14	7	14	7	14	8	14	7	14	7
100	112	14	6	14	6	14	6	14	5	14	6	14	6	14	6	14	5	14	7
100	198	14	5	14	8	14	8	14	8	14	6	14	5	14	6	14	5	14	7
100	271	14	6	14	7	14	6	14	6	14	6	14	6	14	5	14	5	14	5
100	252	14	7	14	7	14	7	15	8	14	7	14	7	15	6	15	6	15	7
100	29	15	7	15	10	14	9	14	10	14	9	15	8	14	9	14	9	14	9
100	40	15	5	14	8	14	8	15	5	14	8	15	5	14	7	15	7	15	8
100	186	15	5	14	8	14	10	14	9	15	6	15	5	14	8	14	8	14	8
100	258	15	6	15	6	15	6	15	7	15	7	15	6	15	7	15	8	15	7
100	28	15	4	15	9	15	9	15	6	15	5	15	4	15	6	15	8	15	6
100	43	15	7	15	7	15	7	15	6	15	8	15	7	15	8	15	7	15	7
100	108	15	6	15	8	15	8	15	7	15	7	15	8	15	6	15	5	15	6
100	26	15	5	15	8	15	7	15	5	15	8	15	5	15	5	15	6	15	5
100	36	15	7	15	6	15	7	15	7	15	8	15	5	15	6	15	5	15	7
100	259	16	6	15	8	15	8	15	6	16	6	15	8	15	8	15	8	15	8
100	101	15	10	16	6	16	6	16	8	15	11	15	11	15	9	15	9	15	9
100	117	16	8	16	6	16	6	16	6	16	8	16	8	16	7	16	5	16	6
100	124	16	7	16	8	16	8	16	7	16	7	16	7	16	7	16	7	16	7
100	116	16	6	17	6	17	7	16	6	16	6	16	6	16	5	16	6	16	6
100	248	20	8	20	11	20	10	21	14	20	8	21	9	20	9	21	15	20	10
100	18	20	7	20	8	20	8	20	7	20	7	20	7	20	6	20	8	20	9
100	77	20	9	21	9	21	8	20	9	20	10	20	11	20	8	20	8	20	8
100	90	21	9	21	10	21	11	22	10	21	9	21	9	21	8	21	9	21	10
100	14	21	8	21	10	21	10	21	8	21	8	21	8	21	7	21	9	21	9
100	81	21	9	21	11	21	11	21	8	21	9	21	9	21	10	21	8	21	9
100	169	21	8	21	12	21	12	21	9	21	8	21	9	21	8	22	12	22	12
100	21	21	9	21	13	21	13	21	10	21	10	21	9	21	7	21	7	21	7
100	151	22	7	22	10	22	12	22	7	22	8	22	8	22	7	22	8	22	8
100	17	22	8	22	10	22	10	22	9	22	7	22	8	22	7	23	9	22	7
100	244	22	12	22	13	22	13	22	11	22	11	22	11	22	11	23	11	22	14
100	168	22	7	23	15	22	12	22	7	22	7	22	7	22	6	22	6	22	6
100	249	22	10	22	12	22	11	23	16	22	10	22	10	22	10	23	15	22	16

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
100	78	22	10	22	9	22	9	22	9	22	9	22	10	22	10	23	9	22	10
100	160	23	8	22	8	22	14	23	10	23	8	23	8	23	8	23	10	23	11
100	5	22	7	22	11	22	10	22	9	22	7	22	7	22	8	23	12	22	8
100	6	22	8	23	8	23	8	22	8	22	8	22	7	22	9	22	9	23	6
100	87	22	11	22	10	22	10	22	10	22	11	22	11	22	11	23	9	23	10
100	152	22	8	22	12	22	12	22	8	22	9	22	7	22	7	22	7	22	7
100	25	23	8	22	9	23	7	22	9	23	8	23	8	23	8	23	10	23	10
100	83	22	9	23	9	22	10	22	10	22	9	22	9	22	8	22	8	22	8
100	99	22	8	22	10	22	10	23	8	22	10	22	8	22	11	24	10	22	10
100	162	23	8	23	11	23	9	23	9	23	8	23	8	23	8	24	11	23	10
100	228	23	11	23	10	23	10	23	13	23	11	23	11	23	11	24	15	23	15
100	247	23	10	23	11	23	12	24	12	23	12	23	11	23	9	24	12	23	11
100	80	23	9	23	10	23	10	24	8	23	10	23	9	23	10	24	11	23	14
100	161	23	7	23	10	23	8	23	11	23	8	23	7	23	8	24	9	23	11
100	174	23	8	23	16	23	15	23	10	23	8	23	8	23	8	25	11	23	11
100	241	23	10	23	11	23	12	23	11	23	12	23	10	23	8	23	8	23	8
100	94	23	8	23	11	23	12	23	9	23	9	23	8	23	10	25	11	23	10
100	156	23	8	23	13	23	13	24	7	23	8	24	7	23	8	24	10	24	10
100	234	23	13	23	16	23	16	24	18	23	13	23	11	24	15	26	16	23	16
100	164	23	9	23	11	23	11	23	9	23	9	23	9	23	7	23	7	23	7
100	76	23	10	24	10	23	13	24	9	23	10	23	10	23	10	25	7	23	10
100	86	24	9	23	11	23	11	23	9	23	8	23	10	23	9	23	10	23	10
100	19	23	8	23	9	23	12	24	8	23	9	23	8	23	8	25	13	23	12
100	1	24	7	24	9	24	9	24	8	24	7	24	7	24	7	25	9	24	10
100	230	24	15	24	15	24	12	25	16	24	13	24	15	24	14	26	13	24	13
100	88	24	8	24	9	24	11	25	9	24	9	24	8	24	9	25	9	24	10
100	243	24	11	24	12	24	12	25	15	24	10	24	11	24	11	25	12	24	14
100	9	24	9	24	8	24	9	24	9	24	8	24	8	24	8	25	8	24	9
100	172	24	8	24	12	24	10	24	11	24	10	24	8	24	9	24	11	25	10
100	23	24	10	24	12	24	11	24	8	24	9	24	10	24	10	24	12	25	9
100	250	25	13	24	10	25	9	26	11	25	13	25	13	25	12	26	14	25	11
100	4	24	10	24	11	24	14	24	12	24	10	24	11	24	9	24	9	24	9
100	8	24	10	24	13	24	16	24	9	24	9	24	10	24	10	26	11	25	11
100	166	25	9	24	13	24	13	25	13	24	10	25	8	25	7	26	14	25	13
100	13	25	7	24	13	24	11	24	8	24	7	25	7	24	7	24	7	24	7
100	245	25	12	24	14	24	12	25	15	24	13	24	13	24	13	25	13	24	14
100	165	25	9	25	15	25	14	25	12	25	10	25	9	25	9	27	15	25	11
100	22	25	8	25	11	25	10	25	10	25	8	25	9	25	8	26	13	25	10
100	85	25	10	25	10	25	11	25	11	25	9	25	10	25	11	27	10	25	11
100	163	25	8	25	12	25	13	25	13	25	8	25	9	25	10	26	10	26	13
100	91	25	10	26	11	25	13	25	12	25	10	25	10	25	10	25	10	25	10
100	173	25	9	25	11	25	10	25	10	25	10	25	9	25	10	26	13	25	12
100	100	26	9	26	14	26	13	27	11	26	9	26	9	26	10	27	12	26	11
100	12	26	8	26	11	26	9	26	8	26	7	26	7	26	7	27	10	26	10
100	157	26	9	26	13	26	13	27	12	27	9	26	9	27	9	28	13	27	15
100	7	26	9	26	9	26	9	26	10	26	8	26	9	26	8	28	10	26	10
100	397	26	11	26	13	27	16	26	12	26	10	26	11	26	11	27	12	26	12
100	235	28	12	27	12	27	18	29	16	27	12	28	11	27	13	30	12	28	16
100	246	27	15	27	11	27	15	27	15	27	15	27	15	27	11	27	11	27	11
100	227	28	12	27	14	27	14	28	14	27	15	28	13	28	12	29	12	28	12
100	84	27	13	27	11	27	10	29	12	27	12	27	13	27	13	30	13	28	12
100	175	27	9	27	16	27	13	27	8	27	9	27	9	27	8	27	8	27	8

Número de tareas	Instancia	R1		R2		R11		R13		R14		R15		R16		R17		R18	
		Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es	Tr	Es
100	93	28	10	28	10	28	12	29	11	28	11	28	10	28	11	29	12	28	14
100	144	50	23	50	22	50	22	53	21	50	16	50	17	49	20	52	17	52	23
100	215	52	17	51	27	50	24	50	17	50	24	52	17	50	24	50	24	51	24
100	204	54	22	53	27	52	26	54	20	52	24	52	26	54	23	54	29	53	23
100	141	52	18	52	24	52	23	54	20	52	20	52	18	54	21	53	20	53	23
100	206	54	20	53	25	52	27	54	24	53	20	52	20	54	20	54	18	53	19
100	146	54	25	54	21	53	24	53	23	54	26	54	25	53	24	53	24	53	24
100	223	53	17	53	26	53	26	54	25	53	19	54	17	53	19	54	17	53	19
100	210	56	26	54	28	53	25	55	21	55	24	53	27	55	26	56	27	54	28
100	436	56	22	55	23	54	19	54	22	54	19	52	24	52	21	54	19	54	19
100	139	54	21	55	22	55	23	54	18	53	24	54	21	52	17	54	21	54	18
100	442	55	23	55	24	53	29	55	28	53	29	55	29	55	19	53	29	53	31
100	291	55	23	55	25	58	27	56	26	55	25	54	23	53	26	56	23	56	34
100	293	57	24	56	29	55	24	55	22	55	24	55	25	54	25	54	26	55	24
100	137	58	29	56	26	55	26	55	28	55	28	53	23	59	26	55	27	53	25
100	143	56	24	55	22	55	22	56	22	55	22	53	23	53	20	55	21	53	20
100	214	57	25	56	25	55	29	55	27	55	31	56	25	57	26	56	19	57	28
100	300	57	26	58	29	56	24	58	26	55	26	56	24	55	26	57	30	56	24
100	449	59	27	56	27	56	24	56	22	56	27	57	27	56	24	56	24	55	26
100	279	58	30	56	30	56	29	56	29	55	24	56	31	55	24	57	26	55	26
100	444	58	30	57	22	57	24	54	24	57	21	57	30	57	26	57	22	57	25
100	285	59	26	58	27	57	22	57	23	57	22	57	25	56	20	57	22	57	22
100	290	59	26	57	25	57	28	57	28	57	25	59	26	55	23	59	21	55	24
100	288	60	29	58	33	59	23	59	26	59	29	59	26	60	30	59	25	57	24
100	432	59	28	58	27	58	21	59	19	57	21	57	23	59	21	60	21	58	23
100	448	59	23	57	26	58	24	58	24	57	26	60	20	56	24	60	22	56	24
100	286	60	29	59	28	59	27	60	25	57	23	59	29	60	28	59	31	57	25
100	434	60	22	60	31	60	24	60	23	60	20	60	22	60	18	61	23	61	28
100	278	60	28	61	30	60	25	61	22	59	24	58	28	58	24	58	26	60	24
100	298	64	27	63	31	61	26	62	25	61	31	63	27	63	27	61	26	59	28
100	282	64	24	62	34	61	33	62	24	62	27	64	24	63	32	63	23	64	24
100	147	62	19	61	22	61	23	60	21	60	25	62	22	61	21	62	18	62	29
100	289	64	27	64	31	63	31	63	26	63	23	63	33	64	26	63	31	64	25

Tabla 40. Resultados de los experimentos computacionales con instancias de Otto et al. (2013) para R1, R2, R11 y R13-R18. Fuente: propia

Apéndice 4 – Resultados de la experiencia computacional de MULTI_HEUR_PART y OPWK_PART

Los resultados de los experimentos computacionales de MULTI_HEUR_PART y OPWK_PART, con las 250 instancias de Otto *et al.* (2013), se presentan en la Tabla 41.

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
50	12	6	5	6	5
50	153	8	3	8	3
50	231	7	4	7	4
50	244	7	6	7	4

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
50	167	8	2	8	2
50	170	8	2	8	2
50	173	8	2	8	2
50	232	8	3	8	3
50	236	8	3	8	3
50	239	8	3	7	4
50	249	8	4	8	4
50	90	8	3	8	3
50	92	7	3	7	3
50	97	7	5	7	5
50	79	8	5	8	5
50	168	9	3	9	3
50	202	10	3	9	4
50	204	11	3	11	3
50	288	11	5	11	5
50	298	11	5	11	5
50	75	12	3	12	3
50	139	12	5	12	4
50	150	11	5	11	5
50	206	12	4	12	4
50	221	11	4	11	4
50	223	11	5	11	5
50	286	12	6	12	6
50	289	12	5	12	5
50	216	12	4	12	4
50	278	13	5	12	6
50	283	13	5	12	6
50	287	13	6	13	6
50	66	12	4	12	4
50	128	13	6	12	6
50	132	13	6	13	6
50	140	13	6	12	6
50	143	12	7	12	7
50	276	13	6	12	6
50	53	13	4	13	4
50	60	13	4	12	4
50	67	13	5	13	4
50	69	12	5	12	5
50	146	13	7	13	7
50	208	13	5	13	4
50	135	14	5	13	5
50	141	13	7	13	7
50	144	13	7	13	6
50	147	13	7	13	7
50	210	13	5	13	5
50	217	14	4	14	4
50	277	13	7	13	7
50	62	13	5	13	5
50	71	13	5	13	5
50	127	14	7	14	7
50	134	14	7	14	7
50	295	16	10	16	10
50	119	26	12	25	12

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
50	179	27	9	27	9
50	185	27	11	26	17
50	186	26	12	26	13
50	189	26	10	26	11
50	269	28	11	27	11
50	180	27	8	27	8
50	40	26	10	26	9
50	196	28	11	28	13
50	265	27	11	27	11
50	272	28	12	26	15
50	112	27	10	27	9
50	114	28	10	27	12
50	120	27	15	27	13
50	176	28	8	27	15
50	26	27	11	27	11
50	264	27	13	27	12
50	275	28	15	28	15
50	107	28	14	28	11
50	113	28	13	28	11
50	183	29	10	29	9
50	193	30	10	30	11
50	195	29	10	29	10
50	198	28	11	28	9
50	270	29	13	28	13
50	47	28	13	28	11
50	29	29	10	29	12
50	109	30	10	30	9
50	123	32	11	32	11
50	184	38	13	38	13
50	42	24	10	24	10
50	105	25	9	24	9
50	188	25	7	25	7
50	44	25	13	25	7
50	45	25	14	26	11
50	49	25	9	25	7
50	33	25	8	25	9
50	200	25	16	26	9
50	41	26	11	26	11
50	43	26	9	25	11
50	32	25	12	25	11
50	187	26	13	26	9
50	261	28	13	28	12
50	104	28	11	28	11
50	273	27	18	27	11
50	117	27	13	27	10
50	50	27	9	27	8
50	110	28	9	26	12
50	182	27	9	27	11
50	115	28	10	28	10
50	30	27	11	26	16
50	251	28	13	27	14
50	258	29	13	28	12
50	177	28	8	28	8

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
50	178	28	11	28	11
50	192	28	11	28	13
50	48	28	9	28	9
50	106	28	11	28	10
50	191	29	11	28	11
50	31	28	7	28	7
50	267	29	12	28	13
50	253	28	17	28	13
50	255	30	15	29	15
50	194	28	11	29	10
50	197	28	14	28	9
50	28	28	12	28	12
50	260	29	12	29	12
50	122	29	15	29	12
50	124	29	14	29	12
100	257	13	4	13	4
100	114	13	9	13	9
100	181	13	7	13	7
100	190	13	10	13	10
100	187	14	4	14	4
100	192	14	5	14	5
100	266	14	5	13	10
100	27	14	5	14	5
100	273	14	5	14	5
100	112	14	5	14	5
100	198	14	5	14	5
100	271	14	5	14	5
100	252	14	6	14	6
100	29	14	9	14	9
100	40	14	7	14	7
100	186	14	8	14	8
100	258	15	5	15	5
100	28	15	4	15	4
100	43	15	5	15	5
100	108	15	5	15	5
100	26	15	4	15	4
100	36	15	4	15	4
100	259	15	8	15	8
100	101	15	9	15	9
100	117	16	6	16	6
100	124	16	6	16	6
100	116	16	6	16	6
100	248	20	8	20	8
100	18	20	5	20	5
100	77	20	8	20	8
100	90	21	8	21	8
100	14	21	6	21	6
100	81	21	7	21	7
100	169	21	8	21	8
100	21	21	7	21	7
100	151	22	6	22	6
100	17	22	6	22	6
100	244	22	9	22	9

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
100	168	22	6	22	6
100	249	22	8	22	8
100	78	22	8	22	8
100	160	22	8	22	8
100	5	22	6	22	6
100	6	22	7	22	7
100	87	22	10	22	10
100	152	22	7	22	7
100	25	22	9	22	9
100	83	22	8	22	8
100	99	22	7	22	7
100	162	23	6	23	6
100	228	23	9	23	9
100	247	23	8	23	8
100	80	23	7	23	7
100	161	23	6	23	6
100	174	23	7	23	7
100	241	23	8	23	8
100	94	23	8	23	8
100	156	23	7	23	7
100	234	23	11	23	13
100	164	23	7	23	7
100	76	23	8	23	8
100	86	23	10	23	10
100	19	23	7	23	7
100	1	24	6	24	6
100	230	24	12	24	12
100	88	24	7	24	7
100	243	24	8	24	8
100	9	24	6	24	6
100	172	24	8	24	8
100	23	24	8	24	8
100	250	24	10	24	10
100	4	24	9	24	9
100	8	24	8	24	8
100	166	24	13	24	13
100	13	24	7	24	7
100	245	24	9	24	9
100	165	25	7	25	7
100	22	25	7	25	7
100	85	25	9	25	9
100	163	25	8	25	8
100	91	25	9	25	9
100	173	25	8	25	8
100	100	26	8	26	8
100	12	26	7	26	7
100	157	26	8	26	8
100	7	26	8	26	8
100	397	26	11	26	11
100	235	27	10	27	10
100	246	27	11	27	11
100	227	27	11	27	11
100	84	27	10	27	10

Número de tareas	Instancia	MULTI_HEUR_PART		OPWK_PART	
		Trabajadores	Estaciones	Trabajadores	Estaciones
100	175	27	8	27	8
100	93	28	9	28	9
100	144	49	20	50	17
100	215	50	24	50	24
100	204	52	18	52	18
100	141	51	19	51	41
100	206	52	20	52	20
100	146	53	21	53	22
100	223	53	17	53	16
100	210	53	25	53	25
100	436	52	21	53	20
100	139	52	21	53	35
100	442	53	28	53	28
100	291	55	20	55	20
100	293	54	25	54	25
100	137	54	24	54	28
100	143	54	17	54	19
100	214	55	27	55	27
100	300	55	26	55	26
100	449	55	26	55	26
100	279	55	24	55	24
100	444	54	24	54	24
100	285	56	20	56	20
100	290	55	23	55	25
100	288	57	24	57	24
100	432	57	21	57	22
100	448	56	24	56	27
100	286	57	25	57	25
100	434	58	20	58	20
100	278	58	24	58	24
100	298	61	22	61	22
100	282	61	26	61	26
100	147	60	23	60	21
100	289	63	23	63	23

Tabla 41. Resultados de los experimentos computacionales con instancias de Otto *et al.* (2013) para MULTI_HEUR_PART y OPWK_PART. Fuente: propia

Apéndice 5 – Comparación del procedimiento GRDY_DTT1 con OPWK_PART y Max_S

GRDY_DTT1 resuelve el problema SMALBP-DTT1 con la minimización del número de estaciones como primer objetivo y el número de trabajadores como segundo. Por un lado, en el apéndice 5.1, GRDY_DTT1 se compara con OPWK_PART, el procedimiento que obtiene los mejores resultados para el SMALBP-DTT1 entre todos los que han sido propuestos en la sección 5. La comparación se realiza teniendo en cuenta que los objetivos son diferentes (recuérdese que OPWK_PART minimiza el número de trabajadores como primer objetivo y el número de estaciones como segundo). Por otro lado, en el apéndice

5.2, GRDY_DTT1 se compara con Max_S, heurística propuesta en Sepahi y Jalali Naini (2014) que resuelve el SMALBP-DTT1 con el objetivo de minimizar el número de estaciones.

En la presente experiencia computacional se utilizan los mismos 250 ejemplares de Otto *et al.* (2013), con las mismas características que los utilizados en las experiencias computacionales anteriores (para más detalle, ver la sección 5.4.1).

Apéndice 5.1 – Comparación de GRDY_DTT1 con OPWK_PART

Los resultados de la comparación entre OPWK_PART y GRDY_DTT1 se presentan en la Tabla 42 y la Tabla 43. La estructura de dichas tablas es igual a la de la Tabla 27.

La Tabla 42 y la Tabla 43 muestran que, en casi la totalidad de las instancias, OPWK_PART obtiene un número de trabajadores inferior respecto a GRDY_DTT1: 93.6% para instancias de 50 tareas y 91.2% para instancias de 100 tareas. En el resto de instancias, el número de trabajadores de OPWK_PART y GRDY_DTT1 coinciden.

En particular, en un elevado porcentaje de instancias (68.8% para instancias de 50 tareas y 72.0% para instancias de 100 tareas), OPWK_PART obtiene un menor número de trabajadores (una media de 4.40 trabajadores menos para instancias de 50 tareas y 4.63 trabajadores menos para instancias de 100 tareas) y GRDY_DTT1 obtiene un menor número de estaciones (una media de 2.24 estaciones menos para instancias de 50 tareas y 3.91 estaciones menos para instancias de 100 tareas).

OPWK_PART obtiene un menor número de trabajadores con las mismas estaciones en el 23.2% de instancias de 50 tareas (con una media de 3.34 trabajadores menos) y en el 17.6% de instancias de 100 tareas (con una media de 2.41 trabajadores menos). Incluso, OPWK_PART presenta un menor número tanto de estaciones como de trabajadores en el 1.6% de instancias de 50 tareas (con una media de 1.00 trabajadores menos y 1.00 estaciones menos) y en el 1.6% de instancias de 100 tareas (con una media de 1.50 trabajadores menos y 1.00 estaciones menos).

En el 0.8% de instancias de 50 tareas y el 2.4% de instancias de 100 tareas, GRDY_DTT1 y OPWK_PART obtienen el mismo número de trabajadores, pero GRDY_DTT1 consigue una solución con un número menor de estaciones. En el 5.6% de instancias de 50 tareas y el 6.4% de instancias de 100 tareas ambos procedimientos obtienen las mismas soluciones.

Resultados de OPWK_PART comparados con GRDY_DTT1 (50 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	0%	0%	1.6%

Resultados de OPWK_PART comparados con GRDY_DTT1 (50 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
			(2/125) 1.00 trabajadores menos 1.00 estaciones menos
Igual número de estaciones	0%	5.6% (7/125)	23.2% (29/125) 3.34 trabajadores menos
Número de estaciones superior	0%	0.8% (1/125) 1.00 estaciones más	68.8% (86/125) 4.40 trabajadores menos 2.24 estaciones más

Tabla 42. Tabla comparativa para instancias de 50 tareas entre OPWK_PART y GRDY_DTT1. Fuente: propia

Resultados de OPWK_PART comparados con GRDY_DTT1 (100 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	0%	0%	1.6% (2/125) 1.50 trabajadores menos 1.00 estaciones menos
Igual número de estaciones	0%	6.4% (8/125)	17.6% (22/125) 2.41 trabajadores menos
Número de estaciones superior	0%	2.4% (3/125) 1.00 estaciones más	72.0% (90/125) 4.63 trabajadores menos 3.91 estaciones más

Tabla 43. Tabla comparativa para instancias de 100 tareas entre OPWK_PART y GRDY_DTT1. Fuente: propia

Apéndice 5.2 – Comparación de GRDY_DTT1 con Max_S

A continuación, GRDY_DTT1 y Max_S se comparan teniendo en consideración que ambos procedimientos tienen como objetivo principal minimizar el número de estaciones (GRDY_DTT1 considera como objetivo secundario la minimización del número de trabajadores). Los resultados de la comparación entre GRDY_DTT1 y Max_S se presentan en la Tabla 44 y la Tabla 45. La estructura de la Tabla 44 y la Tabla 45 es igual a la de la Tabla 27.

La Tabla 44 y la Tabla 45 muestran que en el 48.8% de instancias de 50 tareas y el 39.2% de instancias de 100 tareas GRDY_DTT1 obtiene un número de estaciones inferior respecto a Max_S. En particular, en el 30.4% de instancias de 50 tareas y el 28.8% de instancias de 100 tareas, GRDY_DTT1 obtiene no solo un número de estaciones inferior sino también un número de trabajadores inferior.

Sólo en el 4.8% de instancias de 50 tareas y el 12.0% de instancias de 100 tareas, Max_S alcanza un número de estaciones inferior. En el resto de instancias (46.4% para instancias

de 50 tareas y 48.8% para instancias de 100 tareas), el número de estaciones obtenido por GRDY_DTT1 y Max_S coinciden.

Resultados de GRDY_DTT1 comparados con Max_S (50 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	8.0% (10/125) 1.70 trabajadores más 1.10 estaciones menos	10.4% (13/125) 1.15 estaciones menos	30.4% (38/125) 1.71 trabajadores menos 1.18 estaciones menos
Igual número de estaciones	23.2% (29/125) 1.79 trabajadores menos	15.2% (19/125)	8.0% (10/125) 1.40 trabajadores menos
Número de estaciones superior	4.8% (6/125) 1.83 trabajadores más 1.00 estaciones más	0%	0%

Tabla 44. Tabla comparativa para instancias de 50 tareas entre GRDY_DTT1 y Max_S. Fuente: propia

Resultados de GRDY_DTT1 comparados con Max_S (100 tareas)			
	Número de trabajadores superior	Igual número de trabajadores	Número de trabajadores inferior
Número de estaciones inferior	4.8% (6/125) 2.67 trabajadores más 1.17 estaciones menos	5.6% (7/125) 1.43 estaciones menos	28.8% (36/125) 2.39 trabajadores menos 1.56 estaciones menos
Igual número de estaciones	22.4% (28/125) 1.54 trabajadores menos	19.2% (24/125)	7.2% (9/125) 1.56 trabajadores menos
Número de estaciones superior	11.2% (14/125) 2.07 trabajadores más 1.00 estaciones más	0.8% (1/125) 1.00 estaciones más	0%

Tabla 45. Tabla comparativa para instancias de 100 tareas entre GRDY_DTT1 y Max_S. Fuente: propia

Apéndice 6 – Resultados de la experiencia computacional de MILP-DTT2, HEUR_CT_PART, GRDY_DTT2 y OPCT_PART

Los resultados de los experimentos computacionales de MILP-DTT2, HEUR_CT_PART, GRDY_DTT2 y de OPCT_PART con las 250 instancias de Otto *et al.* (2013) se presentan en la Tabla 42 (siendo "CT" el tiempo de ciclo y "Trab" el número de trabajadores).

Número de tareas	Instancia	UB ^M	LB _{CT}	MILP-DTT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
50	12	5	298	323	20	322	20	341	20	327	20
50	153	6	318	335	24	338	24	338	24	335	24
50	231	7	313	408	20	408	22	408	22	408	22
50	244	6	307	489	16	493	19	493	19	489	16

Número de tareas	Instancia	UB^M	LB_{CT}	MILP-DTT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
50	167	6	355	367	22	368	23	358	24	357	23
50	170	6	341	355	22	355	23	344	24	343	23
50	173	6	400	400	20	400	19	402	20	400	19
50	232	6	384	409	19	409	19	411	21	409	19
50	236	6	340	519	15	525	18	521	19	519	18
50	239	6	351	466	18	512	18	512	18	466	18
50	249	6	346	422	22	422	22	422	21	422	21
50	90	6	338	414	19	414	21	414	23	414	19
50	92	7	299	327	24	341	25	331	26	327	24
50	97	7	398	400	19	400	19	401	23	400	19
50	79	7	314	418	21	418	26	418	26	418	26
50	168	7	426	433	21	426	20	427	23	426	20
50	202	8	526	526	19	526	18	526	21	526	18
50	204	8	588	588	20	588	18	588	19	588	19
50	288	9	528	528	21	545	21	531	22	531	22
50	298	11	523	523	22	523	25	523	26	523	22
50	75	10	596	596	21	596	19	596	22	596	21
50	139	10	564	564	21	564	21	564	26	564	21
50	150	11	595	595	19	595	19	595	23	595	19
50	206	10	569	569	22	569	21	569	23	569	21
50	221	11	565	565	21	565	22	565	22	565	20
50	223	11	607	607	19	607	19	607	22	607	19
50	286	10	543	543	22	543	22	543	24	543	22
50	289	9	531	531	22	531	22	533	25	531	22
50	216	12	619	619	21	619	20	619	22	619	20
50	278	10	605	605	22	605	21	605	24	605	21
50	283	9	566	566	25	572	23	567	26	566	22
50	287	10	545	546	24	546	28	546	28	546	24
50	66	12	552	552	23	552	24	552	26	552	23
50	128	11	537	537	24	537	24	537	27	537	25
50	132	9	562	562	24	562	24	562	26	562	24
50	140	10	607	607	21	607	23	607	25	607	21
50	143	12	574	574	23	574	23	574	27	574	23
50	276	10	560	560	24	560	23	560	26	560	25
50	53	10	560	560	25	560	22	560	25	560	22
50	60	9	608	608	22	608	21	608	23	608	22
50	67	10	590	590	22	590	22	590	25	590	22
50	69	12	604	604	21	604	24	604	25	604	21
50	146	12	570	570	24	570	28	570	29	570	24
50	208	10	564	564	25	564	27	564	27	564	24
50	135	10	595	595	24	595	24	595	29	595	24
50	141	13	632	632	21	632	21	632	25	632	21
50	144	12	577	577	23	577	25	577	27	577	23
50	147	13	585	585	24	585	24	585	28	585	24
50	210	12	617	617	23	617	21	617	24	617	21
50	217	11	623	623	23	623	24	623	25	623	23
50	277	12	582	582	23	582	23	582	27	582	23
50	62	12	612	612	23	612	23	612	25	612	22
50	71	12	564	860	18	564	26	564	27	564	24
50	127	12	586	586	25	586	25	586	30	586	25
50	134	14	642	642	23	642	24	642	25	642	25
50	295	16	613	613	29	613	28	613	31	613	31

Número de tareas	Instancia	UB ^M	LB _{CT}	MILP-DTT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
50	119	17	758	758	42	758	43	758	43	758	42
50	179	21	700	700	44	700	46	700	46	700	44
50	185	20	722	1028	34	738	43	722	46	722	46
50	186	20	742	742	44	742	44	742	44	742	44
50	189	17	823	823	39	836	40	823	39	823	39
50	269	16	787	787	39	787	39	787	40	787	40
50	180	18	712	712	47	712	48	712	48	712	46
50	40	20	760	1284	24	774	46	760	43	760	42
50	196	18	766	766	42	766	43	766	43	766	41
50	265	17	842	842	38	842	38	842	39	842	38
50	272	19	709	709	46	709	47	709	47	709	46
50	112	18	788	788	37	788	40	788	40	788	40
50	114	21	891	891	38	891	37	891	37	891	35
50	120	20	807	807	42	807	41	807	45	807	45
50	176	18	729	729	46	729	46	729	45	729	45
50	26	18	720	720	42	720	43	720	43	720	42
50	264	22	799	799	42	799	42	799	42	799	42
50	275	23	774	774	46	774	46	774	46	774	45
50	107	20	815	815	42	815	44	815	44	815	42
50	113	20	703	703	48	703	50	703	50	703	48
50	183	19	706	706	46	710	48	706	46	706	46
50	193	18	791	791	44	791	44	791	44	791	44
50	195	20	803	803	43	803	44	803	44	803	43
50	198	18	738	738	45	738	47	738	45	738	45
50	270	19	825	825	39	825	39	825	41	825	41
50	47	23	756	756	46	773	44	756	44	756	44
50	29	17	740	740	42	740	42	740	42	740	42
50	109	17	734	734	44	734	45	734	45	734	44
50	123	17	739	739	45	739	45	739	46	739	45
50	184	21	801	801	46	801	48	801	46	801	46
50	42	18	710	710	42	710	44	710	44	710	42
50	105	15	869	869	30	869	31	869	32	869	29
50	188	16	693	693	48	706	49	694	46	693	45
50	44	15	758	758	43	758	43	758	40	758	40
50	45	19	736	736	43	750	42	736	43	736	42
50	49	22	674	674	44	674	44	674	44	674	44
50	33	18	728	728	45	742	46	728	44	728	44
50	200	19	685	685	46	685	47	685	47	685	47
50	41	16	870	870	35	870	38	870	38	870	35
50	43	21	684	1430	24	690	44	684	47	684	47
50	32	20	721	892	39	738	47	721	43	721	43
50	187	19	695	695	45	695	45	695	45	695	43
50	261	20	822	822	41	822	42	822	43	822	40
50	104	21	745	745	42	745	43	745	44	745	43
50	273	18	852	852	39	852	42	852	42	852	38
50	117	17	718	718	44	718	44	718	45	718	43
50	50	19	720	720	45	720	45	720	45	720	45
50	110	15	858	858	34	858	35	858	35	858	34
50	182	21	863	863	37	863	40	863	40	863	36
50	115	17	739	739	44	739	44	739	46	739	44
50	30	19	783	783	44	792	44	783	44	783	43
50	251	21	701	701	46	701	46	701	46	701	46

Número de tareas	Instancia	UB ^M	LB _{CT}	MILP-DTT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
50	258	18	769	769	43	769	44	769	45	769	43
50	177	17	734	734	46	734	48	734	46	734	46
50	178	16	777	779	43	781	46	777	41	777	41
50	192	16	774	774	44	788	45	774	43	774	43
50	48	19	754	754	45	754	45	754	45	754	45
50	106	21	734	734	45	759	49	734	45	734	45
50	191	19	753	753	45	753	46	753	45	753	45
50	31	15	778	778	44	778	44	778	44	778	44
50	267	17	758	758	44	758	45	758	45	758	45
50	253	22	730	730	47	730	47	730	47	730	46
50	255	20	830	830	40	830	43	830	43	830	40
50	194	18	799	799	44	799	46	799	46	799	43
50	197	19	797	797	43	800	45	797	43	797	43
50	28	18	812	812	43	812	43	812	43	812	43
50	260	21	816	816	41	816	42	816	42	816	41
50	122	21	812	812	44	820	48	812	45	812	43
50	124	22	773	773	45	773	48	773	45	773	45
100	257	12	320	811	24	356	44	356	42	356	42
100	114	13	338	1671	14	340	47	340	47	340	47
100	181	13	342	-	-	345	42	345	42	345	42
100	190	13	414	-	-	437	34	414	35	414	35
100	187	12	531	-	-	531	27	531	27	531	27
100	192	12	393	-	-	402	41	393	39	393	38
100	266	13	345	1919	18	428	41	392	40	392	40
100	27	12	400	1783	17	400	39	400	39	400	39
100	273	13	364	-	-	365	47	365	45	365	45
100	112	12	383	2680	15	384	43	384	43	384	43
100	198	12	344	1759	23	345	42	345	43	345	42
100	271	12	320	960	22	383	46	372	44	372	44
100	252	14	344	1612	22	345	51	345	51	345	49
100	29	14	393	-	-	411	43	393	39	393	39
100	40	13	299	-	-	303	52	303	52	303	52
100	186	14	364	-	-	386	44	364	43	364	43
100	258	13	343	1722	18	349	48	349	48	349	47
100	28	13	340	1213	19	341	46	341	46	341	46
100	43	13	329	-	-	330	48	330	48	330	46
100	108	13	364	1092	20	366	48	366	48	366	47
100	26	13	287	-	-	303	52	303	52	303	50
100	36	13	332	-	-	334	50	334	50	334	47
100	259	15	337	933	25	353	52	339	52	339	50
100	101	15	352	1386	22	370	55	352	53	352	53
100	117	14	375	-	-	375	48	375	48	375	48
100	124	14	352	1103	20	353	50	353	50	353	48
100	116	16	421	1855	18	421	40	421	48	421	48
100	248	18	530	-	-	530	42	530	42	530	41
100	18	17	557	-	-	557	39	557	37	557	37
100	77	20	578	1235	21	578	36	578	43	578	43
100	90	19	588	3612	26	588	45	588	43	588	43
100	14	18	583	-	-	583	39	583	38	583	38
100	81	18	589	1174	24	589	40	589	40	589	38
100	169	21	601	-	-	614	40	601	39	601	38
100	21	21	570	-	-	570	41	570	43	570	43

Número de tareas	Instancia	UB ^M	LB _{CT}	MILP-DTT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
100	151	19	608	-	-	608	38	608	39	608	37
100	17	20	620	-	-	620	38	620	38	620	38
100	244	18	577	1357	28	590	41	577	45	577	42
100	168	19	572	-	-	584	43	572	41	572	41
100	249	19	556	1631	27	563	49	556	47	556	47
100	78	19	539	-	-	539	48	539	47	539	47
100	160	22	589	-	-	589	41	589	41	589	41
100	5	21	565	2163	24	565	42	565	42	565	42
100	6	22	587	-	-	587	43	587	41	587	41
100	87	19	535	1698	29	549	48	535	50	535	50
100	152	22	625	-	-	645	39	625	40	625	40
100	25	20	607	-	-	607	40	607	41	607	40
100	83	20	643	-	-	643	35	643	42	643	37
100	99	20	553	-	-	553	45	553	45	553	43
100	162	20	635	-	-	635	40	635	40	635	37
100	228	20	552	-	-	552	48	552	48	552	47
100	247	20	596	-	-	596	45	596	45	596	45
100	80	21	606	-	-	606	41	606	41	606	38
100	161	21	569	-	-	569	43	569	43	569	40
100	174	21	590	-	-	590	44	590	44	590	42
100	241	21	604	-	-	604	38	604	43	604	39
100	94	20	579	2185	23	579	52	579	52	579	48
100	156	23	619	-	-	619	40	619	40	619	37
100	234	21	577	-	-	590	47	577	51	577	46
100	164	22	582	-	-	582	41	582	43	582	41
100	76	21	565	2182	28	565	46	565	46	565	44
100	86	23	614	-	-	614	44	614	44	614	39
100	19	22	548	-	-	548	46	548	46	548	44
100	1	21	590	-	-	590	47	590	47	590	46
100	230	22	563	-	-	578	48	563	52	563	52
100	88	21	590	-	-	590	48	590	48	590	48
100	243	22	575	-	-	588	50	575	48	575	48
100	9	21	616	-	-	630	44	616	43	616	43
100	172	24	583	-	-	583	44	583	44	583	40
100	23	22	605	-	-	605	42	605	42	605	41
100	250	22	627	-	-	627	44	627	44	627	42
100	4	23	626	-	-	635	41	626	42	626	42
100	8	22	635	-	-	635	41	635	41	635	40
100	166	24	571	-	-	571	48	571	48	571	48
100	13	24	638	-	-	651	43	638	45	638	45
100	245	23	558	-	-	558	53	558	53	558	51
100	165	23	591	-	-	591	48	591	48	591	47
100	22	21	571	-	-	593	46	571	47	571	47
100	85	24	586	-	-	586	53	586	53	586	50
100	163	24	614	-	-	614	44	614	44	614	42
100	91	25	569	-	-	569	46	569	58	569	57
100	173	22	589	-	-	589	45	589	45	589	41
100	100	24	571	2896	26	571	51	571	51	571	49
100	12	22	588	-	-	609	48	588	49	588	49
100	157	25	558	-	-	558	51	558	51	558	50
100	7	24	642	-	-	671	43	642	44	642	44
100	397	23	582	2741	26	603	55	582	51	582	51

Número de tareas	Instancia	UB ^M	LB _{CT}	MILP-DDT2		HEUR_CT_PART		GRDY_DTT2		OPCT_PART	
				CT	Trab	CT	Trab	CT	Trab	CT	Trab
100	235	24	598	-	-	598	50	598	50	598	46
100	246	24	568	-	-	580	52	568	56	568	54
100	227	24	614	-	-	614	50	614	52	614	50
100	84	25	585	-	-	585	57	585	57	585	52
100	175	24	593	-	-	611	52	593	49	593	49
100	93	24	589	-	-	589	53	589	53	589	51
100	144	32	725	-	-	760	87	725	84	725	84
100	215	39	750	-	-	750	81	750	81	750	76
100	204	37	795	-	-	809	84	795	79	795	79
100	141	41	773	-	-	801	89	773	85	773	85
100	206	43	822	-	-	822	81	822	81	822	78
100	146	40	773	-	-	791	82	773	84	773	84
100	223	40	736	-	-	736	90	736	90	736	87
100	210	40	789	-	-	789	82	789	82	789	78
100	436	39	722	-	-	751	90	722	90	722	86
100	139	37	728	-	-	760	86	728	87	728	87
100	442	40	860	-	-	924	82	860	76	860	76
100	291	34	726	-	-	763	87	726	87	726	87
100	293	36	789	-	-	800	83	789	88	789	88
100	137	40	774	-	-	806	92	774	87	774	87
100	143	38	727	-	-	760	90	728	91	727	91
100	214	38	792	-	-	-	-	792	88	792	88
100	300	36	772	-	-	823	86	772	88	772	88
100	449	39	780	-	-	-	-	780	88	780	88
100	279	44	901	-	-	-	-	901	75	901	75
100	444	34	838	-	-	876	82	840	78	838	78
100	285	39	784	-	-	849	90	784	86	784	86
100	290	37	872	-	-	-	-	872	81	872	81
100	288	41	822	-	-	-	-	822	87	822	87
100	432	36	772	-	-	816	98	774	91	772	91
100	448	40	810	-	-	-	-	810	86	810	86
100	286	41	813	-	-	-	-	813	87	813	87
100	434	35	813	-	-	-	-	813	86	813	86
100	278	36	863	-	-	961	88	863	81	863	81
100	298	42	845	-	-	897	86	845	83	845	83
100	282	42	869	-	-	-	-	869	83	869	83
100	147	37	784	-	-	-	-	784	88	784	88
100	289	40	795	-	-	-	-	795	85	795	85

Tabla 46. Resultados de los experimentos computacionales con instancias de Otto *et al.* (2013) para MILP-DDT2, HEUR_CT_PART, GRDY_DTT2 y OPCT_PART. Fuente: propia