

UNIVERSIDAD DE OVIEDO

DEPARTAMENTO DE ADMINISTRACIÓN DE EMPRESAS

Programa de doctorado: Economía y empresa

**ANÁLISIS Y MEJORA MEDIANTE TÉCNICAS METAHEURÍSTICAS
DE SUBASTAS MULTI-PRODUCTO ONLINE PARA EL SECTOR
SIDERÚRGICO**

TESIS DOCTORAL

AUTOR: **D. Borja Ena Rodríguez**

DIRECTOR: **Dr. D. Alberto Gómez Gómez**

GIJÓN, 2018



RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Análisis y mejora mediante técnicas metaheurísticas de subastas multi-producto online para el sector siderúrgico	Inglés: Analysis and improvements through metaheuristic techniques of online multi-product auctions within the steel industry.
2.- Autor	
Nombre: Borja Ena Rodríguez	DNI/Pasaporte/NIE:
Programa de Doctorado: ECONOMÍA Y EMPRESA	
Órgano responsable: Universidad de Oviedo	

RESUMEN (en español)

Objetivo, interés y metodología

El objetivo principal de la tesis doctoral "Análisis y mejora mediante técnicas meta-heurísticas de subastas multi-producto online para el sector siderúrgico", realizada por D. Borja Ena Rodríguez y dirigida por el Dr. D. Alberto Gómez Gómez, es mejorar el proceso de ventas de productos de acero "non-prime", en primer lugar resolviendo el problema de agrupación de productos en lotes de manera que éstos sean lo más homogéneos posibles y bajo restricciones tanto de los productos como del tamaño de lote, y en segundo lugar analizando todo el proceso global de venta para estudiar la idoneidad del mismo y proponer otras alternativas.

Durante el proceso de fabricación de productos de acero, no todos los productos cumplen el nivel de calidad exigido por el cliente final. Estos productos, denominados "non-prime" se venden mediante una subasta online semanal, siendo necesario una agrupación previa de los mismos en lotes, de manera que éstos sean lo más homogéneos posibles atendiendo a varias características de los productos. Esta agrupación es un problema muy complejo de resolver, y es precisamente la parte fundamental del proceso de ventas, ya que cuanto más homogéneos sean los lotes, más atractivos serán para los clientes, que a su vez realizarán más pujas, incrementando el beneficio. Aproximadamente medio millón de toneladas se venden anualmente con este sistema en todas las plantas de productos planos del grupo. Resolver de una manera adecuada el problema representa, pues, un tremendo



impacto económico, a la vez que un reto desde el punto de vista algorítmico, ya que no se ha resuelto ningún problema similar hasta la fecha.

Inicialmente se realiza una descripción del problema, exponiendo primero todo el proceso de ventas de productos non-prime, centrándose en la parte de subastas y las reglas de agrupamiento que deben de cumplirse durante la preparación de los lotes, para finalizar describiendo los requisitos que debe de cumplir el algoritmo a desarrollar.

A continuación se analiza la literatura existente de subastas, tanto las principales variantes existentes como los principales factores que influyen en el proceso, para estudiar si el método de subasta elegido por la empresa (cerrada de primer precio, con precio reserva) es el más adecuado, o se pueden proponer alternativas.

Tras esto, para poder plantear un algoritmo de resolución del problema se realiza primero un estado del arte de la literatura existente del problema de empaquetado por lotes (Bin Packing Problem), analizando sus variantes y las técnicas utilizadas para resolverlo. Dado que el problema a resolver no coincide con ninguna de las variantes, se define este nuevo problema como “Problema de empaquetado de lotes homogéneos con mínimo peso”, se formula matemáticamente y se propone una metodología de resolución estructurada en 3 fases, consistentes en una división inicial de los productos basándose en sus características comunes, un algoritmo de clustering y una última fase de optimización de la asignación de productos a lotes finales, para la cual se desarrollan 3 heurísticos (FIFO, Greedy y Distancias) y un meta-heurístico GRASP.

Para comparar la bondad de la solución y dado que no existe un problema similar, se generan 80 instancias de prueba basadas tanto en las instancias clásicas de Bin Packing Problem, como en datos reales del problema a resolver, y se comparan los resultados obtenidos con cada heurístico y el meta-heurístico GRASP. Finalmente, se prueba el algoritmo con un caso de real para demostrar tanto su funcionamiento como su rendimiento.



Resultados

Del análisis teórico realizado sobre subastas se concluye que es coherente que la subasta sea cerrada y con precio reserva. Se propone como recomendación realizar una subasta descendente para aquellos productos que no se han vendido con la subasta inicial, para aumentar el beneficio global del sistema de venta.

En lo referente al problema de agrupamiento, tras analizar las pruebas realizadas sobre las 80 instancias generadas, se puede concluir que el algoritmo propuesto resuelve el problema planteado, siendo el meta-heurístico GRASP la mejor solución para la fase final, ya que mejora las soluciones obtenidas utilizando solamente los heurísticos individuales en el 84% de las instancias. Las mejores soluciones se han conseguido utilizando el algoritmo FIFO como generador de las soluciones iniciales del GRASP. Para su aplicación al caso real se propone lanzar la tercera fase del algoritmo 2 veces, una con soluciones iniciales generadas por el FIFO y otra con soluciones iniciales generadas con el heurístico de Distancias. De esta forma, se consigue un tiempo de computación para 26 plantas y 2771 productos de unos 4 minutos y se garantiza una muy buena solución.



RESUMEN (en Inglés)

Objective, interest and methodology

The aim of the doctoral thesis “Analysis and improvements through metaheuristics techniques of online multi-product auctions within the steel industry”, by D. Borja Ena Rodríguez and directed by Dr. D. Alberto Gómez Gómez is to improve the selling process of non-prime steel products, firstly by resolving the problem of grouping steel products into bundles, making these bundles as homogeneous as possible, under both product features and bundle size constraints, and secondly by analyzing all the selling process to study its performance and propose possible alternatives.

During the Steel making process, not all the products achieve the quality required by the customers. These products, denoted “non-prime” products, are sold in an online auction that takes place every week. Before putting the products on auction, they have to be grouped in bundles, trying to make these bundles as homogeneous as possible. This bundle generation process is a very complex problem to solve, and its precisely the main part of the selling process, since the more homogeneous the bundles are, the more attractive to the customer and more bids will take place, resulting in the end in more profits for the company. Around half a million steel tons are currently sold with this system in all flat Europe plants in the group. Hence, being able to solve properly the problem represents a tremendous economic impact, apart from a challenge from the algorithmic point of view, since no similar problem has been solved so far.

First, a description of the problem is done, explaining all the non-prime products selling process, focusing on the auction phase and the grouping rules that must be considered during the bundling preparation, and describing the requirements of the algorithm to be developed.

Afterwards, a survey of existing auctions state of the art of is done, analyzing



both the main variants and the main factors that influence the process, to be able to check if the current auction method used by the company (first-price sealed bid with hidden price) is the most adequate one, or if some alternatives can be proposed.

Next, to be able to propose an algorithm to solve the problem, an analysis is done of the state of the art of Bin Packing Problem to study all its different variants and the techniques used to solve it. Since the problem proposed in this thesis hasn't been analyzed before, the problem is defined as "Homogeneous Bin Packing Problem with Minimum Weight". This problem is formulated mathematically, and a methodology to solve it based in 3 phases is defined: initial division of products based on its common features, a clustering algorithm and finally an optimization phase of assigning products to final bundles. For this last phase 3 heuristics (FIFO, Greedy and Distances), as well as a GRASP meta-heuristics are developed.

To be able to ensure the quality of the solution, and since no similar problem exists so far, 80 tests instances are generated, based on the current literature of Bin Packing Problem, as well as in actual data of the problem to be solved. Each heuristic and the meta-heuristic is tested against the test instances to compare the results. Finally, the algorithm is tested with an actual use case to prove both its applicability and its performance.

Results

From the theoretical analysis done about auctions it is possible to conclude that is coherent to use a sealed-bid with reserve price auction. A proposal is done to incorporate a descending-bid auction for those products not sold in the initial auction to be able to improve the selling process benefit.

From the tests done with the 80 tests instances, it can be concluded that the developed algorithm solves the proposed problem, being the GRASP algorithm the best solution to use in the third phase. GRASP improve the solutions obtained with the individual heuristics in 84% of the test instances. The best solutions were obtained using FIFO heuristic as initial solution generator for the GRASP. For the use of the



algorithm in the real case, it is proposed to launch this third phase of the algorithm with the GRASP twice: first using the FIFO as initial solution generator, and then using the Distances heuristic for the same purpose. This way, the computation time for 26 plants and 2771 products is around 4 minutes, ensuring a very good solution.

Agradecimientos

Quiero dar las gracias en primer lugar a mi director de tesis, el Dr. D. Alberto Gómez Gómez, por su ayuda y dedicación durante el desarrollo de la tesis, así como por sus consejos y directrices tanto para la finalización de la misma como durante todos estos 10 años de trabajo juntos.

También quiero agradecer a toda la gente que me ha apoyado durante la realización de este trabajo. En especial, a mi mujer y a mi hija, por su comprensión y su ayuda para poder encontrar el tiempo necesario; y a mis padres, cuya ilusión ha sido una motivación para el desarrollo de esta tesis.

Para Noelia y Paola

Para mis padres

RESUMEN

Objetivo

El objetivo principal de la tesis doctoral “Análisis y mejora mediante técnicas meta-heurísticas de subastas multi-producto online para el sector siderúrgico”, realizada por D. Borja Ena Rodríguez y dirigida por el Dr. D. Alberto Gómez Gómez, es mejorar el proceso de ventas de productos de acero “non-prime”, en primer lugar resolviendo el problema de agrupación de productos en lotes de manera que éstos sean lo más homogéneos posibles y bajo restricciones tanto de los productos como del tamaño de lote, y en segundo lugar analizando todo el proceso global de venta para estudiar la idoneidad del mismo y proponer otras alternativas.

Interés del Problema

Durante el proceso de fabricación de productos de acero, no todos los productos cumplen el nivel de calidad exigido por el cliente final. Estos productos, denominados “non-prime” se venden mediante una subasta online semanal, siendo necesario una agrupación previa de los mismos en lotes, de manera que éstos sean lo más homogéneos posibles atendiendo a varias características de los productos. Esta agrupación es un problema muy complejo de resolver, y es precisamente la parte fundamental del proceso de ventas, ya que cuanto más homogéneos sean los lotes, más atractivos serán para los clientes, que a su vez realizarán más pujas, incrementando el beneficio. Aproximadamente medio millón de toneladas se venden anualmente con este sistema en todas las plantas de productos planos del grupo. Resolver de una manera adecuada el problema representa, pues, un tremendo impacto económico, a la vez que un reto desde el punto de vista algorítmico, ya que no se ha resuelto ningún problema similar hasta la fecha.

Metodología

Inicialmente se realiza una descripción del problema, exponiendo primero todo el proceso de ventas de productos non-prime, centrándose en la parte de subastas y las reglas de agrupamiento que deben de cumplirse durante la preparación de los lotes, para finalizar describiendo los requisitos que debe de cumplir el algoritmo a desarrollar.

A continuación se analiza la literatura existente de subastas, tanto las principales variantes existentes como los principales factores que influyen en el proceso, para estudiar si el método de subasta elegido por la empresa (cerrada de primer precio, con precio reserva) es el más adecuado, o se pueden proponer alternativas.

Tras esto, para poder plantear un algoritmo de resolución del problema se realiza primero un estado del arte de la literatura existente del problema de empaquetado por lotes (Bin Packing Problem), analizando sus variantes y las técnicas utilizadas para resolverlo. Dado que el problema a resolver no coincide con ninguna de las variantes, se define este nuevo problema como “Problema de empaquetado de lotes homogéneos con mínimo peso”, se formula matemáticamente y se propone una metodología de resolución estructurada en 3 fases, consistentes en una división inicial de los productos basándose en sus características comunes, un algoritmo de clustering y una última fase de optimización de la asignación de productos a lotes finales, para la cual se desarrollan 3 heurísticos (FIFO, Greedy y Distancias) y un meta-heurístico GRASP.

Para comparar la bondad de la solución y dado que no existe un problema similar, se generan 80 instancias de prueba basadas tanto en las instancias clásicas de Bin Packing Problem, como en datos reales del problema a resolver, y se comparan los resultados obtenidos con cada heurístico y el meta-heurístico GRASP. Finalmente, se prueba el algoritmo con un caso de real para demostrar tanto su funcionamiento como su rendimiento.

Resultados

Del análisis teórico realizado sobre subastas se concluye que es coherente que la subasta sea cerrada y con precio reserva. Se propone como recomendación realizar una subasta descendente para aquellos productos que no se han vendido con la subasta inicial, para aumentar el beneficio global del sistema de venta.

En lo referente al problema de agrupamiento, tras analizar las pruebas realizadas sobre las 80 instancias generadas, se puede concluir que el algoritmo propuesto resuelve el problema planteado, siendo el meta-heurístico GRASP la mejor solución para la fase final, ya que mejora las soluciones obtenidas utilizando solamente los heurísticos individuales en el 84% de las instancias. Las mejores soluciones se han conseguido utilizando el algoritmo FIFO como generador de las soluciones iniciales del GRASP. Para su aplicación al caso real se

propone lanzar la tercera fase del algoritmo 2 veces, una con soluciones iniciales generadas por el FIFO y otra con soluciones iniciales generadas con el heurístico de Distancias. De esta forma, se consigue un tiempo de computación para 26 plantas y 2771 productos de unos 4 minutos y se garantiza una muy buena solución.

ABSTRACT

Objective

The aim of the doctoral thesis “Analysis and improvements through metaheuristics techniques of online multi-product auctions within the steel industry”, by D. Borja Ena Rodríguez and directed by Dr. D. Alberto Gómez Gómez is to improve the selling process of non-prime steel products, firstly by resolving the problem of grouping steel products into bundles, making these bundles as homogeneous as possible, under both product features and bundle size constraints, and secondly by analyzing all the selling process to study its performance and propose possible alternatives.

Interest of the Problem

During the Steel making process, not all the products achieve the quality required by the customers. These products, denoted “non-prime” products, are sold in an online auction that takes place every week. Before putting the products on auction, they have to be grouped in bundles, trying to make these bundles as homogeneous as possible. This bundle generation process is a very complex problem to solve, and its precisely the main part of the selling process, since the more homogeneous the bundles are, the more attractive to the customer and more bids will take place, resulting in the end in more profits for the company. Around half a million steel tons are currently sold with this system in all flat Europe plants in the group. Hence, being able to solve properly the problem represents a tremendous economic impact, apart from a challenge from the algorithmic point of view, since no similar problem has been solved so far.

Methodology

First, a description of the problem is done, explaining all the non-prime products selling process, focusing on the auction phase and the grouping rules that must be considered during the bundling preparation, and describing the requirements of the algorithm to be developed.

Afterwards, a survey of existing auctions state of the art of is done, analyzing both the main variants and the main factors that influence the process, to be able to check if the

current auction method used by the company (first-price sealed bid with hidden price) is the most adequate one, or if some alternatives can be proposed.

Next, to be able to propose an algorithm to solve the problem, an analysis is done of the state of the art of Bin Packing Problem to study all its different variants and the techniques used to solve it. Since the problem proposed in this thesis hasn't been analyzed before, the problem is defined as "Homogeneous Bin Packing Problem with Minimum Weight". This problem is formulated mathematically, and a methodology to solve it based in 3 phases is defined: initial division of products based on its common features, a clustering algorithm and finally an optimization phase of assigning products to final bundles. For this last phase 3 heuristics (FIFO, Greedy and Distances), as well as a GRASP meta-heuristics are developed.

To be able to ensure the quality of the solution, and since no similar problem exists so far, 80 tests instances are generated, based on the current literature of Bin Packing Problem, as well as in actual data of the problem to be solved. Each heuristic and the meta-heuristic is tested against the test instances to compare the results. Finally, the algorithm is tested with an actual use case to prove both its applicability and its performance.

Results

From the theoretical analysis done about auctions it is possible to conclude that is coherent to use a sealed-bid with reserve price auction. A proposal is done to incorporate a descending-bid auction for those products not sold in the initial auction to be able to improve the selling process benefit.

From the tests done with the 80 tests instances, it can be concluded that the developed algorithm solves the proposed problem, being the GRASP algorithm the best solution to use in the third phase. GRASP improve the solutions obtained with the individual heuristics in 84% of the test instances. The best solutions were obtained using FIFO heuristic as initial solution generator for the GRASP. For the use of the algorithm in the real case, it is proposed to launch this third phase of the algorithm with the GRASP twice: first using the FIFO as initial solution generator, and then using the Distances heuristic for the same purpose. This way, the computation time for 26 plants and 2771 products is around 4 minutes, ensuring a very good solution.

Índice de contenido

1	Introducción	12
1.1	Presentación	12
1.2	Objetivo de la tesis.....	14
1.3	Esquema general de la tesis.....	15
2	Descripción del problema.....	17
2.1	Introducción al sector del acero	17
2.1.1	Importancia del sector del acero	17
2.1.2	Descripción del proceso de fabricación del acero	20
2.1.2.1	Materias primas	20
2.1.2.2	Rutas de horno alto y de horno eléctrico.....	21
2.1.2.3	Laminación e instalaciones acabadoras	22
2.2	Subastas de productos planos “non-prime” en el sector del acero	25
2.2.1	Proceso de subastas.....	26
2.2.1.1	Tiempo de preparación de la subasta	28
2.3	Reglas de agrupamiento y homogeneidad en los lotes.....	29
2.3.1	Características comunes a todos los elementos de un lote.....	29
2.3.2	Características consideradas para la homogeneidad de un lote	30
2.4	Requisitos del algoritmo	34
3	Análisis de subastas.....	36
3.1	Principales variantes	37
3.2	Rendimiento	42
3.3	Factores humanos.....	45
3.3.1	Aversión al riesgo	45
3.3.2	Arrepentimiento de los pujadores.....	46
3.3.3	Colusión.....	47
3.3.4	Fraude y reputación	48
3.4	Factores a considerar al diseñar una subasta online.....	49
3.4.1	Elección del mecanismo de subasta.....	49

3.4.2	Restricciones sobre las pujas	51
3.4.3	Duración	52
3.5	Agrupaciones de productos en lotes para subastas	54
3.6	Recomendaciones propuestas.....	57
4	Problema del empaquetado en recipientes (Bin Packing Problem).....	60
4.1	Descripción del Bin packing problem.....	60
4.2	Variaciones del problema clásico	62
4.2.1	Versiones duales del problema.....	63
4.2.2	Tamaño variable de los recipientes	64
4.2.3	Variaciones en la asignación	64
4.2.4	Restricciones sobre los artículos	67
4.2.5	Generalizaciones del problema.....	68
4.2.6	Conclusiones y similitud de las variantes con el problema a analizar	69
4.3	Principales heurísticos utilizados en problemas Bin Packing	71
4.3.1	Principales algoritmos para la variante online.....	71
4.3.2	Algoritmos semi-online	73
4.3.3	Principales algoritmos para la variante offline	74
4.4	Bin Packing con Conflictos (Bin packing with Conflicts)	76
4.4.1	Definición del problema.....	76
4.4.2	Algoritmos existentes.....	77
4.4.2.1	First fit decreasing modificado (Modified FFD).....	77
4.4.2.2	Aproximación asintótica.....	78
4.4.2.3	Heurísticos basados en coloreado de grafos.....	79
4.4.2.4	Heurísticos basados en cliques.....	81
4.4.2.5	Heurístico adaptado de menor hueco en lote	83
4.4.2.6	Branch-and-price	84
4.4.2.7	Heurístico de aproximación voraz.....	84
4.4.3	Conclusiones extraídas de la literatura.....	85
5	Estudio previo sobre clusterización.....	87
5.1	Medidas de distancia y similitud.....	87
5.2	Técnicas de clusterización	90
5.2.1	Clusterización jerárquica.....	90

5.2.2	Clusterización basada en el error cuadrático (cuantificación vectorial)	91
5.2.3	Clusterización difusa	93
5.2.4	Clusterización basada en técnicas de búsqueda combinatoria	94
5.2.5	Clusterización basada en grafos.....	94
6	Solución propuesta.....	99
6.1	Problema de Empaquetado de Lotes Homogéneos con Mínimo Peso	99
6.2	Pre-procesado de datos y división inicial en conjuntos.....	103
6.3	Clusterización por homogeneidad	105
6.3.1	Distancia definida entre productos.....	105
6.3.1.1	Normalización y transformación de variables categóricas	106
6.3.1.2	Influencia dinámica de los parámetros sobre la distancia	109
6.3.2	Algoritmo de clustering.....	110
6.3.2.1	Parámetros de entrada	111
6.3.2.2	Estructura del algoritmo.....	111
6.3.2.3	Ejemplo de influencia de los parámetros.....	116
6.4	Asignación de productos a lotes.....	129
6.4.1	Pruebas previas	129
6.4.2	Estructura del algoritmo	133
6.4.3	Parámetros de entrada	135
6.4.4	Función de coste	136
6.4.5	Heurístico FIFO.....	138
6.4.6	Heurístico Greedy	139
6.4.7	Heurístico basado en distancias.....	141
6.4.8	Meta-heurístico GRASP.....	143
7	Diseño de los experimentos de prueba.....	147
7.1	Instancias de prueba en la literatura	147
7.1.1	Instancias del problema de empaquetado en recipientes.....	147
7.1.2	Instancias del problema de empaquetado en recipientes con conflictos	148
7.2	Generación de instancias de prueba para el HBPPMW.....	150
8	Pruebas realizadas y validación de la solución.....	158
8.1	Análisis del número de repeticiones de los heurísticos.....	158

8.2	Experimentos realizados con las instancias de prueba	160
8.2.1	Comportamiento de las 3 heurísticas	160
8.2.1.1	Función de coste	162
8.2.1.2	Número de lotes y productos sin asignar	166
8.2.1.3	Homogeneidad de los lotes.....	178
8.2.1.4	Tiempo computacional.....	180
8.2.1.5	Resumen de las pruebas con los heurísticos.....	181
8.2.2	Comportamiento del GRASP	181
8.2.2.1	Rendimiento del GRASP	182
8.2.2.2	Función de coste	187
8.2.2.3	Tiempo computacional.....	192
8.2.2.4	Resumen de las pruebas con el GRASP	193
8.3	Experimento realizado con un caso de prueba real	194
8.3.1	Datos de entrada.....	194
8.3.2	Configuración del algoritmo.....	194
8.3.3	Solución.....	195
9	Conclusiones y líneas de futuro.....	199
9.1	Resumen de las principales aportaciones.....	199
9.2	Líneas de futuro	202
Anexo	203
Apéndice A.	Instancias para las pruebas previas buscando la solución óptima	203
Apéndice B.	Ejemplos de instancias de Benchmarking.....	208
Apéndice C.	Resultados de los experimentos realizados con las instancias generadas	219
Bibliografía	244

Índice de figuras

Figura 1. Producción mundial de acero en 2016 y esperada en 2017 (World Steel Association)	18
Figura 2. Esquema del proceso de fabricación de acero (ArcelorMittal: el proceso siderúrgico)	24
Figura 3. Proceso de asignación de productos “non-prime”	26
Figura 4. Esquema general del proceso de subastas de productos non-prime	27
Figura 5. Similitud de grados de acero – Familia HR	32
Figura 6. Similitud de grados de acero – Familia CR	32
Figura 7. Similitud de grados de acero – Familia EG	32
Figura 8. Similitud de grados de acero – Familia HD	32
Figura 9. Similitud de grados de acero – Familia OC	33
Figura 10. Nueva propuesta de modelo de ventas	58
Figura 11. Clasificación propuesta de las variantes del Bin Packing Problem	62
Figura 12. Fases de resolución del problema	102
Figura 13. Ejemplo de representación en el espacio de las dimensiones “espesor” y “grado de acero”	106
Figura 14. Ejemplos de grupos de grado de acero para la familia HR	108
Figura 15. Distancias para un grupo de 36 elementos	110
Figura 16. Proceso de clustering de un conjunto	114
Figura 17. Esquema del proceso de clustering	115
Figura 18. Estructura del algoritmo de clustering	116
Figura 19. Evolución de la salida del algoritmo de clustering al variar el threshold	120
Figura 20. Evolución de la salida del algoritmo de clustering al variar el MinSize	123
Figura 21. Evolución de la salida del algoritmo de clustering al variar el MaxSize	127
Figura 22. Salida del algoritmo de clustering en una ejecución con datos reales de 1 semana	128

Figura 23. Evolución temporal con el número de elementos para la solución exacta	132
Figura 24. Estructura del algoritmo CREATE_BUNDLES	134
Figura 25. Pseudocódigo de la función CAN_ADD	135
Figura 26. Pseudocódigo del heurístico FIFO	139
Figura 27. Pseudocódigo del heurístico Greedy.....	141
Figura 28. Pseudocódigo del heurístico basado en distancias.....	142
Figura 29. Esquema del algoritmo GRASP.....	143
Figura 30. Esquema de la búsqueda local del GRASP	144
Figura 31. Función de búsqueda en vecindario	144
Figura 32. Función para intercambiar productos entre lotes	145
Figura 33. Función de asignación de elementos sueltos a lotes	146
Figura 34. Valores de espesor de bobinas reales.....	152
Figura 35. Función de densidad de probabilidad del espesor.....	153
Figura 36. Valores de ancho de bobinas reales.....	153
Figura 37. Función de densidad de probabilidad del ancho	154
Figura 38. Valores de espesor de recubrimiento (lado 1) de bobinas reales.....	155
Figura 39. Valores de espesor de recubrimiento (lado 2) de bobinas reales.....	155
Figura 40. Función de densidad de probabilidad del espesor de recubrimiento lado 1.....	156
Figura 41. Función de densidad de probabilidad del espesor de recubrimiento lado 2.....	156
Figura 42. Evolución del tiempo de computación con 50, 100, 200 y 500 repeticiones	159
Figura 43. Esquema del algoritmo y número de repeticiones empleadas en las instancias de prueba	161
Figura 44. Distribución de mejores soluciones para cada tamaño de instancias	166
Figura 45. Comparativa de lotes generados para las instancias de 120 productos.....	167
Figura 46. Comparativa de elementos sin asignar para las instancias de 120 productos	168

Figura 47. Comparativa de lotes + elementos sin asignar para las instancias de 120 productos	168
Figura 48. Comparativa de lotes generados para las instancias de 250 productos.....	170
Figura 49. Comparativa de elementos sin asignar para las instancias de 250 productos	170
Figura 50. Comparativa de lotes + elementos sin asignar para las instancias de 250 productos	171
Figura 51. Comparativa de lotes generados para las instancias de 500 productos.....	172
Figura 52. Comparativa de elementos sin asignar para las instancias de 500 productos	173
Figura 53. Comparativa de lotes + elementos sin asignar para las instancias de 500 productos	173
Figura 54. Comparativa de lotes generados para las instancias de 1000 productos.....	175
Figura 55. Comparativa de elementos sin asignar para las instancias de 1000 productos	175
Figura 56. Comparativa de lotes + elementos sin asignar para las instancias de 1000 productos	176
Figura 57. Heurísticos con mejores números de lotes y menor número de elementos sin asignar	177
Figura 58. Heurísticos con mejor número de lotes + número de elementos sin asignar	177
Figura 59. Heurísticos con mejor homogeneidad en las instancias de prueba.....	178
Figura 60. Evolución del tiempo de ejecución para cada heurística	180
Figura 61. Distribución de mejores soluciones con el GRASP y cada heurístico	191
Figura 62. Evolución del tiempo de ejecución	193

Índice de tablas

Tabla 1. Importaciones y exportaciones mundiales por país (World Steel Association)	19
Tabla 2. Variantes de subastas más utilizadas, con sus principales características (elaboración propia).....	41
Tabla 3. Comparativa de estudios realizados de subasta por lotes frente a productos individuales (elaboración propia).....	55
Tabla 4. Medidas de similitud más usuales.....	89
Tabla 5. Distancias asignadas para productos de la familia OC	107
Tabla 6. Distancias asignadas para productos de la familia HD	108
Tabla 7. Efecto de variación de threshold en el algoritmo de clustering.....	120
Tabla 8. Efecto de variación de MinSize en el algoritmo de clustering	123
Tabla 9. Efecto de variación de MaxSize en el algoritmo de clustering.....	126
Tabla 10. Valores de la función de coste con 50, 100, 200 y 500 repeticiones del heurístico Greedy.....	159
Tabla 11. Valores de la función de coste para cada heurístico con las instancias de 120 productos	162
Tabla 12. Valores de la función de coste para cada heurístico con las instancias de 250 productos	163
Tabla 13. Valores de la función de coste para cada heurístico con las instancias de 500 productos	164
Tabla 14. Valores de la función de coste para cada heurístico con las instancias de 1000 productos	165
Tabla 15. Lotes y productos sin asignar para cada heurístico con las instancias de 120 productos	167
Tabla 16. Lotes y productos sin asignar para cada heurístico con las instancias de 250 productos	169

Tabla 17. Lotes y productos sin asignar para cada heurístico con las instancias de 500 productos	172
Tabla 18. Lotes y productos sin asignar para cada heurístico con las instancias de 1000 productos	174
Tabla 19. Comparativa de mejores soluciones frente a homogeneidad y número de lotes + elementos sueltos	179
Tabla 20. Tiempos medios de ejecución de las instancias de prueba con cada heurística.....	180
Tabla 21. Comparativa de soluciones con los heurísticos frente al GRASP para 120 productos	182
Tabla 22. Comparativa de soluciones con los heurísticos frente al GRASP para 250 productos	183
Tabla 23. Comparativa de soluciones con los heurísticos frente al GRASP para 500 productos	184
Tabla 24. Comparativa de soluciones con los heurísticos frente al GRASP para 1000 productos	185
Tabla 25. Mejor solución obtenida con el GRASP frente a cualquier heurístico para cada instancia	186
Tabla 26. Valores de la función de coste del GRASP con cada heurístico con las instancias de 120 productos	187
Tabla 27. Valores de la función de coste del GRASP con cada heurístico con las instancias de 250 productos	188
Tabla 28. Valores de la función de coste del GRASP con cada heurístico con las instancias de 500 productos	189
Tabla 29. Valores de la función de coste del GRASP con cada heurístico con las instancias de 1000 productos	190
Tabla 30. Tiempos medios de ejecución de las instancias de prueba.....	192
Tabla 31. Distribución de elementos en clústeres	196
Tabla 32. Comparativa de algoritmo ante un caso real con threshold = 1'5 y threshold = 1 ...	198

Tabla 33. Pesos de los productos utilizados para las pruebas previas.....	203
Tabla 34. Distancias entre cada par de productos utilizadas en las pruebas previas	207
Tabla 35. Instancia de prueba 120_00	211
Tabla 36. Instancia de prueba 250_00	218
Tabla 37. Resultados FIFO 120 productos.....	220
Tabla 38. Resultados FIFO 250 productos.....	221
Tabla 39. Resultados FIFO 500 productos.....	222
Tabla 40. Resultados FIFO 1000 productos.....	223
Tabla 41. Resultados Greedy 120 productos	224
Tabla 42. Resultados Greedy 250 productos	225
Tabla 43. Resultados Greedy 500 productos	226
Tabla 44. Resultados Greedy 1000 productos	227
Tabla 45. Resultados Distancias 120 productos.....	228
Tabla 46. Resultados Distancias 250 productos.....	229
Tabla 47. Resultados Distancias 500 productos.....	230
Tabla 48. Resultados Distancias 1000 productos.....	231
Tabla 49. Resultados FIFO + GRASP 120 productos	232
Tabla 50. Resultados FIFO + GRASP 250 productos	233
Tabla 51. Resultados FIFO + GRASP 500 productos	234
Tabla 52. Resultados FIFO + GRASP 1000 productos	235
Tabla 53. Resultados Greedy + GRASP 120 productos.....	236
Tabla 54. Resultados Greedy + GRASP 250 productos.....	237
Tabla 55. Resultados Greedy + GRASP 500 productos.....	238
Tabla 56. Resultados Greedy + GRASP 1000 productos.....	239
Tabla 57. Resultados Distancias + GRASP 120 productos	240

Tabla 58. Resultados Distancias + GRASP 250 productos	241
Tabla 59. Resultados Distancias + GRASP 500 productos	242
Tabla 60. Resultados Distancias + GRASP 1000 productos	243

1 Introducción

1.1 Presentación

La presente tesis tiene como objetivo resolver un problema real de una empresa siderúrgica mediante técnicas de optimización. Cada semana, gran cantidad de productos de acero “non-prime” (de segunda calidad) de varias plantas de Europa se venden online a través de un mecanismo de subasta. Cada producto no es subastado de manera individual, sino que previamente se ha de hacer una agrupación de los mismos de acuerdo a una serie de características y restricciones. Realizar esta agrupación es una tarea muy compleja, y a la vez, es la que más impacto tiene sobre el precio final de puja de cada lote debido a dos factores: por un lado la propia agrupación de productos con diferentes características tiene un valor intrínseco, y por otro lado, lo atractivo que pueda resultar el lote para los pujadores influye decisivamente en el coste final. Estos hechos hacen que el agrupamiento de los productos sea un aspecto clave del proceso de venta mediante subasta.

En esta tesis se analiza en profundidad este problema, para determinar todas las características que se deben considerar de cara a realizar una agrupación óptima de los productos, así como el proceso completo de subastas para comprender los diferentes factores que influyen. Se analizan los estudios más relevantes, así como las técnicas y aproximaciones utilizadas para resolver este tipo de problemas de agrupación de productos en lotes, lo que servirá de base para proponer una solución al problema.

Desde el punto de vista algorítmico, se propone una solución basada en diversas fases que permitan abordar la complejidad del problema, basándose en los modelos existentes en la literatura para abordar problemas menos complejos. Se desarrollan varios algoritmos diferentes para analizar el comportamiento en cuanto a calidad de resultados (número de lotes y homogeneidad de los mismos) y tiempo de ejecución. Para poder evaluar los resultados de los algoritmos propuestos, previamente se analizan las instancias utilizadas en problemas similares en la literatura, adaptando estas instancias para servir de base al problema a resolver en esta tesis. De igual forma, para demostrar la aplicabilidad del algoritmo propuesto se

prueba ante un caso real, tomando como entrada todos los productos que se subastarán una semana para todas las plantas de productos planos de Europa.

Por otro lado, se analiza el proceso de subastas en el sector del acero teniendo en cuenta cuales son las principales variantes de subastas que se pueden utilizar y el rendimiento teórico de las mismas, así como los principales factores humanos que pueden alterar el beneficio esperado. Tras analizar la literatura existente sobre el tema, se proponen recomendaciones para no solamente optimizar la agrupación de los productos para la subasta, sino para mejorar de manera global todo el proceso.

1.2 Objetivo de la tesis

El principal objetivo de la tesis es resolver el problema de agrupación de productos siderúrgicos en lotes lo más homogéneos posibles y bajo restricciones tanto de características de los productos como de tamaño de lote, para su posterior subasta online. A su vez, esto engloba diferentes sub-objetivos:

- Análisis de los principales estudios realizados sobre las variantes de subastas implicadas en el problema, de cara a analizar si el método de subasta utilizado es el idóneo.
- Propuesta de alternativas a utilizar durante el proceso de venta de productos “non-prime”.
- Análisis de las variantes de problemas de agrupación de productos en lotes abordados en la literatura, y del estado del arte de las diferentes técnicas y estrategias utilizadas para resolverlos.
- Propuesta de una formulación matemática del problema a resolver.
- Propuesta de una solución que permita abordar en un tiempo razonable el problema de agrupación por lotes de productos heterogéneos.
- Diseño de un método que permita comparar cómo de parecidos son dos o varios productos en función de sus múltiples características.
- Propuesta de una solución que permita dar más prioridad a unas características de los productos que a otras a la hora de formar lotes homogéneos.
- Desarrollo de varias heurísticas / meta-heurísticas y comparación de las mismas en cuanto a tiempo de ejecución y calidad de las soluciones.
- Propuesta de instancias para validar los resultados del modelo

1.3 Esquema general de la tesis

La presente tesis está organizada en 9 capítulos incluyendo éste.

En el **capítulo 2** se describe el problema a resolver en la tesis, enfocándolo en el marco del proceso de fabricación y venta de productos de acero y resaltando la importancia del mismo. Para ello se describe brevemente el proceso de fabricación para a continuación exponer en detalle el proceso de venta mediante subastas de productos planos “non-prime”, resaltando la influencia en los beneficios de una agrupación homogénea previa. Para finalizar el capítulo, se analizan las restricciones que se deben de tener en cuenta a la hora de realizar dichas agrupaciones, y se detallan los requisitos que debe de tener la solución propuesta.

Dentro del **capítulo 3** se analizan los diferentes trabajos existentes en la literatura referentes al tipo de subasta utilizado para la venta de productos “non-prime”: cerrada de primer precio, online y multi-unidad. Aunque actualmente la empresa no se plantea modificar el sistema, el objetivo es analizar las variables influyentes, así como los factores a considerar en cada una, para analizar si el método utilizado actualmente es el idóneo y proponer posibles métodos alternativos que podrían ser utilizados en el futuro. Para finalizar el capítulo se incluye un estudio sobre la idoneidad de realizar agrupaciones de productos previas a la subasta, así como las conclusiones de las observaciones.

Posteriormente, en el **capítulo 4** se realiza un análisis detallado de los estudios existentes sobre el Bin Packing Problem, debido a que la problemática planteada en esta tesis encaja en el marco de estos problemas de optimización combinatoria. Primero se realiza una descripción del problema general, para después analizar las diferentes variantes del mismo, tratando de encontrar la analogía con nuestro caso. De igual forma, se analizan las técnicas de resolución utilizadas para abordar este tipo de problemas, para partir de esa información antes de proponer una solución al caso de la tesis.

El **capítulo 5** contiene un resumen de las principales técnicas de clusterización utilizadas en la literatura, debido a su importancia potencial como parte de una futura solución al problema planteado.

En el **capítulo 6** se desarrolla la solución planteada en la tesis para resolver el problema desde el punto de vista algorítmico, desarrollando los diferentes heurísticos y meta-heurísticos utilizados.

En el **capítulo 7** se analizan las instancias de prueba utilizadas en problemas similares en la literatura, y se plantean unas instancias adaptadas al problema de la tesis.

Posteriormente, en el **capítulo 8** se recogen las pruebas realizadas con las instancias desarrolladas, y se comparan las soluciones obtenidas con los algoritmos propuestos, analizando los resultados tanto en calidad de la solución como en tiempo computacional. Para finalizar, se prueba el algoritmo ante un caso real, mostrando las soluciones obtenidas.

Por último, en el **capítulo 9**, se enumeran las aportaciones de este trabajo, las conclusiones alcanzadas y las líneas futuras de investigación surgidas durante el desarrollo de esta tesis.

2 Descripción del problema

En este capítulo se realiza una descripción del problema a resolver en la tesis. Primeramente se realiza una introducción general del mundo del acero, incluyendo datos sobre su influencia económica y explicando el proceso de fabricación del mismo, para a continuación centrarse en el proceso de subastas de productos planos “non-prime”. Se explica en detalle el proceso de subastas, la importancia y los volúmenes que se venden con este procedimiento, y se muestra la importancia de realizar una agrupación previa de los productos en lotes antes de ser sacados a subasta. Para finalizar, se detallan todo el conjunto de restricciones que se deben de tener en cuenta a la hora de realizar las agrupaciones de productos y se especifican los requisitos que debe de tener la solución propuesta.

2.1 Introducción al sector del acero

En este apartado se realiza una introducción general sobre el sector del acero, analizando su influencia en la economía actual, tanto de manera directa como indirecta a través de otros sectores involucrados en el proceso, y se muestran datos de volúmenes producidos, importados y exportados en el mundo. Se explica también de manera genérica el proceso de fabricación, así como las principales instalaciones y materias primas utilizadas en el mismo.

2.1.1 Importancia del sector del acero

El sector del acero es uno de los principales pilares de la economía mundial. El acero es la pieza fundamental de otros sectores como la construcción, el automóvil, los transportes, energía, alimentación, agua, herramientas y maquinaria. Las principales ventajas del acero son sus excelentes propiedades mecánicas y estructurales (moldeable en frío y caliente, soldable, duro y resistente a corrosión) junto con un proceso de fabricación relativamente barato comparado con el resto de materiales (la energía requerida para extraer hierro del mineral es un 25% menor que la requerida para el aluminio). El acero es amistoso con el medio ambiente

ya que es 100% recuperable debido a sus propiedades magnéticas, y 100% reciclable (Johnson, et al., 2008). Además, parte de un mineral que tiene alta disponibilidad, ya que el hierro representa el 5,6% de la corteza terrestre, y existen minas distribuidas por todo el mundo. Las características finales del material dependen de las propiedades químicas del grado de acero: actualmente se desarrollan alrededor de unos 2000 grados de acero diferente (Thelning, 2013), en función de las características finales del producto.

Los diferentes productos de acero se clasifican en dos grandes grupos: planos y largos. Dentro de los productos planos se incluyen desbastes (slabs), bobina (caliente, fría, galvanizada), hojalata y chapa gruesa. Estos productos se utilizan principalmente en los sectores del automóvil, maquinaria pesada, tuberías, construcción, empaquetado y electrodomésticos. Los productos largos incluyen planchones, palanquilla, carril, alambρόn y secciones, siendo sus principales mercados la construcción, la ingeniería mecánica, la energía y el transporte.

Según datos de la asociación mundial del acero, en 2016 se fabricaron casi 1500 millones de toneladas de acero, y las predicciones para el 2017 apuntan a que el consumo seguirá estable (un 0.4% más de aumento). En la Figura 1 se muestra la producción especificada por zonas geográficas.

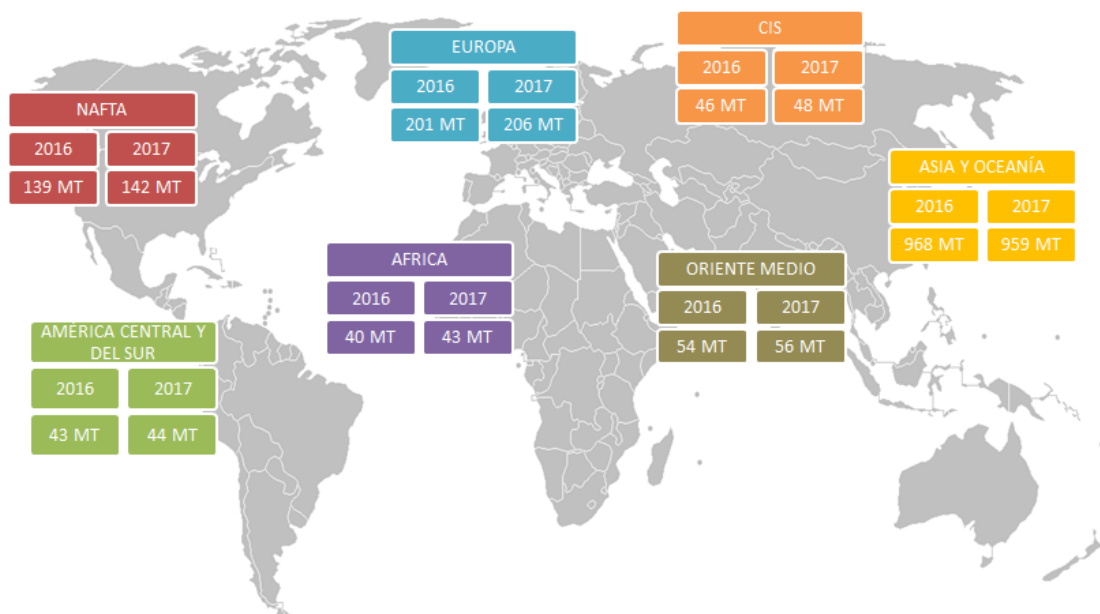


Figura 1. Producción mundial de acero en 2016 y esperada en 2017 (World Steel Association)

Los volúmenes de acero intercambiados entre los diferentes países tienen un gran impacto en las diferentes economías. En la Tabla 1 se muestran los principales países exportadores e importadores de acero, y las toneladas que mueven fuera de sus fronteras o importan del exterior (WorldSteelAssociation, 2016).

Ranking	Total exportaciones	MT	Ranking	Total importaciones	MT
1	China	111.6	1	Unión Europea (28)	37.7
2	Japón	40.8	2	Estados Unidos	36.5
3	Unión Europea (28)	33.8	3	Alemania	24.8
4	Corea del Sur	31.2	4	Corea del Sur	21.7
5	Rusia	29.7	5	Italia	19.9
6	Alemania	25.1	6	Turquía	18.6
7	Ucrania	17.7	7	Vietnam	16.3
8	Italia	16.5	8	Tailandia	14.6
9	Bélgica	15.2	9	Francia	13.7
10	Turquía	15.0	10	India	13.3
11	Francia	14.0	11	China	13.2
12	Brasil	13.7	12	México	12.7
13	Taiwan	11.2	13	Bélgica	12.1
14	Holanda	10	14	Indonesia	11.4
15	Estados Unidos	10.0	15	Polonia	9.2
16	España	9.6	16	España	8.9
17	India	7.6	17	Canadá	8.0
18	Austria	7.4	18	Egipto	7.9
19	Reino Unido	7.3	19	Taiwan	7.5
20	Canadá	6.0	20	Reino Unido	7.2

Tabla 1. Importaciones y exportaciones mundiales por país (World Steel Association)

La industria del acero da empleo directo a más de dos millones de personas en el mundo, además de otros dos millones de contratistas y otros cuatro millones en industrias auxiliares. Es una fuente de empleo para 50 millones de personas que trabajan en empresas de construcción, transporte y energía.

En España existen 22 plantas de producción de acero. Además, existen 50 instalaciones de laminación y transformación. Aunque la mayor concentración se sitúa en la cornisa cantábrica, especialmente Asturias y País Vasco, existen fábricas en once Comunidades Autónomas.

2.1.2 Descripción del proceso de fabricación del acero

En este apartado se detallan las materias primas utilizadas para la fabricación del acero y el proceso de fabricación del mismo a través de sus dos rutas posibles: utilización de horno alto o mediante horno eléctrico.

2.1.2.1 Materias primas

Las materias primas fundamentales requeridas en los procesos de fabricación de arrabio y acero son:

- Materias primas férricas (minerales de hierro, chatarra y pre-reducidos).
- Materias primas combustibles y agentes reductores (carbón, cok siderúrgico, gases y otros combustibles).
- Materias primas fundentes (caliza, dolomía, cal, elementos para aleación, etc.).

El mineral de hierro es una materia prima fundamental para la siderurgia. Existen diversos tipos de minerales de hierro (magnetita, limonita, siderita, etc.) fundamentalmente distinguidos por su composición de hierro y oxígeno. Se debe enriquecer el mineral extraído reduciendo su contenido en galga mediante lavado, separación por flotación u otros procedimientos, obteniéndose al final minerales gruesos que podrían ser usados como materia prima en el horno alto, y materiales finos, que también pueden ser usados en el horno alto

previo paso por procesos de sinterización o de pelletización. Estos materiales finos una vez tratados sirven para mejorar la calidad del arrabio.

La chatarra es la materia prima más usada en la industria del acero, cubriendo del orden del 40% de sus necesidades globales. Su cantidad depende de que el acero se fabrique mediante un convertidor LD o mediante horno eléctrico. La calidad de la chatarra se define por su contenido en hierro, por su contenido en elementos residuales (cromo, níquel, estaño, cinc, etc.) y por sus condiciones físicas, como densidad, distribución de tamaños, etc.

El objetivo principal del horno alto es romper las moléculas del óxido, para liberar al hierro del oxígeno con el que está combinado. Para ello se emplea un elemento que tiene más tendencia que el hierro a combinarse con el oxígeno (que reduzca al hierro): el carbono. En la naturaleza el carbón es la sustancia que más carbono aporta. El carbón es una sustancia heterogénea, de composición muy variable, según su procedencia. De menor a mayor rango los carbones se pueden dividir en: turbas, lignitos, hullas y antracitas. El más importante para el proceso metalúrgico es la hulla, que es la materia prima de la que se obtiene el cok.

Sin embargo, el carbón no puede ser agente reductor en el lecho de fusión del horno alto por varias razones: falta de resistencia mecánica necesaria para soportar la carga férrica, falta de porosidad (sin la cual no se favorecen las reacciones químicas entre los gases y la carga sólida), y elevado contenido en volátiles, lo que daría lugar a un exceso de gases en el tragante del horno. Por estos motivos en el lecho del horno alto se emplea un producto resultante de someter el carbón a un tratamiento térmico, este producto es el cok. A veces el carbón pulverizado también se emplea en el horno alto como combustible sustituto del cok en pequeña proporción. Dentro del horno, el cok funciona a la vez como combustible (genera el calor necesario), reductor (suministrandó el carbono para atrapar el oxígeno combinado inicialmente con el hierro) y soporte físico de las cargas férricas (responsable de la resistencia de la mezcla).

2.1.2.2 Rutas de horno alto y de horno eléctrico

Existen dos métodos de fabricación de acero: la ruta de horno alto (siderurgia integral) y la ruta de horno eléctrico. A continuación se resumen los dos diferentes métodos de fabricación.

El proceso más común de generación de acero es utilizando horno alto y convertidor de soplado de oxígeno (BOF). En el convertidor de soplado de oxígeno, el hierro se combina con chatarra de acero (en cantidades menores al 30%) junto con fundentes. Se introduce una lanza que sopla oxígeno puro, incrementando la temperatura a 1700 °C. La chatarra se funde, las impurezas se oxidan y el contenido de carbón se reduce en un 90%, resultando al final del proceso en acero líquido. A continuación pueden aplicarse procesos de metalurgia secundaria, donde las propiedades del acero se determinan mediante la adición de otras aleaciones y componentes (Boro, Cromo, Molibdeno...) para garantizar las especificaciones exactas del grado de acero. Los convertidores de soplado de oxígeno representan actualmente el 70% de la producción mundial de acero. El resto se produce mediante hornos eléctricos de arco.

El proceso de fabricación de acero mediante horno eléctrico (EAF) no necesita fabricación previa de arrabio. Reutiliza acero existente, con lo que también evita la necesidad de materias primas y su procesado (generación de cok y sinter). El horno se carga con chatarra de acero, y en ocasiones se añade también hierro de reducción directa o acero fundido para controlar el balance térmico. La corriente eléctrica entre dos electrodos del horno proporciona el calor necesario al proceso hasta elevar la temperatura a unos 1600 °C, fundiendo la chatarra. Para eliminar las impurezas se añaden fundentes que formarán la escoria..

La diferencia más evidente es la diversidad del suministro energético: en el convertidor por soplado de oxígeno, como la materia prima ya es líquida el calor necesario para el proceso es menor, y viene liberado por las reacciones de oxidación que tienen lugar en el convertidor. La energía necesaria en el horno eléctrico es muy superior, ya que la materia prima es sólida. Esta energía es, naturalmente, eléctrica. En el proceso de horno alto las materias primas son arrabio (80-90%) y chatarra (10-20%), mientras que en el horno eléctrico se usa chatarra (70-80%) y pre-reducidos de hierro (20-30%). La ruta de horno alto es útil para la fabricación de aceros de bajo carbono, mientras que el horno eléctrico de arco puede producir aceros de alta aleación.

2.1.2.3 Laminación e instalaciones acabadoras

Una vez que el acero ha sido producido en un convertidor de soplado de oxígeno o en un horno eléctrico de arco, el material es colado y se enfría en forma de desbastes. Estos desbastes son tratados en una serie de instalaciones denominadas instalaciones acabadoras,

ya que darán la forma y propiedades estructurales finales que tendrá el material. Las principales líneas e instalaciones por las que pueden pasar los productos planos (bobinas, planchones) son:

- Tren de bandas en caliente: en esta instalación se laminan los desbastes mediante una serie de cajas, reduciendo el espesor de la banda y enrollándola finalmente en bobinas.
- Decapado: se realiza una limpieza de la capa de óxido que se ha formado en la laminación en caliente. Se cortan además la cabeza, cola y bordes de la bobina para que estén mejor dimensionadas en los procesos posteriores.
- Tándem: la banda se somete a un proceso de laminación en frío bajo presión, con el objetivo de reducir el espesor hasta las dimensiones requeridas por el uso final del producto.
- Recocido: a consecuencia de la laminación en frío, el material pierde parte de sus propiedades metalúrgicas, con lo que es necesario someterlo de nuevo a un proceso de recocido para regenerar su estructura interna.
- Temperizado: en este proceso se mejora la planitud y se conforma al producto con la dureza y acabado superficial requerido.
- Galvanizado: las bobinas son recubiertas con zinc y aleaciones, para proteger al acero frente a los fenómenos de corrosión.

En cuanto a los productos largos, a partir de los desbastes iniciales suelen pasar directamente a la línea propia donde se fabricará el producto final. Las principales son:

- Tren de Chapa gruesa: suele estar constituido por una única caja reversible en donde se preparan las chapas del tamaño adecuado.
- Alambrón: transforma las palanquillas (desbastes) en alambrón liso o corrugado con las dimensiones requeridas. Durante el proceso se elimina la cascarilla y se realizan inspecciones para comprobar la calidad final del producto.

- Carril: los desbastes se moldean con la forma del carril utilizando cajas dúo con la forma y especificaciones requeridas.

La Figura 2 muestra un resumen del proceso siderúrgico utilizando tanto la ruta de horno alto como de horno eléctrico de arco.

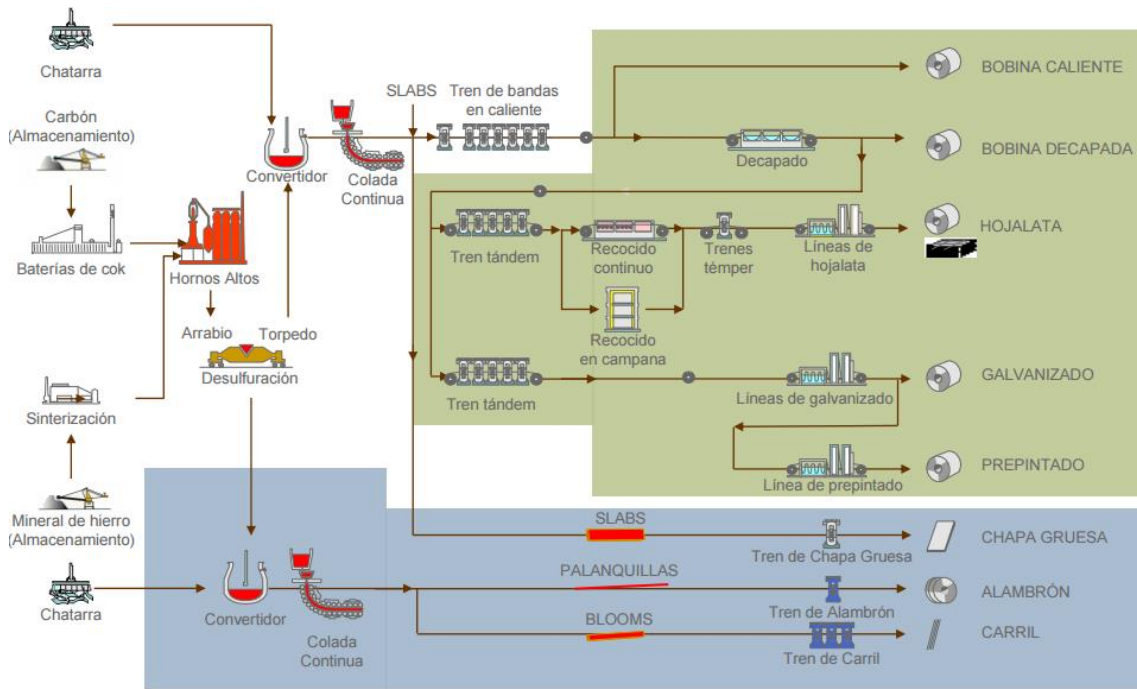


Figura 2. Esquema del proceso de fabricación de acero (ArcelorMittal: el proceso siderúrgico)

2.2 Subastas de productos planos “non-prime” en el sector del acero

Durante el proceso de fabricación de productos planos de acero (bobinas, chapa) se sigue un exhaustivo control de la calidad de los mismos, para garantizar que cada producto llegue a su comprador con las características de composición química, dureza y recubrimiento acordadas. En el caso de los productos de mayor calidad, debido a los diferentes tratamientos químicos y físicos propios del proceso de fabricación, y a veces a imprevistos no contemplados, no todas las bobinas que se fabrican cumplen todos los requisitos deseados para llegar al nivel de calidad exigido, por lo que no se pueden vender a los clientes originales. Un ejemplo de esto son las bobinas destinadas a automóvil, que deben cumplir los más estrictos niveles de calidad. Si una bobina producida para el sector del automóvil no cumple todos los requisitos, no se podrá vender a ese cliente, pero la bobina se podrá aprovechar y vender a otro fabricante de otro producto en acero que no necesite ese grado de calidad máximo.

Estos productos, denominados “non-prime” (no llegan a la calidad exigida, o presentan algún defecto físico) se ofrecen en un primer momento a empresas afiliadas y clientes habituales (normalmente de la región local asociada a cada planta), o algunos se envían a otras plantas para utilizar en otras líneas (por ejemplo, si una planta no tiene instalaciones acabadoras, una bobina caliente que no tenga la calidad exigida se puede enviar a otra planta del grupo para recibir un tratamiento estructural o de recubrimiento y destinarla a otro consumidor final). Los productos de todas las plantas que no se venden directamente son subastados online, agrupando cada semana los productos de todas las plantas de Europa. Si los productos no se consiguen vender después de un determinado número de semanas, éstos son devueltos a la planta para convertirlos en chatarra y reaprovecharlos en el proceso. En la Figura 3 se representa un esquema básico del flujo de asignación de estos productos.

El proceso de subasta representa pues, una parte fundamental de la venta de productos. Cada semana se subastan online productos de todas las plantas de productos planos de Europa, existiendo actualmente unos 300 clientes habituales. Aproximadamente

unas 10.000 Toneladas de acero se venden semanalmente mediante este sistema. En el siguiente apartado se explica en detalle el proceso de subasta online de los productos.

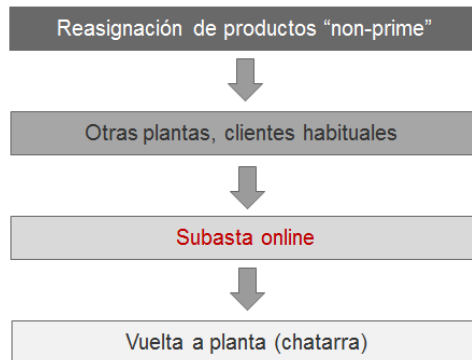


Figura 3. Proceso de asignación de productos "non-prime"

2.2.1 Proceso de subastas

El proceso completo de subastas de productos engloba los siguientes pasos:

- 1) Preparación de los productos para la subasta: los productos no salen todos individualmente a subasta, sino que previamente se hace una agrupación en lotes ("bundles"). Los productos se han de agrupar en lotes lo más homogéneos posibles, siendo el tamaño de dichos lotes variable en función de cada planta. Realizar una buena agrupación en lotes es a la vez tan complejo como importante, ya que los beneficios finales dependerán de cómo se hayan hecho estos lotes. Este proceso de agrupación en lotes es el problema principal a resolver en esta tesis. En el apartado 2.3 se describe con detalle qué reglas han de tenerse en cuenta a la hora de juntar productos, y qué características de los mismos se consideran para determinar la homogeneidad. Al finalizar la agrupación de todos los productos se sacan a subasta todos los lotes junto con unos cuantos productos individuales que no se han juntado con ningún otro, ya sea por tamaño del lote o por falta de similitud entre ellos.
- 2) Subasta de los productos: todos los productos salen a subasta el mismo día de la semana a una hora determinada. El mecanismo de subasta que la empresa utiliza es

subasta cerrada de primer precio con precio reserva (en el capítulo 3 se analiza este método en detalle). Durante un período determinado los posibles compradores pueden realizar pujas sobre cualquier lote (subasta multi-unidad), hasta que se cumpla el plazo de finalización y sin conocer las pujas del resto de competidores.

- Asignación de los productos a ganadores: tras acabar el plazo de pujas, el ganador de cada lote es aquel que haya realizado la puja más alta. Este valor de puja se compara con el precio reserva que la empresa establece para cada lote: si la puja supera al precio reserva se adjudica y se envía el lote al ganador; en el caso que la puja ganadora no supere el valor mínimo establecido por la empresa, el lote no se adjudica al mayor pujador y los productos volverán a aparecer en la subasta de la semana siguiente.

La Figura 4 resume el proceso de manera esquemática:

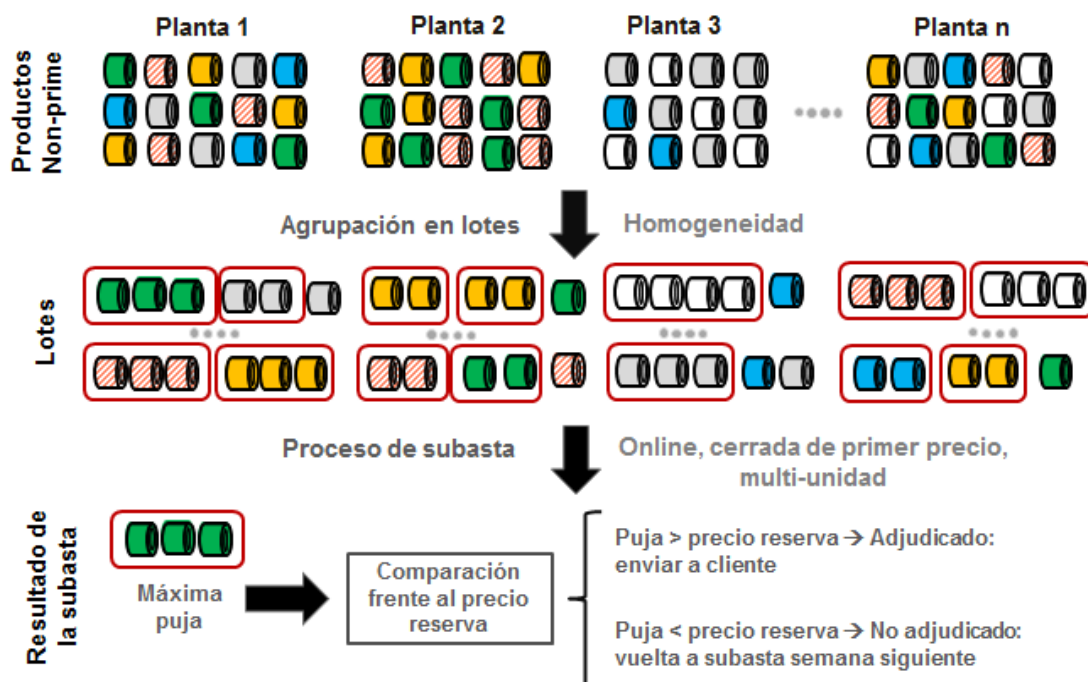


Figura 4. Esquema general del proceso de subastas de productos non-prime

Se estima que homogeneidad del lote repercute sobre el número de pujas del mismo, con lo que si se consiguen hacer mejores lotes éstos serán más atractivos para los pujadores y como consecuencia se conseguirá incrementar el beneficio total de la subasta.

2.2.1.1 Tiempo de preparación de la subasta

Un aspecto muy importante a resaltar es el tiempo disponible para preparar los productos que serán subastados online. La persona encargada dispone de una mañana para generar los lotes y subirlos a la página web para que los clientes realicen sus pujas. Esto quiere decir que en un tiempo aproximado de 4 horas se deben de realizar las siguientes tareas:

- Actualizar los datos de productos non-prime de cada planta
- Comprobar y actualizar las reglas de generaciones de lotes, ya que en ocasiones algunas de estas reglas pueden cambiar, por ejemplo el tamaño máximo de lote en alguna planta específica.
- Generar los lotes correspondientes a cada una de ellas en función de una serie de reglas que se explicarán en el siguiente apartado, tratando de que cada lote sea lo más homogéneo posible
- Introducir todos los lotes resultantes en la aplicación web para que éstos estén disponibles a la hora que comienza la subasta.

La consecuencia principal es que el algoritmo a desarrollar para la generación óptima de lotes ha de ser capaz de procesar la información y generar resultados en un período de tiempo muy corto, para que se puedan realizar diferentes optimizaciones si es necesario variando algunos parámetros del algoritmo.

2.3 Reglas de agrupamiento y homogeneidad en los lotes

Cada producto que se va a agrupar en lotes previamente a la subasta tiene unas características que lo definen. Algunas de estas características se utilizan como reglas que deben de cumplirse dentro de cada lote (por citar un ejemplo, todos los productos de un lote han de ser del mismo tipo: bobina, chapa...), y otras de ellas se utilizan para tratar que los productos dentro de un lote sean lo más homogéneos posibles. Las características que definen un producto son: planta, localización, forma, categoría, grado de acero, familia, subfamilia, número de lados de recubrimiento, aceitado, peso, ancho, espesor y espesor del recubrimiento. Se explican a continuación cada una de ellas, diferenciándolas en dos grupos: primero aquellas características que deben ser comunes dentro de los elementos de un lote, y después las características que se utilizan para definir la homogeneidad.

2.3.1 Características comunes a todos los elementos de un lote

Los valores de los siguientes parámetros de un producto deben de ser idénticos para todos los elementos de un lote: planta, localización, forma, categoría, familia y número de lados de recubrimiento. A continuación se explica el significado de cada parámetro:

- **Planta:** este parámetro especifica la planta origen del producto. Todos los elementos que se agrupan en un lote deben tener este parámetro igual, debido a razones logísticas, para facilitar el envío de los productos a su pujador ganador final. Normalmente el tamaño de los lotes coincide con la capacidad máxima de envío del medio de transporte. Para cada planta existe un tamaño mínimo que cada lote debe de tener, y un tamaño máximo definido por la restricción logística del tamaño del vehículo de transporte.
- **Localización:** puede ser que productos que se fabrican en la misma planta estén almacenados en sitios distintos situados a varios kilómetros. Al igual que con el concepto anterior, todos los elementos de un lote han de tener mismo parámetro “localización” debido a razones logísticas.

- Forma: especifica el tipo de producto. Valores típicos de este parámetro son: bobina, bobina recortada o chapa. Los clientes normalmente sólo necesitan un tipo de producto, de ahí que al agrupar en lotes todos los productos del mismo tengan que tener la misma forma.
- Categoría: en función del grado de defecto que tengan los productos se clasifican en varias categorías. Las categorías existentes son las siguientes, ordenadas de mayor calidad (menos defectos) a menor: categoría 11, categoría 1, categoría 2, categoría 3 y categoría 4. Éste campo también debe ser común a todos los elementos que compongan un lote.
- Familia: la familia de un producto especifica el tipo del mismo, desde el punto de vista del proceso siderúrgico. Ejemplos de familia son bobina caliente, bobina decapada, electrogalvanizada o pintada. De igual manera que sucede con la forma, un tipo de cliente normalmente sólo consume productos de un tipo de familia, con lo que es otra característica que debe ser igual dentro de un lote.
- Número de lados de recubrimiento: para los productos que llevan algún tipo de recubrimiento (por ejemplo bobina galvanizada que lleva recubrimiento de zinc, o bobina pintada) es necesario saber si ése recubrimiento se ha aplicado sobre ambos lados o solamente por uno. A la hora de agrupar dentro de un lote se deben de juntar productos que tengan el mismo número de lados de recubrimiento (0: sin recubrimiento, 1: recubrimiento por un lado, 2: recubrimiento por ambos lados).

2.3.2 Características consideradas para la homogeneidad de un lote

Las características que deben de ser comunes en un lote pueden utilizarse para hacer una división inicial de productos que garantiza que los productos son similares desde un punto de vista de negocio lógico, y se pueden mezclar. Sin embargo, a la hora de conformar los lotes que se sacarán a subastas es necesario realizar las agrupaciones de la manera más homogénea posible. Para analizar precisamente cómo de homogéneo es un lote en relación a los productos que lo componen se utilizan el resto de características: subfamilia, grado de acero, aceitado,

peso, ancho, espesor y espesor de recubrimiento. Es posible juntar en un lote productos con diferentes valores para cada uno de estos parámetros (por ejemplo bobinas de diferente peso, de diferente ancho, o con diferentes grados de acero), siendo necesario intentar que el lote sea lo más homogéneo posible, considerando a la vez todos los parámetros. Se explican a continuación cada una de estas propiedades:

- Subfamilia: dentro de una familia de productos existen varias divisiones en subfamilias, dependiendo de las características finales que se hayan aplicado. Éste parámetro es un caso peculiar, ya que existen unas subfamilias para las cuales sólo se pueden agrupar productos similares en el mismo lote, y otras en las que se ha de intentar favorecer juntar productos de ese tipo, pero sin ser obligatoriamente los mismos. Los grupos de subfamilias donde productos de diferente tipo pueden agruparse son:
 - “OCR” , “OHR” , “OAS” , “OAZ” , “OZ” , “OZA” , “OZE” , “OZO”
 - A su vez, dentro de éstas, se ha de dar prioridad a intentar juntar productos en estos dos grupos:
 - “OCR” con “OHR”
 - “OAS” con “OAZ” o “OZ” o “OZA” o “OZE” o “OZO”
 - “HD-Z” , “HD-ZM” , “HD-ZMO” , “HD-ZA”

Para el resto de subfamilias, todos los productos de un lote han de tener necesariamente el mismo valor. Esto añade una complejidad al problema, ya que nos encontramos antes dos situaciones distintas (permitir mezclar diferentes subfamilias en lotes u obligar a que sea necesariamente la misma), que tendrá que tenerse en cuenta a la hora de desarrollar el algoritmo de optimización.

- Grado de acero: el grado de acero de un producto especifica las propiedades y composición química del mismo. Se pueden juntar diferentes grados de acero pero se ha de dar prioridad a juntar algunos grados según los requisitos que se muestran a

continuación. Esta clasificación la han realizado los expertos de las plantas y definen qué grados son más parecidos para cada tipo de familia:

Familia HR			
Grupo	Grados		
Grupo1	TYPE_DD11	TYPE_DD11ULC	TYPE_S185
Grupo2	TYPE_S235	TYPE_S275	
Grupo3	TYPE_HSS460	TYPE_HSS500	
Grupo4	TYPE_HSS660	TYPE_HSS_GT_600	
Grupo5	HIGH_CARBON_TYPE_C45_C60	HIGH_CARBON_TYPE_C25_C40	
Grupo6	TYPE_HSS_550	TYPE_HSS600	
Grupo7	TYPE_E295	TYPE_E335	TYPE_E360

Figura 5. Similitud de grados de acero – Familia HR

Familia CR				
Grupo	Grados			
Grupo1	TYPE_DC01	TYPE_HC220	TYPE_HC260	TYPE_HC260LA
Grupo2	TYPE_DC03	TYPE_DC04	TYPE_DC05_06	
Grupo3	TYPE_HC300	TYPE_HC300LA	TYPE_HC340LA	TYPE_HC380LA TYPE_HC420LA TYPE_DP450 TYPE_DP500
Grupo4	TYPE_DP600	TYPE_DP780	TYPE_DP980	BORON_STEEL

Figura 6. Similitud de grados de acero – Familia CR

Familia EG				
Grupo	Grades			
Grupo1	TYPE_DC01	TYPE_HC220	TYPE_HC260	TYPE_HC260LA
Grupo2	TYPE_DC03	TYPE_DC04	TYPE_DC05_06	
Grupo2	TYPE_HC300	TYPE_HC300I	TYPE_HC340I	TYPE_HC380I
Grupo4	TYPE_DP600	TYPE_DP780	TYPE_DP980	BORON_STEEL

Figura 7. Similitud de grados de acero – Familia EG

Familia HD							
Grupo	Grados						
Grupo1	TYPE_DX51	TYPE_DX52	TYPE_S220GD				
Grupo2	TYPE_DX53	TYPE_DX54	TYPE_DX56				
Grupo3	TYPE_S350GD	TYPE_S390GD	TYPE_HCT450X				
Grupo4	TYPE_S450GD	TYPE_S550GD	TYPE_HCT490X	TYPE_HCT590X	TYPE_HCT780X	TYPE_HCT980X	TYPE_HCT780/980X TYPE_HX700LAD

Figura 8. Similitud de grados de acero – Familia HD

Familia OC							
Grupo	Grados						
Grupo1	TYPE_DX51	TYPE_DX52	TYPE_DX53	TYPE_DX54	TYPE_DX56	TYPE_S220GD	TYPE_S350GD
Grupo2	TYPE_S390GD	TYPE_S450GD	TYPE_S550GD				

Figura 9. Similitud de grados de acero – Familia OC

- Aceitado: especifica si el producto lleva una capa de aceitado o no. Se pueden mezclar productos aceitados con productos no aceitados, lógicamente siendo más interesante juntar productos de las mismas características (que todos los productos de un lote lleven aceitado o que todos los productos del lote no lo lleven).
- Peso: especificado en toneladas.
- Ancho: especificado en milímetros. Los compradores se suelen mover en unos rangos de ancho de utilización para fabricar sus productos, por lo que se permite juntar bobinas de diferente ancho, intentando minimizar la diferencia.
- Espesor: especificado en milímetros. Al igual que con el ancho, por las mismas razones se permite juntar bobinas de diferente espesor, tratando de que la diferencia entre las bobinas del lote sea mínima.
- Espesor de recubrimiento: en algunas ocasiones viene especificado en micrómetros(μm), y otras veces en gramos por metro cuadrado (g/m^2), por lo que es necesario reconocer y convertir a una unidad común las medidas para poder comparar varios productos.

2.4 Requisitos del algoritmo

Una vez analizado el problema, se pueden definir los criterios y funcionalidades que debe de cumplir el algoritmo de optimización. Los requisitos del mismo son los siguientes:

- Agrupar todos los productos de las diferentes plantas en lotes, teniendo en cuenta las características obligatorias que deben ser comunes en cada lote: planta, localización, forma, categoría, familia y subfamilia (solamente para algunos casos concretos).
- Minimizar el número de lotes, intentado que cada lote esté lo más cercano posible a su límite superior de capacidad, y cumpliendo a la vez la restricción de tamaño mínimo de lote (toneladas mínimas para que el lote sea subastable).
- Minimizar el número de productos que se quedan sin incluir en lotes.
- Realizar lotes lo más homogéneos posible, utilizando como parámetros para medir la homogeneidad la subfamilia, grado de acero, aceitado, peso, ancho, espesor y espesor de recubrimiento.
- Tiempo de ejecución: como se ha comentado anteriormente, el tiempo disponible para crear los lotes de todas las plantas es menor de 1 mañana, y es probable que en ocasiones la persona encargada de crear los lotes realice varias ejecuciones del mismo cambiando parámetros y comprobando los resultados. El número de productos que el algoritmo debe de ser capaz de optimizar en 1 ejecución puede rondar los 3.000 elementos distribuidos en unas 20 plantas. Es indispensable que el algoritmo permita realizar los lotes muy rápidamente, a pesar de la complejidad del mismo y del número de elementos involucrados. El tiempo máximo acordado por ejecución es de 5 minutos, lo cual deja aproximadamente unos 10 segundos como máximo por planta.
- Desarrollar un método de comparativa de productos y que permita al algoritmo de manera rápida poder determinar la homogeneidad de los mismos.
- Permitir dar más importancia a unos parámetros que a otros: para determinados productos o clientes, dependiendo del tipo de uso final del producto puede que sea

más importante que los productos dentro de un lote se parezcan mucho en espesor, ancho o cualquier otra de las características. Es necesario que el algoritmo sea capaz de adaptarse dinámicamente y de permitir configurar este tipo de importancia relativa de las características.

3 Análisis de subastas

En este capítulo se realiza un análisis de los principales trabajos existentes en la literatura referentes al mundo de las subastas. La literatura en este tema es muy amplia, con lo que el análisis se ha centrado exclusivamente en las variantes de subasta objetivo de optimizar en esta tesis: cerrada de primer precio, online y de productos por lotes. El objetivo es analizar las cualidades y factores importantes de estas 3 modalidades de subasta, para poder concluir si el método utilizado actualmente es el idóneo, o si no, en caso de que la empresa estuviera abierta en un futuro, dar una recomendación sobre la variedad de subasta a utilizar. En el caso de plantearse como posible un cambio en el sistema, el estudio serviría para proponer mejoras al sistema actual.

Al principio del capítulo se realiza una introducción explicando las principales variantes de subasta más utilizadas y las características que las definen. De cara a analizar los diferentes factores que se tienen en cuenta en el problema, se realizan varios sub-apartados: estudio del rendimiento teórico de los diferentes tipos de subastas, principales factores humanos influyentes (aversión al riesgo, arrepentimiento y colusión entre pujadores), factores a considerar en las subastas on-line (mecanismo de subasta, restricciones a aplicar sobre las pujas, duración de la misma) y consideraciones generales sobre fraude y reputación en las subastas. Así mismo, se desarrolla un resumen de los principales análisis sobre subastas multi-unidad, con el objetivo de analizar la idoneidad o no de agrupar productos en lotes, basándose en los diferentes estudios referentes al tema.

Para finalizar, se incluye un apartado de conclusiones obtenidas y se proponen recomendaciones sobre el modelo de subastas a utilizar para la venta de productos en el sector del acero.

3.1 Principales variantes

Una subasta es un mecanismo de mercado que cuenta con un conjunto explícito de reglas mediante la que se determinan la asignación de recursos y sus precios, en función de las pujas (valoraciones económicas del bien subastado) presentadas por los participantes. Es una manera muy simple y práctica de determinación de precios en un comercio unilateral.

Existen varias características que definen una subasta, con lo que también existen diferentes clasificaciones de subastas en función de diferentes términos como indican (Parsons, et al., 2011). Así, se pueden distinguir las siguientes:

- **Directa o inversa:** en función de que la subasta se enfoque desde el punto de vista del comprador o del vendedor podemos distinguir 2 tipos de subastas. En una subasta directa los compradores pujan por productos de los vendedores (son las más comunes), mientras que en el caso de las subastas inversas los posibles vendedores compiten para vender los bienes o servicios a los compradores.
- **Unidireccional o bidireccional:** en las subastas unidireccionales la subasta sólo se realiza en uno de los canales presentados anteriormente (o pujan los compradores o pujan los vendedores), mientras que en el caso de subastas bidireccionales tanto posibles compradores como vendedores pujan al mismo tiempo.
- **Abiertas (open-cry) o de ofertas selladas (sealed-bid):** en las subastas abiertas todos los pujadores conocen las pujas de los demás, mientras que en las de ofertas selladas los participantes envían las ofertas durante cierto período de tiempo durante el cual las pujas se mantienen en secreto. Al final del tiempo establecido de subasta las diferentes pujas son evaluadas con respecto al criterio de adjudicación para decidir el ganador. A su vez, dentro de las ofertas selladas se distinguen:
 - Una etapa (*single round sealed-bid*): los posibles compradores envían una única puja por el producto.

- Múltiples etapas (multi-round sealed-bid): se pueden realizar pujas durante un período definido. Al final de cada período se publican las ofertas recibidas y se abre un nuevo período de ofertas.
- **Unitarias o multi-unidad:** en la unitarias los pujadores solamente pueden pujar por un ítem en cada subasta, mientras que en las multi-unidad se permite a los pujadores pujar por más de un bien a la vez. Las subastas multi-unidad son frecuentes sobre todo en internet debido a la disponibilidad de procesar la información automáticamente.

Todas estas variantes se combinan entre sí y definen el conjunto de reglas finales que se aplican en cada caso. Las 4 variantes más comunes utilizadas son las siguientes (Zhang, et al., 2013):

- **Inglesa:** es una subasta con pujas ascendentes. El ítem a subastar suele empezar con un precio bajo, y los posibles compradores realizan sus pujas secuencialmente o simultáneamente, hasta que no se reciben más pujas. Cada pujador puede “oír” la oferta enviada por un pujador rival y tiene un tiempo determinado para decidir si pujar más alto. Normalmente estas subastas incluyen un precio reserva, o precio mínimo que el comprador define para que se acepte la subasta ganadora. En las subastas físicas las ofertas suelen ser recibidas en el orden de varios segundos, mientras que en las subastas por internet estos períodos de puja pueden ser de minutos u horas.
- **Holandesa:** una subasta holandesa es una subasta de precio descendente. El vendedor establece un precio alto por un ítem determinado, el precio se va decreciendo a medida que transcurre el tiempo. En el momento en que un pujador acepta el precio haciendo una puja la subasta termina y el pujador se lleva el artículo por el precio determinado. Es frecuente utilizar este tipo de subastas en productos perecederos, o también en la venta de pasajes de avión.
- **Subasta cerrada de primer precio (first-price sealed bid):** en este tipo de subastas se requiere a los posibles compradores que envíen sus ofertas durante un cierto período de tiempo. Las ofertas se guardan y se evalúan una vez terminado el periodo de puja

para determinar el ganador. La oferta con la puja más alta es la que gana el producto, y la cantidad a pagar es la puja ofertada.

- **Subasta cerrada de segundo precio:** también conocida como subasta del virrey (vickrey auction). El funcionamiento es igual que en la subasta de primer precio, y el ganador es también el pujador que realiza la puja más alta. Sin embargo, en este tipo de subastas el precio que el ganador ha de pagar por el producto es la segunda puja más alta.

Estos 4 tipos suelen ser los más habituales, aunque otras variantes utilizadas son:

- **Anglo-holandesa:** es una subasta mezcla de inglesa y holandesa. Inicialmente los compradores realizan pujas ascendentes como en una subasta inglesa (paso 1 de la subasta), hasta que sólo quedan 2 pujadores finalistas. Cuando solamente quedan dos pujadores, realizan una subasta cerrada de primer precio (paso 2 de la subasta), que es equivalente a una subasta holandesa de dos personas (Krishna, 2009). El precio inicial del segundo paso es mayor que el precio pujado por el último pujador durante la fase inicial de subasta inglesa. Este mecanismo ha sido empleado por ejemplo para conceder licencias de operación en el espectro radioeléctrico.
- **Vickey-Clarke-Groves:** este tipo de subastas es una variante de la subasta cerrada de segundo precio, orientadas a la subasta de un conjunto de M artículos. Cada pujador realiza una puja por un número de artículos $n < M$. Todas las pujas se computan y cada ítem se asigna a un pujador intentando maximizar los beneficios. El sistema asigna los artículos de un manera socialmente óptima: carga a cada pujador individual el “daño” que ha causado a otros pujadores. El objetivo es que cada pujador realice su propia valoración realista del artículo.
- **Esperar cola:** en este tipo de subastas no hay un proceso explícito de puja. En su lugar, los compradores se apuntan en una cola, y el vendedor ofrece los artículos secuencialmente a los compradores en un tiempo desconocido. Aquellos compradores que están primero en la lista se llevan los mejores productos aunque para ello han tenido que estar más tiempo esperando.

- **Combinatoria:** en las subastas combinatorias los pujadores realizan ofertas sobre varios productos, y especifican a la vez un conjunto de ítems sin alguno de los cuales no comprará el resto. Todas las pujas han de tenerse en cuenta, así como los requisitos mínimos de cada pujador para decidir el reparto final de los productos, aceptando o rechazando el conjunto total de ítems para cada pujador.
- **Cooperativa:** las subastas cooperativas son relativamente recientes, debido a la emergencia de las subastas por internet. Los vendedores ofrecen descuentos de precio en función del número de ítems que se compran. Los compradores cooperativamente se unen para realizar una puja conjunta intentando acceder de esta manera a los descuentos.

En la Tabla 2 se han resumido las principales subastas utilizadas, y las características de cada una:

Método de subasta	Puja	Privacidad	Ganador	Precio a pagar
Inglesa	Ascendente	Abierta	Puja más alta	Precio pujado
Holandesa	Descendente	Abierta	Primera puja	Precio pujado
Cerrada de primer precio	Durante tiempo determinado	Cerrada	Puja más alta	Precio pujado
Cerrada de segundo precio	Durante tiempo determinado	Cerrada	Puja más alta	2ª puja más alta
Anglo-Holandesa	Ascendente primero / descendente después	Abierta	Primera puja en fase descendente	Precio pujado
Vickrey-Clarke-Groves	Durante tiempo determinado	Cerrada	Puja más alta	2ª puja más alta Penalización por el sobrecoste al pujador mayor

Tabla 2. Variantes de subastas más utilizadas, con sus principales características (elaboración propia)

En el siguiente apartado se analiza el rendimiento de los tipos de subastas más usuales, a partir de los trabajos existentes en la literatura.

3.2 Rendimiento

En este apartado se analizan los estudios realizados sobre el rendimiento de los 4 tipos más comunes de subastas (inglesa, holandesa y cerrada de primer y de segundo precio), mostrando por un lado estudios de comparativa de dichos tipos, y por otro estudios concretos sobre la subasta cerrada de primer precio, que es la que se utiliza actualmente para la venta de productos de acero.

Ya en 1961 se postula el principio de equivalencia de ingresos (Vickrey, 1961), el cual postula que tanto la subasta inglesa, holandesa, de primer precio y de segundo precio proporcionan el mismo valor esperado al vendedor, siempre y cuando se cumplan las siguientes condiciones:

- Los pujadores son neutrales ante el riesgo: tener que realizar un pago igual a cero después de participar en la subasta es igual que tener que pagar cero sin haber realizado ninguna puja.
- Cada pujador conoce precisamente cuánto vale para él el artículo por el que puja. Cada valoración entre pujadores es independiente (los pujadores no se confabulan) y las distribuciones de estas valoraciones son conocidas.
- Los pujadores son simétricos en el sentido de que las suposiciones sobre otros pujadores y el vendedor son comunes para todos los pujadores.
- El pago final depende solamente de las pujas.

En el caso de que estas condiciones no se cumplan, existen diferentes estudios para determinar si las subastas de tipo cerrado son mejores que las de precio abierto (la mayoría de comparativas se realizan entre las de primer precio y la inglesa). En términos generales la literatura demuestra que las subastas cerradas dominan a las de precios abiertos en los siguientes escenarios: 1) cuando los compradores son sensibles al riesgo (Matthews, 1987) y 2) cuando hay posibilidad de que los compradores confabulen (McAfee & McMillan, 1987):

- Cuando los pujadores son sensibles al riesgo, las subastas cerradas de primer precio tienen mayor valor esperado porque un pujador con miedo al riesgo será conducido por el miedo a perder la apuesta con lo que realizará pujas más altas para asegurar llevarse el producto. Por el contrario, en las subastas abiertas no se produce este efecto (Maskin & Riley, 2000) ya que los pujadores seguirán en la subasta mientras que el precio del artículo esté por debajo de su valoración.
- Para el segundo supuesto, cuando hay posibilidad de que los pujadores colaboren entre ellos, McAfee y McMillan afirman que el beneficio esperado será mayor en las subastas de precio cerrado. Otros estudios muestran que las subastas abiertas facilitan la colaboración ya que los pujadores pueden ver directamente el comportamiento de los otros (Graham & Marshall, 1987) y que las subastas de precio cerrado con precio reserva es el mejor método de defensa para evitar estos comportamientos en los pujadores (Robinson & others, 1983).

Existen más estudios que muestran que la subasta de primer precio genera mayores beneficios que la inglesa tanto desde el punto de vista teórico (McAfee & McMillan, 1987) como desde el punto de vista de experimentos en laboratorio (Kagel & Richard, 2001). Ateniéndose a estudios reales, Miller realiza una comparativa utilizando datos reales en una subasta de tasas de embargo en Illinois, llegando a la conclusión de que utilizando subasta cerrada de primer precio se consiguen beneficios un 22% mayores que con una subasta inglesa (Miller, 2014).

Sin embargo, bajo otras condiciones, también se han encontrado casos que demuestran que no siempre la subasta cerrada de primer precio es la más ventajosa. Así, Che realiza un estudio y afirma que las subastas donde todos los pujadores han de pagar una “multa” en caso de que no se lleven el artículo son más beneficiosas que las de primer precio a sobre cerrado cuando los pujadores tienen restricciones económicas (Che & Gale, 1996). A pesar de este resultado, aplicar este tipo de subastas a un caso real es muy complicado y no podría realizarse en un mercado competitivo, ya que los pujadores que pierden se cansarían de pagar dinero por nada y a la larga acabaría influyendo negativamente. Otro ejemplo es el análisis de Chow, que realiza una comparativa en un caso concreto de subasta de fincas (Chow & Ooi, 2014), llegando a la conclusión de que para ese caso concreto la subasta inglesa ofrece

unos beneficios de un 4% mayores para el vendedor que la subasta cerrada de primer precio. Está claro que a pesar de los estudios teóricos cada mercado y cada caso es independiente y debe ser analizado de manera distinta.

Otro aspecto a destacar bajo el marco del rendimiento de las subasta de primer precio es la denominada “maldición del ganador”. En una subasta de valor común, los artículos que se subastan son de un valor equivalente para todos los pujadores, pero éstos no saben el valor del artículo en el mercado cuando realizan sus pujas. Cada pujador estima el valor del artículo independientemente antes de pujar. Como el ganador de la subasta es aquel que realiza la puja más alta, el ganador es el que hace la mayor estimación de mercado. Asumiendo que la media de las pujas es precisa con respecto al valor real del objeto, el ganador de la subasta habrá sobreestimado el valor del objeto, con lo que el ganador al final tiende a pagar más del valor real de mercado. Haile realiza un conjunto de test paramétricos que permiten realizar una medición de la severidad de la maldición del ganador (Haile, et al., 2003). Bulow y Klemperer muestran cómo el efecto de la maldición del ganador afecta a la lógica tradicional que opina que “a mayor oferta de productos menores precios”, o “a menor número de demandantes menores beneficios”. En su estudio muestra cómo las condiciones lógicas del mercado no siempre son aplicables en el mundo de las subastas, siendo afectadas en gran medida por la maldición del ganador y el comportamiento humano conocedor de este fenómeno (Bulow & Klemperer, 2002).

3.3 Factores humanos

En este apartado se muestran estudios realizados en la literatura para analizar cómo el comportamiento de las personas modifica la tendencia general de la subasta. En los siguientes apartados se analizan los principales trabajos hasta la fecha, clasificados en función de los principales factores humanos que influyen sobre el devenir de la subasta: la aversión al riesgo, el arrepentimiento de los pujadores y la colusión entre los mismos. Dentro de este apartado se incluye también un análisis sobre cómo controlar el fraude y la importancia de la reputación en las subastas online.

3.3.1 Aversión al riesgo

En términos financieros, la aversión al riesgo es la preferencia de un inversor (participante en la subasta) por evitar incertidumbre en sus inversiones (pujas). Por lo general cualquier persona inversora (aplicable también a participantes en subastas) tendrá un comportamiento racional que les lleva a desear el mayor beneficio posible (asumiendo un cierto riesgo), pero a la vez son por naturaleza enemigos del riesgo y evitarán en la medida de sus posibilidades tener que asumirlo. Los pujadores reaccionan significativamente de manera más frecuente a un impulso si éste es cuantitativamente claro en términos de recompensa de antemano (Neugebauer & Selten, 2006). El ejemplo típico de aversión al riesgo es la posibilidad de elegir entre 2 tipos de lotería: en la primera existe una probabilidad del 50% de ganar 100 euros, y una probabilidad del 50% de ganar 0 euros; mientras que en la segunda lotería se obtienen 50 euros con 100% de probabilidad. Ante esto, se distinguen:

- Una persona aversa al riesgo se quedará con la lotería donde gana 50 euros.
- Una persona neutral al riesgo es indiferente entre elegir una lotería u otra.
- Una persona amante del riesgo se quedará con la lotería no segura.

El efecto del grado de aversión al riesgo en los pujadores es tan importante que, como comentábamos anteriormente, mientras que en subastas con pujadores neutrales al riesgo la ganancia estimada es igual para cualquiera de los tipos principales de subastas, en el caso que los pujadores sean sensibles al riesgo realizan pujas mayores para llevarse los artículos que las

que realizarían en equilibrio cuando el riesgo no sea un factor a considerar; con lo cual las subastas de primer precio tienen un valor esperado mayor (Maskin & Riley, 1984).

En (Marechal & Morand, 2011) analiza el impacto que representan los diferentes niveles de aversión al riesgo sobre los márgenes de ganancia de los pujadores y sobre la eficiencia de reparto de bienes y sobre los beneficios esperados de la subasta, considerando una subasta de primer precio con 2 pujadores. Concluyen que los beneficios esperados de la subasta se incrementan a medida que los pujadores tienen actitud más diferente entre ellos frente al riesgo, y que la eficiencia del reparto de la subasta disminuye a medida que aumenta la aversión al riesgo entre ambos. En (Hu, et al., 2010) se analizan los efectos de la aversión al riesgo del comprador y vendedor tanto en subastas de primer precio como de segundo, asumiendo valores privados independientes de pujadores homogéneos, permitiendo al vendedor establecer un precio de reserva óptimo. Tanto en el caso de primer como de segundo precio, el precio óptimo de reserva se reduce cuando los vendedores están influenciados por el riesgo. Así, a mayor aversión al riesgo de los vendedores los autores muestran una mayor eficiencia de la subasta. Desde el punto de vista de los pujadores por tanto, el efecto del riesgo en el vendedor resulta beneficioso.

3.3.2 Arrepentimiento de los pujadores

Otro factor importante, cuyo estudio ha tenido gran auge en los últimos años es el del arrepentimiento de los pujadores. La idea fue propuesta inicialmente en (Engelbrecht-Wiggans & Katok, 2009), basándose en los trabajos previos de Bell sobre el arrepentimiento (Bell, 1983), y la decepción (Bell, 1985). Esta idea se basa en el concepto de que los pujadores no solamente están motivados por los beneficios esperados, sino también por dos tipos de arrepentimiento: del ganador y del perdedor. El arrepentimiento del ganador ocurre cuando el ganador paga más que la segunda oferta más alta (este caso aplicaría en las subastas de segundo precio), y el arrepentimiento del perdedor ocurre cuando un pujador perdedor pierde una buena oportunidad debido a haber pujado demasiado bajo. Las pujas óptimas se incrementan a medida que la sensibilidad al arrepentimiento del perdedor se incrementa, y decrecen si la sensibilidad hacia el arrepentimiento del ganador se incrementa. En el mismo artículo, los autores realizan otro estudio para comprobar qué efecto de los dos comentados hasta ahora (aversión al riesgo y arrepentimiento) tiene más influencia sobre el por qué los

pujadores tienden a realizar pujas más altas en las subastas cerradas de primer precio, llegando a la conclusión de que el factor arrepentimiento tiene más influencia a la hora de realizar pujas mayores. A su vez se demuestra también cómo el comportamiento de pujadores en subastas concretas analizadas en la literatura que no encajaban dentro del comportamiento de aversión al riesgo se pueden explicar con el concepto de arrepentimiento, como se muestra en (Kagel & Levin, 1993), (Cason, 1995) y (Isaac & James, 2000).

3.3.3 Colusión

Para finalizar el estudio desde el punto de vista de comportamiento humano, es necesario al menos comentar un resumen de los diferentes análisis sobre la colusión de los pujadores. El término colusión se refiere a cualquier acto que acuerdan o más personas con el fin de perjudicar a un tercero. A pesar de que es una actividad ilegal en las subastas, es un problema evidente como se muestra en (Asker, 2010) y (Harrington, 2008).

En (Robinson, 1985) se realiza uno de los primeros estudios de colusión, considerando diferentes formatos de subastas, llegando a la conclusión de que las subastas de primer precio son las que tienen una susceptibilidad relativa menor de colusión entre pujadores. Marshall and Meurer proponen recomendaciones para específicamente evitar la colusión: tener identificados a los pujadores y ser pujadores con gran bagaje de pujas (Marshall & Meurer, 2004). De nuevo refuerzan la idea de que en las subastas cerradas es difícil mantener la colusión entre pujadores, y establecen que para que la colaboración sea fructífera los pujadores deben planear participar en las subastas de determinados productos y organizaciones por un tiempo en el futuro; así mismo, la subasta de los artículos debe repetirse regularmente. También existen otra serie de estudios que proponen métodos para detectar la colusión, basándose en el número de pujas y en la categoría de las mismas, como son (Porter & Zona, 1992) y (Asker, 2010).

Todos estos estudios muestran cómo al final, el comportamiento humano y la situación de mercado influye sobre cada apuesta. El mismo artículo tiene un valor relativo diferente para cada pujador, y en función de sus características sociales (grado de aversión al riesgo, arrepentimiento), de la ganancia esperada y de las condiciones del mercado en un momento determinado cada pujador realizará la puja que estime conveniente.

3.3.4 Fraude y reputación

El carácter anónimo de una subasta online crea también oportunidades para el fraude. Como se ha analizado anteriormente, la teoría tradicional de subastas ha discutido el efecto del diseño de la subasta sobre las oportunidades de colusión entre pujadores, pero Internet ofrece mayores oportunidades para el comportamiento fraudulento. Específicamente, en las subastas C2C online, los ganadores puede que no paguen, y en algunos casos, no están ligados por ningún contrato a pagar al vendedor. De la misma forma, el vendedor puede aceptar pagos sin proporcionar luego el producto, defraudando así al pujador. Estos problemas han derivado en dos respuestas: diseñar mecanismos para prevenir y evitar fraude, y dar más valor a la reputación, como se sugiere en los estudios de (Wang & Zoltan Hidvegi, 2001) y (Yokoo, et al., 2002).

La posibilidad de expresar la opinión mediante sistemas de puntuación abre el terreno para analizar la importancia de la reputación en determinar los resultados de la subasta. En (Houser & Wooders, 2006), (McDonald & Slawson, 2002), (Dewan & Hsu, 2001) y (Ba & Pavlou, 2002) se realizan estudios empíricos sobre la relación entre la reputación del vendedor y el resultado de la subasta en Ebay, y concluyen que una mayor reputación del vendedor lleva a más y mejores pujas.

En (Dellarocas, 2000) se desarrolla un modelo matemático para analizar los factores de la confianza en el comercio online, subrayando los fallos en los sistemas actuales y mostrando posibles técnicas para mejorar el diseño. Proponen servicios como pre-calificación a los proveedores e investigar los bienes vendidos.

Para el caso a analizar en esta tesis, parece claro que es necesario un sistema de control sobre los compradores, llevando un seguimiento de los pagos realizados. Actualmente ya se realiza un control de este tipo, llegando incluso a vetar a un pujador debido a impagos en ventas anteriores.

3.4 Factores a considerar al diseñar una subasta online

En el caso a analizar en esta tesis, la subasta es online. El efecto de internet en las subastas ha sido analizado en (Klein & O'Keefe, 1999), destacando como efectos clave: 1) reducción de costes de transacción tanto para el comprador como para el vendedor; 2) accesibilidad para más participantes (pujadores y vendedores): la participación es más sencilla, incrementando el número de pujadores potenciales, y a la vez incrementando la variedad de artículos que salen a subasta; 3) habilidad para dirigir subastas complejas y facilidad de descripción de productos, expandiendo el horizonte de dichos productos.

A estos efectos, Pinker añade otros dos: 1) facilidad de recolección de datos sobre la subasta: el uso de ordenadores hace posible coleccionar datos sobre la subasta y sobre los participantes, que puede utilizarse para diseñar mejores subastas en el futuro con mayor valor comercial; 2) posibilidad de los participantes de unirse en cualquier momento mientras la subasta permanezca abierta, aspecto que debe ser considerado por los diseñadores de las subastas online, ya que el comportamiento de los consumidores puede cambiar a medida que pasa el tiempo (Pinker, 2001).

A pesar de sus grandes beneficios, las subastas online también presentan problemas como que no hay contacto con el objeto subastado por parte del comprador previo a la subasta, a la vez que potenciales problemas de fraude: el ganador de la subasta debe confiar en que el vendedor le enviará el objeto. Una subasta bien diseñada debe ser robusta en referencia a las asunciones sobre el comportamiento de los pujadores, ser eficiente en la asignación de los productos, tener bajos precios de transacción y ser resistente al fraude. En (Pinker, et al., 2003) se especifica que los principales aspectos a considerar a la hora de diseñar una subasta online son el mecanismo, las restricciones sobre las pujas, y la duración de la misma. Estos 3 factores son analizados a continuación.

3.4.1 Elección del mecanismo de subasta

El impacto de las reglas de participación es determinado por un complejo conjunto de factores. Como se ha comentado anteriormente, los 4 tipos de mecanismos de subastas más

extendidos (y a la vez los más estudiados en la literatura) son la Inglesa, la cerrada de primer y segundo precio y la Holandesa (siendo la Inglesa la más utilizada).

Actualmente la subasta Inglesa es la dominante en Internet, probablemente porque usa un mecanismo que la gente encuentra familiar e intuitivo (Milgrom, 1989). Una de las desventajas de la subasta Inglesa es la naturaleza multi-ronda al ser los precios abiertos, con lo que en una subasta tradicional, se requiere la presencia física de los pujadores, lo que añade coste a la participación. Sin embargo en Internet este coste disminuye en gran medida, con lo que estas subastas se hacen todavía más atractivas. Otra de las desventajas de las subastas Inglesas es que tienen más facilidad para el amaño, debido a la importancia de este tema en la siguiente sección se analizará en detalle el problema del fraude.

La principal cuestión a analizar es si en la práctica en Internet las subastas funcionan como deben. En (Lucking-Reiley, 1999) se realizan experimentos con subastas a través de internet de tarjetas y cartas de coleccionista, comparando los beneficios de subasta Holandesa con los de subasta cerrada de primer precio, y lo mismo entre la subasta Inglesa y la cerrada de segundo precio, con el objetivo de comprobar si la equivalencia teórica de beneficios sin considerar actitudes hacia el riesgo entre estos tipos de subastas prevalece en Internet. Por un lado encuentra poca diferencia entre la subasta Inglesa y la cerrada de segundo precio, con lo que aseguran que el comportamiento se mantiene aunque la subasta sea a través de Internet. A su vez, los autores observan los precios conseguidos con la subasta Holandesa son mayores que con las cerradas de primer precio.

Hann and Terwiesch estudian el comportamiento de los pujadores online, concluyendo que, contrariamente a lo que se pudiera esperar, los pujadores no pujan frecuentemente con pequeños incrementos, sino unas pocas veces (menos de 4 normalmente) con incrementos significativos (Hann & Terwiesch, 2003). Los autores explican éste comportamiento como demostrador de que hay un coste de participación en la configuración de estas negociaciones que no hace que funcionen de manera óptima, no siendo posible por lo tanto establecer una configuración mejor que otra, sino que la mejor opción dependerá de la naturaleza del caso. Sin embargo, en (Roth, et al., 2002) se afirma que por regla general la subasta cerrada de segundo precio es la más rentable en las subastas online.

Ateniéndose a estos estudios, tanto la subasta holandesa como la cerrada de segundo precio presentan en principio un rendimiento al menos igual, y quizás superior a la subasta inglesa, siendo dependiente de cada caso y artículos subastados.

3.4.2 Restricciones sobre las pujas

En las subastas online, se utilizan frecuentemente varios parámetros para influenciar las pujas. Así, pueden existir restricciones sobre la puja mínima obligan a los participantes a pujar por encima de un determinado nivel, o precios reserva, que son aquellos precios mínimos que los vendedores aplican una vez se ha decidido el ganador de la subasta. Cuando se aplican precios reserva, si la subasta finaliza antes de alcanzar el precio reserva o ningún pujador llega a ese precio (para el caso de las subastas cerradas) el vendedor tiene derecho a retirar el artículo y no realizar la venta.

En (Vakrat & Seidmann, 2000) se analizan 350 subastas realizadas en SurplusAuction.com, para artículos disponibles en sitios de venta al por menor. Observan que, cuando no hay una puja mínima, los elementos son subastados con descuentos significativos sobre el precio de venta. A medida que aumenta el nivel de puja mínima, una mayor proporción de las subastas acaban con precios que están al nivel de puja mínima, con lo que el beneficio es mayor. Esto se cumple hasta cierto valor, ya que también a medida que aumenta el nivel de puja mínima aparecen muchas subastas sin pujas. Es necesario por tanto encontrar un buen nivel de puja mínima cuando se establece para conseguir maximizar los beneficios. Bajari y Hortacsu refuerzan la idea de que el precio de puja mínima afecta enormemente la participación (Bajari & Hortacsu, 2003), mientras que en (Lucking-Reiley, 1999) se muestra que las pujas mínimas incrementan el precio pero también disminuyen la probabilidad de una venta.

En (Kaiser & Kaiser, 2010) se recomienda que los vendedores que usan un precio reserva secreto usen a la vez un nivel de puja mínimo pequeño. La idea es que el nivel de puja mínimo pequeño atrae a muchos compradores, a la vez que el precio reserva evita que el artículo se venda por precio por debajo del esperado. Bapna et al. reafirman la teoría de que es necesario buscar un punto óptimo de equilibrio a la hora de establecer este tipo de restricciones sobre las pujas (Bapna, et al., 2000).

3.4.3 Duración

El último factor a analizar al considerar el diseño de una subasta online es la duración. Está claro que el tiempo juega un papel muy importante en las subastas por Internet. En contraste con las tradicionales subastas abiertas en persona, que duran minutos, las subastas por Internet pueden durar días o semanas. Debido a que los pujadores no tienen por qué estar presentes y pueden pujar desde prácticamente cualquier lugar en cualquier instante de tiempo, es práctico realizar subastas con mayor duración. Al final, la duración de una subasta online determina el número de pujadores, por lo tanto afecta a su resultado final y a los beneficios.

En Internet, una subasta comienza con un anuncio describiendo el producto, las reglas y el tiempo final, con la esperanza de que suficientes pujadores sean atraídos para generar un alto precio de venta. Si la subasta es demasiado corta y pocos pujadores participan, es más probable que el precio final sea bajo. (Lucking-Reiley, 1999) muestra evidencias en subastas de monedas en eBay de que mayor duración conlleva mayores precios. Sin embargo, esto no es cierto en todos los casos, ya que en ocasiones incrementar la duración de la subasta implica otra serie de costes: en el caso de productos con una rápida depreciación, como componentes electrónicos y de ordenadores, las subastas largas pueden reducir el valor del producto y por tanto los beneficios del vendedor.

En (Pinker, et al., 2003) se analiza el impacto de una subasta Inglesa online para artículos individuales, en el que los pujadores tienen valores privados independientes dados por una distribución uniforme $[v_l, v_h]$, y sin precio reserva. El precio de cierre de la subasta (p_a) es determinado por la segunda mayor puja. Asumiendo un número fijo n de pujadores, el precio esperado de la subasta es

$$1) \quad E[p_a] = v_l + (v_h - v_l)(n - 1)/(n + 1)$$

En una subasta de artículos individuales, éste es el beneficio de la compañía. Si el coste de los pujadores por unidad de tiempo de retraso por participar en la subasta es w , entonces en una subasta de duración T , los pujadores mejorarán las pujas por wT . El precio esperado de cierre entonces es el siguiente:

$$2) \quad E[p_a] = \begin{cases} v_l + (v_h - v_l) \frac{(n-1)}{(n+1)} - wT & \text{si } wT \leq v_l \\ (v_h - wT) \frac{(n-1)}{(n+1)} & \text{si } wT > v_l \end{cases}$$

De esta ecuación se pueden observar varias cosas. En primer lugar, el precio de cierre de la subasta se incrementa a medida que la función de distribución es más ancha ($v_h - v_l$). En segundo lugar, debido a que el factor $\frac{(n-1)}{(n+1)}$ se incrementa con n , el precio de cierre se incrementa con el número de pujadores, con lo que se demuestra también teóricamente mayor participación conlleva mayores beneficios. En tercer lugar, debido a los costes de retraso, el precio esperado disminuye a medida que aumenta T .

Existen 2 aproximaciones principales para manejar el tiempo límite de la subasta en las subastas online. La primera es tener un tiempo límite rígido: no se aceptan más pujas una vez pasado ese tiempo. Con la segunda aproximación la subasta continuará después de alcanzar el tiempo límite siempre y cuando haya habido pujas recientes, donde el término "reciente" normalmente quiere decir 5 o 10 minutos. El tiempo límite rígido ha dado lugar a un fenómeno en el que los pujadores realizan sus pujas en el momento final de la subasta (minutos o segundos) para que otros pujadores no tengan tiempo de sobrepujarlos. Bapna et al. llaman a este tipo de pujadores "oportunistas", y muestran que suelen ser ganadores en aproximadamente un 20% de las subastas observadas (Bapna, et al., 2000). En (Roth, et al., 2002) se realiza un análisis en el que demuestran que las pujas de último minuto presentan un comportamiento diferente, debido al tiempo límite fijo y la probabilidad de no ser capaces de realizar otra puja superior más tarde, concluyendo que en las subastas online no siempre se deben asumir las predicciones teóricas y hay que considerar todos los factores involucrados a la hora de elegir el mejor mecanismo.

3.5 Agrupaciones de productos en lotes para subastas

El propósito de este apartado es analizar si la literatura respalda el agrupamiento de productos previamente a la subasta, o sería mejor subastar los productos de manera individual. A pesar de que la venta agrupada de múltiples productos conlleva economías de agregación (Bakos & Brynjolfsson, 2000), puede resultar también en que el vendedor renuncie a los beneficios adicionales que podría obtener vendiendo varios artículos separadamente, cada uno al mejor pujador. Esto levanta importantes preguntas sobre cómo los vendedores deberían actuar para la venta de múltiples productos en las subastas y, en particular, qué factores determinan la estrategia óptima de venta en este contexto para conseguir mayores beneficios.

Un vendedor que subasta artículos individuales puede hacerlo simultáneamente o secuencialmente. Según los estudios de la literatura, las diferencias en los beneficios entre estas dos estrategias están influenciadas por el grado de complementariedad entre los artículos, como se afirma en (Hausch, 1986), (Milgrom, 2004) y (Rosenthal & Wang, 1996). Milgrom muestra que en ausencia de complementariedad entre los productos ofrecidos, los beneficios de las subastas secuenciales exceden aquellos de las subastas simultáneas ya que los pujadores pueden conseguir nueva información de las subastas anteriores y pujar más agresivamente en las sucesivas. De igual forma, en (Krishna & Rosenthal, 1996) se afirma que cuando hay complementariedad entre los artículos, las subastas simultáneas tienden a producir mayores beneficios. Por su parte, (Stremersch & Tellis, 2002) afirman que la asimetría de los productos tiene un efecto negativo sobre la rentabilidad de las subastas por lotes frente a subastas de componentes separados, lo cual refuerza el hecho de agrupar componentes similares. (Venkatesh & Kamakura, 2003) demuestran que mayor complementariedad (o menor posibilidad de sustitución) conlleva mayores valoraciones de los lotes.

Sólo un pequeño número de artículos han examinado la rentabilidad de subasta por lotes comparadas con subastas de componentes individuales: (Palfrey, 1983), (Krishna & Rosenthal, 1996), (Chakraborty, 1999), (Subramaniam & Venkatesh, 2009) y (Popkowski Leszczyc & Haubl, 2010). Todos estos autores utilizan una aproximación analítica, y no empírica. Las conclusiones de cada estudio se pueden ver reflejadas en la Tabla 3.

	Palfrey (1983)	Krishna and Rosenthal (1996)	Chakraborty (1999)	Subramanian y Venkatesh (2009)	Popkowski y Haubl (2010)
Formato de la subasta	En lotes frente a separados	Combinatoria, secuencial y simultáneo	En lotes frente a separados	En lotes, secuencial y combinatorio	En lotes, secuencial y simultáneo
Complementariedad y sustituibilidad de los componentes	Ninguna	Sólo complementariedad	Ninguna	Complementariedad y sustituibilidad	Complementariedad y sustituibilidad
Heterogeneidad en las valoraciones de los pujadores	Independientemente distribuidas	Independientemente distribuidas o correladas positivamente	Independientemente distribuidas	Independientemente distribuidas	Correladas negativamente y positivamente
Disponibilidad del producto en el exterior	No	No	No	No	Sí
Lotes ofrecidos aislados	No	No	No	No	Sí
Empírico / teórico	Teórico	Teórico	Teórico	Teórico	Empírico y teórico
Conclusiones	Beneficio relativo de las subastas de componentes individuales se incrementa a medida que aumenta el número de pujadores	La competición incrementa la exposición al riesgo, resultando en pujas menos agresivas para subastas de componentes	Cuando el número de pujadores es mayor que un número crítico, las subastas de componentes son más rentables	Las subastas por componentes separados son más beneficiosas cuando no son parecidos	Las subastas en lotes son más rentables cuando los componentes tienen un grado de similitud moderado. La heterogeneidad y la estrategia de ventas influye en los beneficios

Tabla 3. Comparativa de estudios realizados de subasta por lotes frente a productos individuales (elaboración propia)

Del análisis de estos estudios se muestra que las subastas por lotes son menos rentables para productos no complementarios, y que en cambio, son más rentables que las subastas por componentes separados cuando hay complementariedad (incluso con poco

grado) entre los productos. En el caso de análisis de este proyecto de investigación está claro que hay un alto grado de complementariedad entre los productos de acero que se sacan a subasta, y por lo tanto la estrategia de agrupar previamente en lotes es clara. Alineado también con los estudios de la literatura está el hecho de que es mejor hacer lotes de productos más homogéneos (más complementarios), tarea que conlleva una gran dificultad debido a las diferentes características que conforman un producto. La metodología de agrupar utilizada por la compañía para el problema de esta tesis se ve respaldada desde el punto de vista científico.

3.6 Recomendaciones propuestas

En este capítulo se ha hecho un estudio de los trabajos existentes en la literatura referentes a las subastas, centrándose en los tipos que se utilizan actualmente para la subasta de productos siderúrgicos. El objetivo es analizar cuáles son los factores más importantes que afectan a una subasta, para comprobar si el tipo de subasta, así como la opción de agrupar productos previamente a su venta es la manera idónea de proceder, o en caso contrario poder proponer otra alternativa a la empresa.

En cuanto al rendimiento esperado según el tipo de subasta, a pesar de que en condiciones ideales todas son iguales, diferentes estudios reales muestran que la subasta cerrada de primer precio es mejor que la inglesa. El que los pujadores puedan confabular entre sí hace lógico el uso de una subasta cerrada (respaldado de igual manera por la literatura). De igual manera, la literatura muestra que la subasta cerrada proporciona mejor rendimiento cuando los pujadores son sensibles al riesgo. Estos factores prueban que utilizar una subasta cerrada (como se hace actualmente) es la mejor opción; ya que además también evita la aparición de pujadores de última hora oportunistas.

Existen estudios que muestran que las subastas holandesas puede proporcionar mayores beneficios que las de primer precio para subastas online (Lucking-Reiley, 1999). Por ello, para intentar sacar el máximo beneficio del proceso, se propone como recomendación realizar una subasta descendente con aquellos productos que no han sido vendidos tras varias semanas en la subasta ascendente de primer precio. El precio puede ir descendiendo de manera gradual hasta bajar al precio de chatarra, momento en el cual un producto dejaría de estar a la venta y volvería realmente a la planta para convertirse en chatarra. De esta manera se espera conseguir mayores beneficios derivados de buenas ofertas “de última hora” a clientes, que seguramente paguen más que el precio de chatarra. El proceso de ventas quedaría por tanto como se muestra en la en la Figura 10.

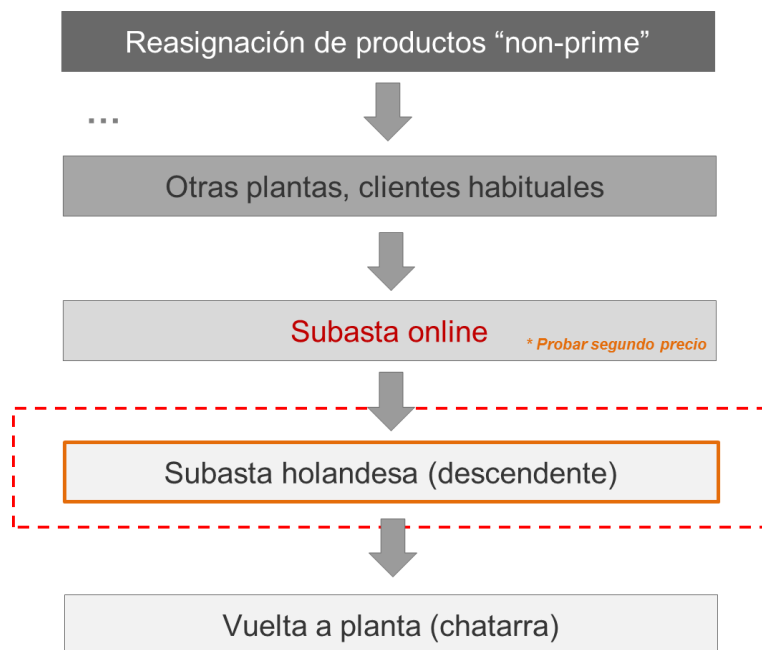


Figura 10. Nueva propuesta de modelo de ventas

La utilidad de establecer un precio mínimo para obtener más beneficio también es respaldada por la literatura; siempre y cuando este precio refleje bien el mercado.

La duración de la subasta ha de ser suficiente para que puedan entrar más pujadores, pero sin perder el coste de oportunidad, con lo que como se realiza ahora parece apropiado. En cuanto al fraude, todos los pujadores son clientes conocidos, actualmente se controla si algún cliente no paga adecuadamente, llegando incluso a vetarlo de la subasta. En el caso de que se registren nuevos clientes solamente se podrá hacer un seguimiento a posteriori tras realizar alguna venta.

Se ha comprobado que la política de la compañía de agrupar productos similares tiene un respaldo científico, destacando la importancia de la homogeneidad de cada lote y, por lo tanto, el valor de esta tesis. En el contexto de la literatura científica, el problema de agrupar productos en lotes se enmarca dentro de los problemas Bin Packing, o problema de la mochila. El caso concreto de los productos non-prime analizado en esta tesis es más complejo ya que existen más de un objetivo a la hora de crear los lotes: que sean lo más homogéneos posibles, que queden el menor número posible de productos sin agrupar, que cada lote tenga un

tamaño mínimo y que los lotes tengan el mayor tamaño posible dentro de un rango especificado. Además, algunas de las características de los productos (grado de acero, categoría) deben de ser iguales para todos los productos del lote. En el siguiente capítulo se realiza un estado del arte del Bin Packing Problem, para analizar las diferentes aproximaciones utilizadas en la literatura antes de proponer una solución al problema propuesto.

4 Problema del empaquetado en recipientes (Bin Packing Problem)

Como se comentaba en el apartado anterior, desde un punto de vista algorítmico el problema de agrupar productos siderúrgicos en lotes puede verse como un caso concreto del clásico problema de empaquetado en recipientes (Bin Packing Problem). Por esta razón, a continuación se analizará el problema de bin packing y sus variantes, así como las diferentes técnicas que existen en la literatura para su resolución.

En este apartado se describe primero el problema original junto con sus diferentes variantes, para luego realizar el estado del arte de aquellas variantes que puedan presentar semejanzas con el objetivo de la tesis, así como de las diferentes técnicas de resolución de dichos problemas. El análisis de las diferentes aproximaciones y técnicas de resolución utilizadas se realiza como paso previo a poder plantear una posible solución al problema de esta tesis, de cara a analizar las mejores directrices del algoritmo a desarrollar, teniendo en cuenta las restricciones y complejidad tanto desde el punto de vista del algoritmo como del limitado tiempo de ejecución.

4.1 Descripción del Bin packing problem

El problema de empaquetado en recipientes (Bin Packing problem) se trata de asignar un conjunto de artículos a un número mínimo de recipientes, sujetos a una restricción de tamaño para cada bloque. En términos computacionales este problema es NP-Hard (Garey & Johnson, 1979), por lo tanto, es improbable encontrar algoritmos de optimización eficientes (tiempo polinomial) que encuentren la solución óptima, siendo por ello usual el empleo de algoritmos aproximados, que no garantizan encontrar la solución óptima, pero intentan acercarse lo más posible dentro de tiempos de computación razonables.

La versión clásica del problema se define por un infinito número de recipientes con capacidad C , y una lista $L = \{a_1, \dots, a_n\}$ de artículos. Cada artículo tiene un tamaño $w_i \equiv w(a_i)$, que satisface que $0 \leq w_i \leq C$, $1 \leq i \leq n$. El problema consiste en empaquetar los artículos en un

número mínimo de recipientes, con la restricción de que la suma de los tamaños de los artículos en cada recipiente no es mayor que C . En la versión clásica del problema todos los recipientes tienen la misma capacidad, con lo que muchas veces se adopta la normalización $C =$

1. Matemáticamente puede expresarse así:

$$\text{Minimizar} \quad z = \sum_{i=1}^n y_i \quad (1)$$

$$\text{Sujeto a} \quad \sum_{j=1}^n w_j \cdot x_{ij} \leq C \cdot y_i, \quad i \in N = \{1, \dots, n\} \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j \in N \quad (3)$$

$$y_i = 0 \mid 1, \quad j \in N \quad (4)$$

$$x_{ij} = 0 \mid 1, \quad j \in N \quad (5)$$

$$w_i \leq C, \quad \forall j \in N \quad (6)$$

Donde

$$y_i = \begin{cases} 1, & \text{si el recipiente } i \text{ es usado} \\ 0, & \text{en cualquier otro caso} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{si el elemento } j \text{ se asigna al recipiente } i \\ 0, & \text{en cualquier otro caso} \end{cases}$$

La restricción (2) garantiza que la suma de los artículos no exceda el tamaño del recipiente, mientras que la restricción (3) garantiza que cada producto se asigne únicamente a un recipiente. Las restricciones (4) y (5) definen las variables como binarias, mientras que la restricción 6 obliga a que el peso de cada artículo sea menor que el tamaño del recipiente.

4.2 Variaciones del problema clásico

Existen numerosas variantes del problema de Bin Packing anteriormente descrito. Se resumen a continuación las variantes encontradas, apoyándose en la clasificación propuesta en (Coffman Jr, et al., 2013), que divide las variantes en: variaciones respecto al tamaño máximo de los recipientes, versiones duales del problema, variaciones en la asignación de los artículos a los lotes y, por último, otras condiciones adicionales. En esta tesis se propone una nueva clasificación, incluyendo para cada apartado otras variantes encontradas en la literatura que no estaban presentes en la clasificación original. Se han agrupado las variantes encontradas en los siguientes tipos: versiones duales del problema, variaciones en tamaño de recipiente, variaciones en la asignación de los artículos, variaciones con restricciones sobre los artículos, y generalizaciones del problema. En la Figura 11 se resume la clasificación. En los siguientes apartados se describe cada variante y la aplicabilidad o no para el problema a resolver en la tesis.

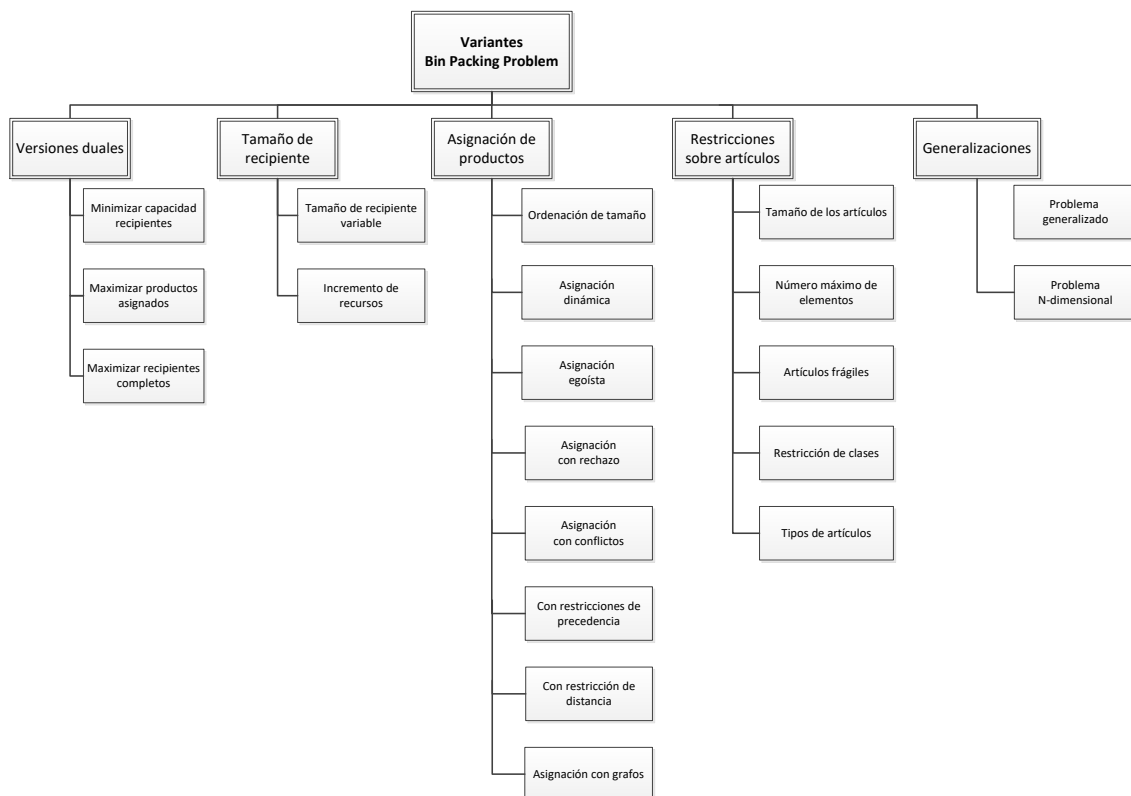


Figura 11. Clasificación propuesta de las variantes del Bin Packing Problem

4.2.1 Versiones duales del problema

Existen otras variantes en las que la definición del problema es la misma, pero cambia el objetivo a conseguir. Las 3 posibilidades analizadas en la literatura son:

Minimizar la capacidad de los recipientes: se asume que el número de recipientes es fijo, y el objetivo en este caso es definir la menor capacidad C tal que dada una lista L de artículos se puedan asignar a un número finito de lotes de capacidad C (Hall, 1996). Este problema también se denomina programación de máquinas en paralelo o multi-procesador, y cuenta con numerosas aplicaciones en la industria. En el problema a resolver en esta tesis, al ser el tamaño máximo fijo por planta debido a restricciones logísticas no tiene sentido minimizar la capacidad de los recipientes.

Maximizar el número de productos asignados: en este caso el número de recipientes es un número finito y fijo, y el objetivo es asignar el mayor número posible de artículos a los lotes. Este problema es importante en aplicaciones informáticas, para maximizar el número de elementos almacenados en uno o más niveles de memoria, o para maximizar el número de tareas realizadas en múltiples procesadores en un intervalo de tiempo determinado (Azar, et al., 2000). Aunque no es el objetivo de la tesis, sí que tiene cierta similitud con el problema a resolver, ya que intrínsecamente se desea dejar el menor número posible de bobinas sueltas, sin embargo en el problema de la tesis no existe una limitación en cuanto al número de recipientes.

Maximizar el número de recipientes completos: en esta variante del problema el objetivo es agrupar un conjunto de artículos en un número máximo de recipientes, con la restricción de que el contenido de cada recipiente ha de ser mayor de un cierto límite establecido, a partir de lo cual el recipiente se considera completo (Coffman, et al., 1987). Entre las principales aplicaciones se encuentran el empaquetado de bienes en latas, de forma que cada lata contenga al menos la cantidad especificada, y la estimulación de la actividad económica asignando tareas a un número máximo de fábricas. Como en el caso anterior, en el problema a resolver en este trabajo no hay un número finito de recipientes en el que se deban meter los productos, y aunque hay que intentar que la capacidad sobrante del recipiente sea mínima no es el objetivo principal el maximizar los recipientes completos.

4.2.2 Tamaño variable de los recipientes

En este apartado se incluyen versiones del Bin Packing Problem con alguna diferencia respecto al tamaño de los recipientes del problema original, en el que se considera que todos los lotes son iguales y con una capacidad máxima. Las principales variantes existentes son:

Tamaño de recipiente variable (“Variable-sized Bin”): existen diferentes tipos de recipientes B_1, B_2, \dots, B_k , con tamaños $S(B_1) > S(B_2) > \dots > S(B_k)$. Para cada tipo existe un número infinito de recipientes disponibles, siendo el objetivo asignar una lista L de artículos a un subconjunto de recipientes consiguiendo el menor tamaño total de los mismos. Este problema tiene gran importancia en aplicaciones de reducción de stock o asignación de memoria en los sistemas operativos de los ordenadores (Kang & Park, 2003). Esta variante no aplica a los objetivos del problema a analizar, ya que en nuestro problema existe un tamaño máximo de recipiente para cada planta.

Con incremento de los recursos (“Resource Augmentation”): esta idea fue introducida por Csirik y Woeginger (Csirik & Woeginger, 1998). Al igual que en la versión original, cada artículo tiene un tamaño fijo $(0,1]$, pero los recipientes no tienen capacidad $C=1$, sino que pueden variar y ver su capacidad incrementada en $C \geq 1$. De nuevo, en nuestro caso la problemática a resolver es distinta, debido al tamaño máximo fijo por planta.

4.2.3 Variaciones en la asignación

Algunas variantes de la versión clásica del problema afectan directamente a la forma en la que los artículos pueden ser asignados a los diferentes lotes. En este apartado se muestran las diferentes posibilidades analizadas en la literatura:

Con ordenación de tamaño (“LIB Constrained”): esta restricción se denomina LIB (Larger Item on the Bottom), siendo necesario que un artículo de mayor tamaño vaya siempre colocado debajo de otro menor. El problema ha sido analizado en (Manyem, et al., 2003). Para el caso de las agrupaciones de productos de acero este factor no es relevante.

Asignación dinámica (“Dynamic Bin Packing”): el problema de asignación dinámica fue descrito por primera vez en (Coffman, et al., 1983). En esta variante, existe un número máximo de recipientes donde se pueden asignar los artículos, y los artículos llegan arbitrariamente.

Este problema modela por ejemplo la asignación dinámica de memoria en los sistemas operativos; no tiene semejanza con el problema que se quiere resolver en la tesis.

Asignación egoísta (“Selfish bin packing”): esta variante considera el caso en el que los artículos son controlados por agentes egoístas, y el coste de un recipiente es dividido entre los agentes en proporción al porcentaje que sus artículos ocupen (cada agente por lo tanto querrá que sus productos se asignen a un lote que esté lo más lleno posible). Este problema se corresponde con un juego no cooperativo, con lo que a veces se denomina también juego del empaquetado “bin packing game” y fue analizado por primera vez desde una perspectiva de juego en (Bilo, 2006). No tiene similitud con el problema a resolver en la tesis.

Asignación con rechazo (“Bin Packing with Rejection”): además del coste de los recipientes, el algoritmo considera que cada artículo tiene un coste de rechazo que se ha de pagar si el algoritmo decide no asignar este producto a ningún recipiente. El origen de esta variación viene motivado por el siguiente problema de gestión de archivos: se supone que existen una serie de archivos que son usados por un sistema local. Cada archivo puede ser descargado con antelación o ser descargado por el sistema solo cuando es realmente necesario. La primera opción necesita espacio en un servidor local, y tiene un coste de transmisión local, mientras que un programa usa el archivo. La segunda opción tiene un coste de comunicación mientras el sistema descarga el archivo de un servidor externo. Un algoritmo que gestiona el sistema de archivos locales ha de elegir si guardar un archivo en el dispositivo de almacenamiento local o descargarlo directamente, en función del coste de cada opción. Fue propuesto por primera vez en (Dósa & He, 2006). En nuestro caso no es posible determinar el coste que tendría dejar a alguna bobina fuera de cualquier recipiente, con lo que no se puede aplicar esta aproximación.

Asignación con conflictos (“Bin Packing with Conflicts”): en esta variante, existe un conjunto de artículos $L = a_1, a_2, \dots, a_n$ con tamaños s_1, s_2, \dots, s_n y un grafo de conflictos $G(L, E)$. El objetivo es encontrar el número mínimo de conjuntos tales que la suma de los tamaños de cada conjunto es como mucho 1. Este problema es una generalización tanto del clásico problema de empaquetado de productos (cuando $E=0$), como del problema de coloreado de grafos (cuando $s_i=0$ para todo i) (Epstein & Levin, 2008). Esta variante, si bien no tiene toda la

complejidad de nuestro caso de uso, sí que tiene mucha similitud por lo que será estudiado con detalle en el apartado 4.4.

Restricciones de precedencia (“Bin Packing with Partial Orders”): en esta generalización, se aplican restricciones de precedencia entre los artículos, que se aplican sobre el orden relativo de los recipientes: $si <$ denota el orden parcial dado las restricciones de precedencia, entonces $si a_i < a_j$ quiere decir que $si a_i, a_j$ son empaquetados en B_r y B_s respectivamente, entonces $r \leq s$. En la literatura se han considerado dos aplicaciones prácticas: la primera es el problema de balanceado en líneas de montaje, en el que las líneas de montaje son estaciones idénticas (los lotes), donde los productos paran por un periodo de tiempo igual a la capacidad del recipiente. El tamaño de los productos son las duraciones de las tareas que se deben realizar, y un orden parcial es impuesto: $a_i < a_j$ quiere decir que la estación de trabajo a la que se asigna a_i debe de ir antes que a la que se asigna a_j (Wee & Magazine, 1982). La segunda aplicación surge en programación multi-tarea: aquí cada producto se corresponde con un proceso de duración unitaria con un requisito de memoria igual al tamaño del artículo. La capacidad del recipiente mide la disponibilidad de memoria total, obligando a algunas tareas a ir antes que otras mediante la precedencia (Garey & Johnson, 1979). No tiene relación con el problema planteado en esta tesis.

Con restricción de Distancia (“Bin Packing under Distance Constraint”) / Artículos agrupados (“Bin Packing with Clustered Items”): en esta generalización, se considera una propiedad de los artículos, y se permite agrupar los productos siempre y cuando la diferencia de valores entre esa propiedad (por ejemplo el peso) sea menor que un entero D . Este problema tiene aplicación en contextos geográficos en la que se agrupan por ejemplo ciudades por distancia entre ellas (Chandra, et al., 1978). Esta variante también encaja bastante bien con el problema a analizar, aunque en nuestro caso existen varias propiedades que deben ser analizadas por cada artículo.

Asignación de productos con grafos (“Sets and Graphs Packing”): en esta variante del problema se considera una capacidad de los lotes C . Una colección de artículos se puede asignar a un lote solamente si su unión tiene como mucho C elementos, siendo el objetivo asignar los productos en el menor número posible de recipientes. Este problema se utiliza en problemas de optimización de máquinas que fabrican circuitos electrónicos, en las que las

máquinas puedan guardar hasta un número C de componentes. En (Katona, 1997) se analiza un caso especial de asignación con grafos: un grafo es denominado p -polyp si consiste en p caminos simples de la misma longitud y comparten un nodo final común. A partir de esto, se define *Polyp Packing* como: asignar un conjunto de caminos de diferentes longitudes en un número mínimo de grafos polyps de nodos disjuntos. Sería posible plantear una solución al problema planteado en cuanto a la compatibilidad de los productos de un lote, sin embargo la obligación de tratar de que la homogeneidad sea máxima no está contemplada en esta variante.

4.2.4 Restricciones sobre los artículos

Se incluyen en este apartado otras restricciones encontradas en la literatura y aplicadas en problemas de empaquetado de productos.

Restricción en el tamaño de los artículos (“Item-Size Restrictions”): en este caso se considera que se tienen un número finito k de tamaños de artículo, con lo que solamente existen un número N de posibles combinaciones de productos en un recipiente. Este problema fue propuesto por primera vez en (Baewicz & Ecker, 1983), y se ha demostrado que tiene una complejidad menor que el problema original, siendo posible resolverlo en tiempo lineal (Lenstra Jr, 1983). Este problema tiene su aplicación en asignación de memoria en ordenadores, donde las capacidades de los dispositivos y los tamaños de bloques de memoria tienen un tamaño finito de potencias de 2. En la tesis el problema es más complejo, ya que el tamaño de los productos no sigue ninguna regla.

Restricción en el número máximo de elementos (“Cardinality Constrained”): esta variante considera que el número máximo de artículos que se pueden asignar a un recipiente está limitado por un entero positivo p . El problema tiene aplicaciones prácticas en programación de multiprocesadores en el que el número de procesadores es fijo, o en sistemas operativos multi-tarea donde el número de tareas es limitado (Kellerer & Pferschy, 1999). En el problema a resolver en la tesis no existe restricción de número máximo de elementos en cada recipiente.

Asignación de artículos frágiles (“Fragile Objects Bin Packing”): este problema se define en (Bansal, et al., 2009), en el que cada artículo tiene dos atributos: tamaño y fragilidad.

El objetivo es asignar los artículos de manera que la suma de los tamaños de los productos en cada recipiente no sea mayor que la fragilidad de cada artículo en el recipiente. Chan et al. generalizan el problema, analizando el caso en el que el ratio entre una fragilidad máxima y mínima es limitado por un valor k . Tampoco esta variante (Chan, et al., 2007) es aplicable en nuestro caso.

Restricciones de clases (“Class Constrained Bin Packing”): en el problema del empaquetado por lotes cada artículo a_i tiene dos parámetros: el tamaño s_i y el color c_i , siendo $s_i \in (0,1]$ y un color representado por un entero positivo: $c_i \in [1, q]$. El número de colores existentes dentro de un producto también está limitado por un parámetro Q . Al conjunto de artículos con el mismo color se le denomina clase. El objetivo es asignar los artículos en el menor número de lotes de manera que cada lote tenga como mucho Q clases diferentes de colores (Shachnai & Tamir, 2001).

Tipos de artículos (“Bin Packing with Item Types”): analizando la agrupación de anuncios en páginas web, Adler et al. encuentran una variante del problema del empaquetado en el que los artículos se clasifican en tipos: el problema es asignar los productos en recipientes de manera que ningún recipiente tenga dos artículos del mismo tipo. No es similar al problema a resolver (Adler, et al., 2002).

4.2.5 Generalizaciones del problema

Existen 2 variaciones que plantean el problema de manera genérica, denominadas “problema de empaquetado de artículos generalizado” y “n-dimensional”. Estas variantes se explican a continuación.

Problema de empaquetado de artículos generalizado (“Generalized Bin Packing Problem”): bajo este nombre se incluye una variación que consiste en clasificar los artículos en “obligatorios de incluir” y “no obligatorios”, cada uno con un cierto volumen y beneficio. A su vez, existe un conjunto de recipientes con un volumen dado y coste, el objetivo es seleccionar el subconjunto de artículos “no obligatorios” que junto con los “obligatorios de incluir” minimicen el coste neto total de los recipientes. Este coste neto se calcula como la diferencia entre el coste total de los lotes seleccionados y el beneficio total de los productos asignados (Baldi & Bruglieri, 2016). Tampoco es similar al problema a resolver.

Problema de empaquetado de artículos n-dimensional (“N-Dimension Bin Packing Problem”): bajo este nombre se encuentran una serie de variantes del problema en la que existe más de una dimensión relevante para los artículos. Por ejemplo, en el caso de 2 dimensiones, el problema se define como: dado un conjunto de productos rectangulares, cada artículo con un peso w y una altura h , y un conjunto de lotes de idéntico tamaño con un peso W y una altura H , el objetivo es encontrar el mínimo número de recipientes necesarios para asignar todos los artículos, intentado llenar los recipientes de la mejor manera posible (Dahmani, et al., 2015). De la misma manera se encuentran ejemplos del problema con 3 dimensiones, como en (Saraiva, et al., 2015).

4.2.6 Conclusiones y similitud de las variantes con el problema a analizar

Una vez analizadas las diferentes variantes encontradas del problema, se observa que no existe hasta el momento en la literatura ninguna variante que presente todas las complejidades que atañen al problema que se plantea en esta tesis. Un algoritmo que resolviera de manera global todo el problema debería incluir las siguientes variantes:

- Límite inferior en los lotes → no se pueden agrupar varios artículos en el mismo recipiente si no suman un peso mínimo.
- Con conflictos → no pueden agruparse en el mismo lote aquellos artículos que tengan distintos valores en las categorías: planta, localización, forma, categoría, familia, subfamilia (para algunos casos concretos) y número de lados de recubrimiento.
- Con restricción de distancia / artículos agrupados: los artículos tienen una serie de características (subfamilia, grado de acero, aceitado, peso, ancho, espesor y espesor de recubrimiento) que deben ser lo más parecidas posibles. En los problemas de productos agrupados sólo se considera 1 propiedad de cada producto mientras que en este caso se deben considerar 7. Se ha de notar también que el caso concreto del problema con restricción de distancia se puede considerar un caso puntual del problema con conflictos, si se considera un problema de

empaquetado con conflictos en el cual 2 productos entran en conflicto si su distancia es mayor que una distancia d_{max} permitida.

- En el caso concreto del parámetro subfamilia, no encontramos antes dos situaciones distintas (permitir mezclar diferentes subfamilias en lotes u obligar a que sea necesariamente la misma), en función del tipo de subfamilia del producto. Esto complica aún más el problema.

Cada una de estas variantes del problema es computacionalmente muy compleja, por lo que si se trata de realizar un algoritmo en un único paso sería improbable conseguir una solución en los tiempos necesarios para el problema de la tesis. En lugar de eso, lo que se realizará es una división previa de los datos de entrada atendiendo a aquellos parámetros que deben de ser comunes en cada recipiente (planta, localización, forma, categoría, familia, y número de lados de recubrimiento) para así dividir el problema original en n sub-problemas, y poder realizar un algoritmo más eficiente.

Una vez hecha esa división, el problema final que el algoritmo debe de resolver engloba las características del problema de empaquetado en recipientes con conflictos y con límites de tamaño (en cada ejecución del algoritmo habrá un tamaño mínimo y máximo en función de la planta), con la condición adicional de que en cada recipiente los artículos deben de ser lo más homogéneos posible según las características comentadas. No se ha encontrado ningún problema similar en la literatura. Para poder plantear una solución, se analizarán en detalle los estudios realizados sobre el problema de empaquetado con conflictos; de la misma manera primero se realizará también un estudio global de técnicas utilizadas en general en cualquier problema de Bin Packing de cara a ampliar el posible marco de soluciones a utilizar.

4.3 Principales heurísticos utilizados en problemas Bin Packing

En este apartado se muestran los principales heurísticos utilizados para resolver el Bin Packing Problem, lo cual servirá de base para analizar los diferentes enfoques utilizados hasta el momento en problemas similares al planteado en esta tesis. Como se ha comentado en el apartado 4.1, el problema tiene una complejidad NP-hard, con lo que conseguir encontrar la solución óptima requiere de tiempos de computación muy elevados. Este hecho hace que la tendencia a resolver este problema haya sido mediante algoritmos heurísticos, generando candidatos de posibles soluciones siguiendo un patrón determinado, intentando conseguir la mejor solución posible (incluso a veces la óptima) con buenos tiempos de ejecución.

Tradicionalmente, los algoritmos de resolución del problema se han clasificado en online y offline. Los algoritmos online asignan los artículos a un recipiente según van apareciendo, con lo que no existe conocimiento de los siguientes artículos que aparecerán, mientras que en los algoritmos offline sí que se tiene conocimiento de todos los artículos en la secuencia, siendo posible prepararlos en un orden particular antes de asignarlos a los recipientes. Recientemente en la literatura se habla también de los algoritmos semi-online, en los que se permiten reasignar artículos, pero sin conocer todo el conjunto completo. A continuación se muestran los principales algoritmos utilizados en las tres variantes.

4.3.1 Principales algoritmos para la variante online

Los principales algoritmos online existentes para solucionar el problema del empaquetado en recipientes son los siguientes:

Algoritmo Next-Fit: el funcionamiento del algoritmo es el siguiente: en primer lugar se abre un recipiente y se asigna el primer artículo al mismo; tras asignar el primer artículo, el algoritmo Next-Fit (NF) asigna cada artículo sucesivamente en el recipiente que contiene el último artículo asignado si es que cabe en el recipiente; si no, NF cierra el recipiente y asigna el artículo actual en un nuevo recipiente vacío.

La complejidad de este algoritmo es $O(n)$. Solamente se abre un nuevo recipiente cada vez, con lo que el espacio es limitado, sin embargo, el ratio de rendimiento asintótico (APR) es relativamente bajo. Este algoritmo fue propuesto en (Johnson, 1973). Fisher apunta una propiedad del algoritmo NF respecto a otros algoritmos de aproximación: este algoritmo asigna cualquier lista de artículos y su inversa en el mismo número de recipientes (Fisher, 1988). La desventaja de este algoritmo es que se cierran recipientes que se podrían aprovechar para añadir otros artículos de menor tamaño.

Algoritmo Worst-Fit: propuesto por el mismo autor del Next-Fit (Johnson, 1974), este algoritmo (WF) surge para intentar superar la desventaja del Next-Fit. La idea de la que parte es no cerrar recipientes para encajar así más artículo; una vez que existen más de un recipiente abierto hay que decidir a cual lote se asigna cada artículo. El funcionamiento del algoritmo es el siguiente: empieza como el Next-Fit, y una vez que no hay ningún recipiente abierto en el que asignar el artículo entonces se abre un nuevo recipiente donde se asigna. En el caso de que haya varios recipientes abiertos en los que el producto encaje, el algoritmo lo asigna al recipiente menos lleno, como se propone en (Vigo, 2013). En el caso de que existan 2 con suficiente espacio y misma capacidad disponible, se asigna al recipiente creado primero. A pesar de que para cada problema en concreto la solución varía, en general el algoritmo WF se comporta peor que el NF. La complejidad del algoritmo es $O(n \log n)$.

Algoritmo First-Fit: el algoritmo First-Fit (FF) es una variante del WF, en este algoritmo se asigna el artículo actual al recipiente de menor índice (el primero creado de los disponibles) en el que encaje. Si no se puede asignar a algún recipiente, se crea uno nuevo y se asigna el artículo al mismo. Este algoritmo no es lineal en el tiempo ni limitado en el espacio. La complejidad del algoritmo es $O(n \log n)$.

Algoritmo Best-Fit: en este algoritmo (BF) se asigna el artículo actual al recipiente abierto que tenga el menor espacio disponible siempre y cuando el artículo quepa; si no hay ningún recipiente abierto en el que encaje el artículo se asigna a un nuevo recipiente vacío. La complejidad del algoritmo es $O(n \log n)$ (Falkenauer & Delchambre, 1992).

Algoritmo Almost-Worst-Fit: este algoritmo (AWF) es una modificación del WF. En este caso el artículo actual siempre se asigna al segundo recipiente menos lleno. En el caso de

que solamente se pueda asignar a un recipiente se asigna, y si no es posible asignar a ninguno se abre uno nuevo.

Algoritmos k -bounded space: en este tipo de algoritmos, para cada artículo, la elección de dónde asignar cada artículo está restringida a un conjunto de como mucho k recipientes abiertos. El caso del algoritmo NF se puede considerar como un algoritmo limitado a 1 espacio (1-bounded space). Para mejorar el ratio de rendimiento asintótico Johnson propone un algoritmo que asigna artículos en función de la regla First-Fit, pero considera como candidatos sólo los k recipientes abiertos más recientes. Cuando un nuevo recipiente ha de ser creado y ya hay k lotes abiertos, entonces el recipiente con menor índice se cierra (Johnson, 1974). En general, los algoritmos limitados en espacio se definen por las reglas de asignación y empaquetado, siendo por ejemplo reglas comunes:

- Asignación: los elementos se asignan siguiendo o la regla First-Fit o la Best-Fit.
- Cerrado de lotes: el lote a cerrar es o el de menor índice o uno de los más llenos.

Algoritmo Refined First-Fit (RFF): este algoritmo, propuesto en (Yao, 1980) clasifica los productos en tipo 1, 2, 3 o 4 en función de que sus tamaños estén en los intervalos $(0, 1/3]$, $(1/3, 2/5]$, $(2/5, 1/2]$, $(1/2, 1]$. RFF asigna 4 secuencias de recipientes, una para cada tipo. Los productos tipo- i se asignan a la secuencia de recipientes tipo- i utilizando la regla First-Fit, con la excepción de que cada 6 artículos de tipo-2 (con tamaño en el intervalo $(1/3, 2/5]$) se junta uno con los de tipo-4. Este algoritmo no es limitado en espacio y tiene una complejidad $O(n \log n)$. Todas las variantes de este algoritmo que dividen los artículos en rangos se denominan algoritmos armónicos.

4.3.2 Algoritmos semi-online

A pesar de que la literatura tradicional distingue entre algoritmos online y offline, existen una serie de algoritmos que se podrían considerar en la frontera entre ambos, en los cuales está permitido reasignar artículos y mirar hacia adelante o hacia atrás antes de asignar el artículo actual, pero sin conocer todo el conjunto completo. Se incluyen en este apartado los principales algoritmos desarrollados:

Algoritmo Buffered Next-Fit (BNF): propuesto por en (Galambos, 1986), este algoritmo usa 2 recipientes abiertos (B1 y B2). Los artículos se van asignando inicialmente a B1, hasta que llega un artículo que no tiene suficiente espacio. Este artículo y los que ya estaban asignados a B1 son reordenados en tamaño decreciente y reasignados en B1 y B2 siguiendo la regla Next-Fit. B1 se cierra, B2 se renombra como B1 y se abre un nuevo lote B2.

Algoritmo Repacking (REP): desarrollado a partir del algoritmo anterior, y propuesto en (Galambos & Woeginger, 1993). Este algoritmo utiliza 3 recipientes abiertos. Cuando un nuevo artículo aparece, se siguen los siguientes pasos:

- El artículo se asigna a un lote vacío
- Todos los artículos en los tres recipientes abiertos, ordenados en tamaño decreciente se reasignan utilizando la regla First-Fit, con el resultado de que o un recipiente se vacía o al menos uno excede el tamaño posible.
- Todos los recipientes cuyo tamaño supera el permitido se cierran y se reemplazan por nuevos recipientes vacíos.

Algoritmo Revised Warehouse(RW): este algoritmo, propuesto en (Grove, 1995), considera que si un artículo no se puede asignar a ningún recipiente abierto por falta de espacio, se puede almacenar temporalmente en un almacén de tamaño W , hasta que el tamaño del propio almacén no permita almacenar más artículos.

4.3.3 Principales algoritmos para la variante offline

Los algoritmos offline consideran que todos los artículos a asignar pueden ser pre-procesados, reordenados, agrupados, etc. antes de ser asignados a recipientes. Los principales algoritmos existentes son:

Algoritmos con reordenación: cuando la lista ordenada de artículo por tamaño decreciente es asignada a recipientes utilizando la regla Next-Fit se obtiene el algoritmo Next-Fit Decreasing (NFD), de la misma manera también existen los algoritmos First-Fit Decreasing (FFD) y Best-Fit Decreasing (BFD). Baker y Coffman prueban que el ratio de aproximación asintótico de NFD es el mejor de los 3 (Baker & Coffman, 1981). En (Yao, 1980) se propone un

complejo algoritmo denominado Refined First-Fit Decreasing, que a su vez fue modificado en (Johnson & Garey, 1985); en este algoritmo, la principal idea es tratar de mejorar la parte de la asignación de los recipientes con productos de tamaño mayor que $\frac{1}{2}$, intentando asignar en estos recipientes pares de elementos con tamaños comprendidos entre $(\frac{1}{6}, \frac{1}{3}]$. Friesen y Langston proponen otra modificación del FFD, denominado algoritmo Best Two-Fit (B2F). Este algoritmo empieza rellenando un recipiente de manera voraz. Cuando no se pueden asignar más artículos al recipiente, y el mismo contiene más de un artículo, se intenta reemplazar el más pequeño por dos artículos sin empaquetar de tamaños al menos $\frac{1}{6}$ (Friesen & Langston, 1991); en el caso de que todos los artículos empaquetados tengan tamaños menores que $\frac{1}{6}$, se aplica el algoritmo estándar FFD. Todos estos algoritmos tienen complejidad $O(n \log n)$.

Group-X-Fit Grouped: este heurístico, propuesto en (Johnson, 1974) consigue soluciones en tiempo real, sin ordenar los artículos. Este algoritmo, denominado Group-X-Fit Grouped (GXFG) depende de la elección de un conjunto de puntos de ruptura o pesos para dividir los artículos. Para cada punto de ruptura el algoritmo divide los artículos según su peso en un conjunto de clases, renombrando los artículos de tal manera que los artículos de la misma clase sean consecutivos, y las clases se ordenen por tamaño decreciente. Los artículos se asignan a recipientes utilizando la regla Best-Fit con respecto al hueco actual de cada lote .

4.4 Bin Packing con Conflictos (Bin packing with Conflicts)

Como se comentaba en el apartado 4.2, el problema a resolver en esta tesis presenta gran similitud con el Bin Packing con conflictos. Debido a esto, en este apartado se realiza un estudio detallado del mismo. Este problema es también el caso genérico del denominado problema de Bin Packing con restricción de distancia, o en ocasiones también llamado Bin Packing con productos agrupados. Se muestra la definición del mismo, las principales aproximaciones utilizadas para resolverlo, las heurísticas de prueba más comunes y los tiempos empleados por los diferentes algoritmos. Este análisis servirá como base para el posterior planteamiento del algoritmo propuesto en la tesis.

4.4.1 Definición del problema

El problema de empaquetado en recipientes con conflictos (BPPC) se define de la siguiente manera: dado un conjunto $V = \{1, 2, \dots, n\}$ de artículos, cada artículo i con un peso no negativo w_i , y un número infinito de recipientes idénticos con capacidad C , y dado un grafo de conflictos $G = (V, E)$, donde E es un conjunto de arcos tal que $(i, j) \in E$ cuando los artículos i y j están en conflicto; el objetivo del BPPC es asignar todos los artículos al menor número de recipientes, asegurando que el peso total de los artículos asignados a un recipiente no excede su capacidad, y que ningún recipiente contiene artículos en conflicto. El problema es NP-hard (Muritiba, et al., 2010), ya que generaliza tanto el problema de empaquetado en recipientes (BPP) como el problema de coloreado de vértices (VCP). Se puede considerar el BPP como un caso concreto del BPPC en el que ningún artículo entra en conflicto con otro (Martello & Toth, 1990).

El BPPC puede ser modelado introduciendo dos conjuntos de variables binarias:

y_h , con valor 1 si el lote h se usa, y 0 sino

x_{ih} , con valor 1 si el artículo i se asigna al lote h , y 0 sino $| (i= 1, 2, \dots, n) (h= 1, 2, \dots, n)$

Matemáticamente el problema puede ser expresado de la siguiente forma (Muritiba, et al., 2010):

$$\min \sum_{h=1}^n y_h \quad (1)$$

$$\sum_{h=1}^n x_{ih} = 1 \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n w_i x_{ih} \leq C y_h \quad h = 1, 2, \dots, n \quad (3)$$

$$x_{ih} + x_{jh} \leq y_h \quad (i, j) \in E, h = 1, 2, \dots, n \quad (4)$$

$$y_h \in \{0, 1\} \quad h = 1, 2, \dots, n \quad (5)$$

$$x_{ih} \in \{0, 1\} \quad i = 1, 2, \dots, n \quad h = 1, 2, \dots, n \quad (6)$$

La restricción (2) requiere que cada artículo sea asignado a un único recipiente. Con la restricción (3) se asegura que no se supera la capacidad máxima de los recipientes. La restricción (4) se utiliza para introducir los conflictos, no permitiendo que se junten productos incompatibles. Las restricciones (5) y (6) garantizan que las variables x e y tienen valores binarios.

4.4.2 Algoritmos existentes

En esta sección se resumen los principales algoritmos propuestos para resolver el problema de empaquetado en recipientes con conflictos.

4.4.2.1 First fit decreasing modificado (Modified FFD)

Este algoritmo es una adaptación directa del algoritmo FFD propuesto en (Coffman Jr, et al., 1984), ordenando previamente los artículos, y asignándolos sucesivamente al primer recipiente en el que haya suficiente capacidad, con un chequeo adicional para asegurar la ausencia de conflictos (Gendreau, et al., 2004).

Cuando todos los artículos son mutuamente sin conflictos, la complejidad del heurístico es $O(n \log n)$ (Coffman Jr, et al., 1984). Sin embargo, ya que $O(n^2)$ conflictos están presentes, la complejidad es también $O(n^2)$.

4.4.2.2 Aproximación asintótica

Jansen et al. proponen un esquema de aproximación asintótica para resolver el problema, restringiéndose a grafos d -inductivos (aquellos en los que los vértices puede ser numerados de manera que cada vértice tenga como mucho d arcos a vértices con mayor numeración). Este algoritmo encuentra una asignación para los artículos de manera que el número de recipientes generados se encuentra dentro de un factor de $(1+\varepsilon)$ del óptimo proporcionado, siempre que el número óptimo sea suficientemente grande (Jansen, 1999).

El algoritmo sigue los siguientes pasos:

- 1) Se parte de una instancia I consistente en un grafo d -inductivo $G=(V,E)$ con $V=\{1,\dots,n\}$ y tamaños $s_1,\dots,s_n \in (0,1]$.
- 2) Basándose en el problema de empaquetado restringido (restricted bin packing problem=RBP), que se define como un problema de empaquetado sin conflictos limitado a las instancias donde los tamaños de los artículos toman como mucho m valores distintos, y el tamaño de cada artículo es como mucho de tamaño δ , se define un $\delta=\varepsilon/2$, eliminando todos los artículos de tamaño menor que δ , obteniendo una instancia J del problema de empaquetado restringido $RBP[\delta,n']$ con n' artículos mayores.
- 3) Se aplica el método de agrupamiento lineal desarrollado en (De La Vega & Lueker, 1981) para obtener sólo un número constante m de artículos de diferentes tamaños $s_1 > s_2 > \dots > s_m \geq \delta$
- 4) Se aplica el algoritmo de Karmarkar y Karp para obtener una solución aproximada para el BPP sin conflictos y sólo para los artículos grandes (Karmarkar & Karp, 1982).
- 5) Se aplica un algoritmo para modificar la solución para cada tipo de lote de manera que cada recipiente contenga sólo un conjunto independiente de artículos grandes.

- 6) Se aplica finalmente un algoritmo para asignar los artículos pequeños que se habían eliminado en el paso 2, usando nuevos recipientes sólo si es necesario.

4.4.2.3 Heurísticos basados en coloreado de grafos

Los heurísticos de este apartado son propuestos en (Gendreau, et al., 2004). A partir de un grafo con conflictos G , se realiza un coloreado de los vértices V mediante un heurístico de saturación gradual propuesto en (Brélaz, 1979). Este heurístico realiza una partición $\{C,U\}$ de V donde C es el conjunto de vértices coloreados y U el de los no coloreados. Los pasos del algoritmo son los siguientes:

- 1) Se inicializan $C:=0$, y $U:=V$. Se ordenan los vértices por orden decreciente de grado.
- 2) Identificar un vértice $v \in U$ de grado máximo y asignarle color 1.
 $C:=C \cup \{v\}$ $U:=U \setminus \{v\}$
- 3) Elegir un vértice $v \in U$ con máximo grado de saturación, es decir, con máximo número de colores a los cuales es adyacente entre los vértices de C . En caso de igualdad, elegir el vértice de grado máximo.
- 4) Asignar a v el menor número posible de número de color. $C:=C \cup \{v\}$ $U:=U \setminus \{v\}$
- 5) Si $C = V$ parar. En caso contrario, volver al paso 3.

Este algoritmo tiene un tiempo computacional de $O(n^2 \log n)$ (Gendreau, et al., 2004).

Los algoritmos basados en coloreado de grafos que utilizan el algoritmo de saturación gradual se muestran a continuación.

Coloreado y FFD

Este algoritmo presenta los siguientes pasos:

- 1) Ejecutar el algoritmo de saturación gradual
- 2) Resolver un BPP por separado de los artículos de cada color utilizando First Fit Decreasing.

La complejidad de este algoritmo es $O(n^2 \log n)$ (Gendreau, et al., 2004).

Coloreado parcial y FFD

Este heurístico es similar al anterior, exceptuando que inicialmente se eliminan los artículos con 0 conflictos y con 1 sólo conflicto. Estos artículos son asignados después a recipientes utilizando el FFD modificado comentado anteriormente. Los pasos del algoritmo, pues, son:

- 1) Eliminar de V el subconjunto Z de vértices que tienen un grado de 0 o 1 en G .
- 2) Aplicar el algoritmo anterior (coloreado y FFD).
- 3) Asignar los artículos de Z a los recipientes recientemente creados, y a nuevos recipientes mediante el algoritmo FFD modificado

La complejidad de este algoritmo es $O(n^2 \log n)$ (Gendreau, et al., 2004).

Coloreado y FFD iterado

Este algoritmo sigue los siguientes pasos:

- 1) Colorear los vértices de V utilizando el algoritmo de saturación.
- 2) Extraer de V el mayor subconjunto P de vértices que tengan el mismo color. Si $|P| \neq 1$, ir al paso 3. De lo contrario asignar cada artículo de V a un recipiente diferente y parar.
- 3) Resolver un BPP en P usando FFD. Identificar el artículo $h \in P$ con el menor peso y definir el conjunto Q de todos los artículos pertenecientes a los recipientes cuya capacidad residual sea mayor que w_h , excepto aquellos que pertenecen al lote más lleno.
- 4) $V := (V \setminus P) \cup Q$. Si $V \neq \emptyset$ ir al paso 1.

La complejidad del paso 1 es de nuevo $O(n^2 \log n)$, el paso 3 tiene una complejidad de $O(n \log n)$ y se aplica como mucho n^2 veces. La complejidad general de este algoritmo por lo tanto es $O(n^3 \log n)$ (Gendreau, et al., 2004).

4.4.2.4 Heurísticos basados en cliques

Estos heurísticos se basan en realizar sub-grafos en los que cada vértice está conectado a cada otro vértice del sub-grafo (cliques), es decir, donde todos los vértices son adyacentes. Gendreau et al. presentan 2 heurísticos que realizan cliques tanto en el grafo sin conflictos $G' = (V, E')$ como en el grafo con conflictos $G = (V, E)$. En G' , un arco $(i, j) \in E'$ existe si y sólo si i y j no presentan conflicto, y $w_i + w_j \leq c$. Los cliques se determinan mediante el heurístico voraz de Johnson, que tiene el siguiente funcionamiento (Johnson, 1974):

Este algoritmo mantiene una partición $\{C, U\}$ de V donde C es el conjunto de vértices incluidos en el clique y U es el conjunto de vértices todavía no incluidos en el clique. Inicialmente C contiene un vértice de grado máximo. En cada paso, el procedimiento transfiere a C un vértice de U conectado a todos los vértices de C y con el máximo grado. Este algoritmo tiene complejidad $O(n \log n)$.

A continuación se muestran los dos heurísticos basados en cliques desarrollados hasta el momento (Gendreau, et al., 2004).

Clique en G' y FFD

El algoritmo sigue los siguientes pasos:

- 1) Determinar un clique grande D en G' utilizando el algoritmo de Johnson. Si $|D| \neq 1$, ir al paso 2. De otra manera asignar cada artículo de V a un lote diferente y parar.
- 2) Resolver un BPP en el clique D utilizando FFD. Identificar el artículo $h \in D$ con el menor peso, y definir el conjunto Q de todos los artículos que pertenecen a lotes cuya capacidad residual es mayor que w_h , exceptuando aquellos que pertenecen al recipiente más lleno.
- 3) $V := (V \setminus D) \cup Q$. Si $V \neq \emptyset$ ir al paso 1.

La complejidad de los pasos 1 y 2 es $O(n \log n)$. Ya que cada uno de estos pasos puede ser ejecutado como mucho n veces, la complejidad total es $O(n^2 \log n)$.

Clique en G y G' , y FFD

Este heurístico se basa en eliminar primero los artículos que no están en conflictos con todo el resto. Luego se determina un clique D de artículos en conflicto. Cada artículo i de D es

considerado para realizar un clique D_i de artículos que no están en conflicto. Por último, resolviendo un BPP, se identifica un subconjunto de D_i que puede ser asignado con i en el mismo recipiente. Los pasos del algoritmo pues, son:

- 1) Determinar el conjunto H de artículos que no están en conflicto con ningún artículo de V , y definir el conjunto $V:=V \setminus H$.
- 2) Utilizando el heurístico de Johnson, determinar un clique D en el grafo con conflictos $G = (V,E)$, y definir $V:=V \setminus D$. Si $V=0$ parar.
- 3) Para cada $i \in D$ realizar un clique D_i , estando contenido i en el grafo sin conflictos G'_i inducido por $V \cup \{i\}$, utilizando el heurístico de Johnson. Utilizar FFD para resolver el BPP en D_i . Eliminar de V todos los artículos incluidos en el lote que contiene a i , y definir $D:=D \setminus \{i\}$. Si $V=0$ ir al paso 4. Si $D=0$ ir al paso 2. De otra manera, repetir el paso 3.
- 4) Asignar los artículos de H a los lotes existentes, y posiblemente a algunos nuevos, utilizando FFD.

La complejidad del paso 1 es $O(n^2)$, la del paso 2 es $O(n \log n)$, mientras que el paso 3, que es realizado como mucho n veces, tiene una complejidad de $O(n \log n)$. Por lo tanto, la complejidad total es $O(n^2 \log n)$.

Subconjuntos de máxima estabilidad

Este heurístico, propuesto en (Maiza & Radjef, 2011), se basa en el agrupamiento de subconjuntos de máxima estabilidad en recipientes, deducidos iterativamente del grafo de conflictos. El principio es convertir el problema de conflictos a una serie de sub-problemas sin conflictos, de manera similar a los procedimientos expuestos anteriormente propuestos por Gendreau et al. El heurístico sigue los siguientes pasos:

- 1) Se define U como un conjunto de artículos no asignados, con grado positivo en el grafo de conflictos, y se define F como un conjunto de artículos libres (artículos con grado 0 en el grafo de conflictos).
- 2) Se define G como un grafo de conflictos determinado sólo por un conjunto U de artículos. Se define un conjunto V_k de máximo clique del grafo de conflictos $G(V,E)$ basándose en el heurístico de Johnson anteriormente descrito.

- 3) Para cada artículo i de V_k , se define un conjunto $V'_k(i)$ de máximo clique, conteniendo el artículo i del sub-grafo complementario $G_c(V', E'_c)$, utilizando también el algoritmo de Johnson. Sea S_i el subconjunto resultante.
- 4) Para cada subconjunto S_i , aplicar un procedimiento clásico de asignación del BPP (por ejemplo BFD o FFD). Una vez acabada la asignación eliminar de U todos los artículos asignados al lote que contiene a i .
- 5) Mientras U no esté vacío volver al paso 2. De lo contrario, asignar los artículos del conjunto F a los recipientes existentes, o a nuevos recipientes, utilizando el método FFD.

Este algoritmo presenta una complejidad de $O(n^2 \log n)$, utilizando los métodos BFD o FFD en el paso 4.

4.4.2.5 Heurístico adaptado de menor hueco en lote

Este heurístico (adapted minimum bin slack) propuesto en (Maiza & Radjef, 2011), basándose en el heurístico similar propuesto por Gupta y Ho para el BPP (Gupta & Ho, 1999). El funcionamiento del heurístico es intentar en cada paso encontrar un conjunto de artículos de manera que la capacidad del recipiente se llene al máximo posible. En cada etapa, se mantiene una lista l' de n' artículos no asignada a ningún recipiente hasta el momento, ordenada en orden decreciente de tamaño. Cada vez que se asigna algún elemento a un recipiente, se elimina de la lista l' , manteniendo el orden. El proceso empieza con $l'=l$ y acaba cuando la lista se vacía. Cada asignación se determina con un procedimiento de búsqueda que prueba todos los posibles subconjuntos de artículos l' de manera que se llena al máximo posible la capacidad de un recipiente, eligiendo aquellos que dejan el menor hueco posible. La búsqueda comienza por aquellos elementos de mayor tamaño.

Para el caso del problema con conflictos, la adaptación del MBS conlleva los siguientes pasos:

- 1) Ordenar los artículos en una lista l' en orden decreciente de tamaño.
- 2) Mientras haya artículos en l' :
 - Aplicar el MBS en l' , testeando la compatibilidad del artículo actual con los artículos ya considerados, buscando el mejor subconjunto.

- Eliminar el subconjunto encontrado de I' .

La complejidad de este algoritmo es $O(2n)$, siendo de $O(u^{n+1})$ si u es el número máximo de artículos que pueden ser asignados en un recipiente.

4.4.2.6 Branch-and-price

Existen una serie de algoritmos basados en el método de branch-and-price para resolver el problema.

En (Muritiba, et al., 2010) se usan límites inferiores y un heurístico de búsqueda tabú como un paso de pre-procesado. La parte de precios del algoritmo branch-and-price se resuelve con un heurístico voraz. El solver MIP de CPLEX 10 es llamado cuando el heurístico no puede encontrar una columna con un coste reducido negativo. La parte de ramificación se realiza mediante una restricción disyuntiva en la variable λ con mayor parte fraccional.

Elhedhli et al. utilizan la regla de ramificación en el que dos elementos se seleccionan y se asignan al mismo recipiente en el primer nodo hijo, y restringidos a estar en diferentes lotes en el otro nodo hijo. Para la estrategia de implementación, en lugar de utilizar un árbol de búsqueda binario, se consideran varios pares de artículos simultáneamente (Elhedhli, et al., 2011). El problema de precios se resuelve con el solver MIP de CPLEX tras añadir desigualdades de máximo clique generados utilizando la librería Qualex (Busygin, 2006).

En (Sadykov & Vanderbeck, 2013) los autores se basan en estos dos algoritmos de Curitiba y Elhedhli et al., con las diferencias principales de que el problema de precios se resuelve mediante un algoritmo para resolver el problema de la mochila con conflictos, la regla de ramificación utilizada es distinta, y se utiliza un heurístico basado en generación de columnas..

4.4.2.7 Heurístico de aproximación voraz

Se incluye en este apartado un caso concreto de BPPC, determinado en (Beaumont, et al., 2010), donde los autores resuelven el denominado problema de empaquetado en recipientes bajo restricción de distancia – Bin Packing under Distance Constraint (BPDC), donde los elementos pertenecen a un espacio métrico, y la distancia mínima entre dos elementos en

un lote ha de ser menor que un valor d_{\max} . Al ser equivalente al BPPC es NP-completo. Para resolver el problema, los autores utilizan un algoritmo de aproximación voraz, basado en el trabajo de Epstein y Levin para el BPPC, utilizando el FFD para construir los lotes (Epstein, et al., 2011). Los pasos del algoritmo son los siguientes:

- 1) Definir U como el conjunto de elementos no asignados
- 2) Se define el grafo de compatibilidad C , asociado a la instancia I con la distancia máxima d_{\max} $C = \text{Com}(I, d_{\max})$
- 3) Mientras haya un conjunto de 3 elementos conectados $\{a,b,c\}$ que puedan encajar en un lote, de manera que el peso de cada uno sea menor que la mitad del lote, y la suma de sus pesos extendidos sea mayor que el tamaño del lote, o un par de elementos conectados (a, b) de manera que entren en un lote:
 - a. Elegir un conjunto de máximo peso extendido, y colocar los elementos de este conjunto en un nuevo lote
 - b. Eliminar los elementos de U
- 4) Construir una partición M de U de manera que el diámetro entre elementos sea como mucho $2d_{\max}$
- 5) Aplicar First-Fit-Decreasing a cada conjunto de M partes

4.4.3 Conclusiones extraídas de la literatura

Debido a la naturaleza del problema y a su complejidad intrínseca, todos los algoritmos analizados son heurísticos o meta-heurísticos sencillos que tratan de acercarse lo máximo posible a la solución óptima. Se emplea a menudo la estrategia de dividir el problema en problemas más pequeños sin conflictos para luego aplicar alguno de los heurísticos clásicos descritos anteriormente. La complejidad computacional de estos algoritmos es siempre $O(n^2 \log n)$ o $O(n^2)$.

En el caso del problema concreto que se quiere analizar en esta tesis, la complejidad aumenta todavía más, ya que al problema con conflictos se le añaden las restricciones anteriormente comentadas del mínimo peso que deben de tener todos los lotes, y de buscar

lotes lo más homogéneos posibles. En cuanto a los datos con los que se lanzará el algoritmo, cada bobina puede tener diferentes pesos, con lo que normalmente existirá un número de pesos similar al de bobinas, aumentando aún más la complejidad.

Además de estos factores, como se ha comentado anteriormente la capacidad de lanzar diferentes optimizaciones en un tiempo muy corto hace necesario el uso de técnicas algorítmicas que permitan encontrar una solución lo mejor posible en un tiempo limitado. La idea de utilizar alguna técnica de clusterización para tratar de agrupar elementos similares, o de separar de antemano aquellos muy diferentes se plantea como una buena aproximación. Por este motivo, en la siguiente sección se realiza un estudio sobre diferentes técnicas de clusterización, para conocer las diferentes posibilidades y estrategias existentes antes de proponer una solución al problema.

5 Estudio previo sobre clusterización

Los algoritmos de clustering agrupan elementos que son parecidos entre sí, y los separan de los que son diferentes, basándose en alguna característica para diferenciarlos (Duran & Odell, 2013). Estas características parecen de gran utilidad para resolver problemas como el que se plantea en esta tesis; por ello, en este apartado se resumen las principales técnicas de clusterización. El conocimiento de estas técnicas servirá de base para proponer ideas de cara a una posible agregación de elementos similares / separación de elementos diferentes. Se describen primero las medidas más usuales que se aplican en cualquier tipo de clúster, para a continuación exponer las diferentes técnicas.

5.1 Medidas de distancia y similitud

En este apartado se describen las principales aproximaciones para medir la similitud / disimilitud entre dos objetos de un clúster o entre varios clústeres. Un objeto de datos se describe mediante un conjunto de características, normalmente representados en un vector multidimensional. Se puede definir entonces una función sobre el conjunto de datos X de manera que satisfaga las siguientes condiciones:

- 1) Simetría: $D(x_i, x_j) = D(x_j, x_i)$
- 2) Positividad: $D(x_i, x_j) \geq 0$, para todo x_i y x_j .

Además, se denomina una métrica si se cumplen las siguientes condiciones:

- 3) Desigualdad triangular: $D(x_i, x_j) \leq D(x_j, x_k) + D(x_k, x_i)$
- 4) Reflexividad: $D(x_i, x_i) = 0$ si $x_i = x_i$

Se puede definir entonces, para un conjunto de datos con N patrones de entrada, una matriz $N \times N$ simétrica, denominada matriz de proximidad, cuyos elementos (i, j) representan la medida de similaridad o diferencia para los patrones i, j .

Típicamente, las funciones de distancia se utilizan para medir condiciones continuas, mientras que las medidas de similaridad son más importantes para variables cualitativas. En la

Tabla 4 se muestran típicas medidas que se pueden utilizar para comparar elementos. La selección de la medida a utilizar se ha de realizar atendiendo a cada problema concreto (Xu & Wunsch, 2005). Generalmente, para objetos que consisten en variables de diferente tipo, es común normalizar las variables al intervalo (0, 1) y usar la medida euclídea. Alternativamente, se pueden transformar en variables binarias y utilizar funciones de similitud.

Medida	Fórmula	Comentarios
Distancia de Minkowski	$D_{ij} = \left(\sum_{l=1}^d x_{il} - x_{jl} ^{1/n} \right)^n$	Métrica. Invariante a traslación y rotación si $n=2$. Las características con mayores valores y tendencias dominan.
Distancia Euclídea	$D_{ij} = \left(\sum_{l=1}^d x_{il} - x_{jl} ^{1/2} \right)^2$	Métrica más usual (Minkowski con $n=2$). Tiende a formar clústers hiper-esféricos.
City-block	$D_{ij} = \sum_{l=1}^d x_{il} - x_{jl} $	Minkowski con $n=1$. Tiende a formar clústers hiper-rectangulares
Chebichev	$D_{ij} = \max x_{il} - x_{jl} $	Minkowski con $n \rightarrow \infty$. Es el máximo de las diferencias absolutas en todas las coordenadas.
Mahalanois	$D_{ij} = (x_i - x_j)^T S^{-1} (x_i - x_j)$ S: matriz de covarianza	Tiende a formar clústers hiper-elipsoidales. Mayor carga computacional.
Correlación de Pearson	$D_{ij} = (1-r_{ij})/2$, siendo $r_{ij} = \frac{\sum_{l=1}^d (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{\sum_{l=1}^d (x_{il} - \bar{x}_i)^2 (x_{jl} - \bar{x}_j)^2}}$	No es una métrica. Derivada del coeficiente de correlación, no es capaz de detectar la magnitud de diferencia.
Punto de simetría	$D_{ir} = \min \frac{\ (x_i - x_r) + (x_j - x_r)\ }{\ (x_i - x_r)\ + \ (x_j - x_r)\ }$	No es una métrica. Calcula la distancia entre un objeto x y un punto de referencia x_r .
Similaridad del coseno	$S_{ij} = \cos \alpha = \frac{x_i^T x_j}{\ x_i\ \ x_j\ }$	Independiente de la longitud del vector. Invariante a rotaciones, pero no a transformaciones lineales.

Tabla 4. Medidas de similitud más usuales

5.2 Técnicas de clusterización

A continuación se describen las principales técnicas de clusterización. La clasificación aquí propuesta está basada en diferentes trabajos de la literatura, principalmente (Xu & Wunsch, 2005) y (Wong, 2015).

5.2.1 Clusterización jerárquica

Los algoritmos de clusterización jerárquica (HC) organizan los datos en una estructura jerárquica acorde a la matriz de proximidad. Sus resultados se suelen representar mediante un árbol binario, con el nodo raíz representando el conjunto total de objetos, y en cada nodo hoja un objeto dato; los nodos intermedios, por lo tanto, representan la amplitud en la que los objetos están unos próximos a otros, mientras que la altura del árbol representa la distancia entre cada par de objetos / clúster. Los resultados últimos del clúster pueden obtenerse cortando el árbol en diferentes niveles.

Los algoritmos de clusterización jerárquica pueden dividirse en 2 métodos: de división y de aglomeración. En los clústeres de división, al principio, todo el conjunto de datos pertenece a un clúster, y un procedimiento sucesivamente lo divide hasta que todos los clústeres son individuales. Este procedimiento, basado en estructura jerárquica es muy complejo computacionalmente hablando, ya que para un conjunto de N elementos, existen $2^{N-1}-1$ posibles divisiones, por ello ha sido menos utilizado en la literatura. Se pueden encontrar dos ejemplos de estos algoritmos, denominados MONA y DIANA, en (Kaufman & Rousseeuw, 2009).

Los algoritmos de aglomeración, contrariamente, comienzan con N clústeres, incluyendo cada uno de ellos un objeto. A continuación se realizan una serie de operaciones de mezcla que tienen como resultado el agrupamiento de objetos en el mismo grupo. El procedimiento general de un algoritmo de clustering jerárquico de aglomeración es el siguiente:

- 1) Comenzar con N clústeres de 1 sólo elemento. Calcular la matriz de proximidad para los N clústeres.

- 2) Buscar la distancia mínima

$$D(C_i, C_j) = \min_{\substack{l \leq m, l \leq N \\ l \neq m}} D(C_m, C_l)$$

Siendo D la función de distancia en la matriz de proximidad, que combina los clústeres i, j para formar un nuevo clúster.

- 3) Actualizar la matriz de proximidad calculando las distancias entre los nuevos clústeres formados y el resto de clústeres. Para calcular las distancias se suele utilizar alguna de las funciones listadas en el apartado anterior.
- 4) Repetir los pasos 2 y 3 hasta que todos los objetos estén el mismo clúster.

El principal problema que presentan los algoritmos de clusterización jerárquica es su falta de robustez, debido a que son sensibles a ruido y outliers (Xu & Wunsch, 2005). Una vez que un elemento ha sido asignado a un clúster no se vuelve a considerar, con lo que clasificaciones erróneas no pueden corregirse. La complejidad computacional de estos algoritmos es al menos $O(N^2)$, con lo que su aplicación es limitada con grandes volúmenes de datos.

5.2.2 Clusterización basada en el error cuadrático (cuantificación vectorial)

En contraste con la clusterización jerárquica, la clusterización particional se basa en asignar un conjunto de objetos en clústeres sin estructura jerárquica. En principio, la partición óptima basada en algún criterio específico podría encontrarse enumerando todas las posibilidades; sin embargo, este método de fuerza bruta no es factible en la práctica, debido a su complejidad computacional (Liu, 1968). Incluso para un pequeño problema de clusterización como organizar 30 objetos en 3 grupos, el número de posibles particiones es 2×10^{14} . Debido a esto, se han desarrollado algoritmos heurísticos para buscar una solución aproximada. Uno de los criterios más utilizados es el error cuadrático. Dentro de los algoritmos basados en este criterio, el más conocido es el denominado K-means, propuesto en (Lloyd, 1982), cuyos pasos son:

- 1) Dividir el conjunto inicial en K subconjuntos, con K centroides

- 2) Calcular la distancia de cada elemento a cada centroide, y asignar los elementos al clúster con el centroide más cercano
- 3) Recalcular los centroides de cada subconjunto con los nuevos elementos asignados
- 4) Repetir los pasos 2 y 3 hasta que no exista ningún cambio de elemento en clúster

Las principales ventajas de este algoritmo son su sencillez y versatilidad de aplicación en diferentes problemas. La complejidad del algoritmo es $O(N K d)$. Se puede utilizar con volúmenes de datos relativamente grandes y existen versiones del algoritmo paralelizadas que reducen el tiempo de ejecución. Sin embargo, también presenta desventajas:

- No existe un método universal eficiente para identificar las particiones iniciales y el número de clústeres.
- El procedimiento iterativo del algoritmo no garantiza que se alcance una solución óptima.
- El algoritmo es muy sensible al ruido y a outliers.
- La utilización de medias limita el algoritmo a variables numéricas.

En (Karypis, et al., 1999) se presenta un algoritmo denominado Chameleon, basado en el grafo de k -vecinos más cercanos, en el cual un arco se elimina si ambos vértices no está dentro de los k puntos más cercanos entre ellos. En un primer paso, se divide el grafo de conectividad en un subconjunto de clúster con la mínima distancia. Combinando la interconectividad relativa y la cercanía, el algoritmo junta los pequeños subconjuntos para proporcionar los clústeres finales. La cercanía se obtiene normalizando la suma de los pesos de los arcos que conectan dos clústeres. Cheng and Lo construyen un hiper-grafo (cada arco puede conectar más de 2 vértices), y utilizan un algoritmo en 2 fases, similar al Chameleon para encontrar clústeres (Cherng & Lo, 2001).

En (Zhu, 2016) el autor presenta un algoritmo para realizar clústeres basados en similitud de 2 atributos, a través de una unidad de medida unificada. Así, se particiona un grafo original en K sub-grafos de manera que cada clúster contenga un sub-grafo densamente conectado con valores de los atributos homogéneos. Para ello, definen una medida de distancia considerando ambos atributos, siendo posible asignarles un peso a los diferentes atributos. Esta idea parece interesante para aplicar en el algoritmo de resolución del problema

de esta tesis, ya que es una manera elegante y funcional de poder agrupar elementos con distintas funcionalidades de forma homogénea. Una vez definida esta medida, en este caso los autores utilizan un clúster basado en K-means.

5.2.3 Clusterización difusa

En todas las técnicas presentadas hasta ahora, cada objeto es asignado solamente a un clúster. En la clusterización difusa, esta restricción es relajada, con lo que un mismo objeto puede pertenecer a varios clústeres, con un cierto grado de membresía (Sakawa, 2013). Este concepto es particularmente útil cuando los límites entre los clústeres son ambiguos, y no están bien separados. El algoritmo más representativo de este tipo es el denominado Fuzzy c-means (FCM) (Bezdek, 2013), desarrollado para superar dichos inconvenientes.

El algoritmo asigna a cada elemento un valor de pertenencia dentro de cada clúster, con lo cual un mismo objeto puede estar en varios clústeres de manera simultánea. A diferencia del k-means, este algoritmo realiza una partición suave del conjunto, de manera que todos los elementos pertenecen en algún grado a todos los clústers. Los pasos son:

- Elegir un número de clústeres
- Asignar aleatoriamente a cada elemento coeficientes por estar asignado a un clúster
- Repetir hasta que el algoritmo haya convergido (hasta que los centroides no cambien)
 - Calcular el centroide para cada clúster
 - Recalcular la partición → para cada elemento, calcular sus coeficientes por pertenecer a cada clúster

El centroide se calcula como la media de todos los elementos de un clúster, ponderado por su grado de pertenencia al mismo.

Al igual que el algoritmo k-means, este algoritmo también es afectado en gran medida por el ruido proveniente de outliers, y la dificultad de identificar las particiones iniciales.

Las teorías de conjuntos difusos pueden ser utilizadas también para crear un clúster de estructura jerárquica. En (Geva, 1999) se propone un algoritmo difuso jerárquico, que puede

explorar efectivamente estructuras de datos a diferentes niveles, de manera similar a los algoritmos jerárquicos, a la vez que establece las conexiones entre cada objeto y clúster en la jerarquía con la membresía. Este diseño hace que el algoritmo elimine una de las principales desventajas de los jerárquicos, siendo capaz de reasignar un objeto una vez se había asignado previamente a otro cluster.

5.2.4 Clusterización basada en técnicas de búsqueda combinatoria

El problema se define como organizar un conjunto de puntos en K clústeres, atendiendo a alguna función objetivo. Incluso para un problema de pequeño tamaño, la complejidad computacional es extremadamente grande. Las técnicas de búsqueda local simples, como algoritmos ascendentes, se utilizan para encontrar particiones, aunque normalmente se quedan atascados en óptimos locales. Diferentes técnicas y metaheurísticas se pueden utilizar para alcanzar mejores soluciones: algoritmos evolutivos, búsqueda tabú, recocido simulado, algoritmos basados en comportamientos de la naturaleza (ACO, SWARM), etc.

El uso de técnicas heurísticas y meta-heurísticas suele proporcionar buenas soluciones en tiempos razonables; la principal desventaja que presentan es la selección de parámetros adecuados que orienten la técnica de búsqueda hacia la parte del espacio que presenta mejores soluciones.

5.2.5 Clusterización basada en grafos

Los conceptos y propiedades de la teoría de grafos hacen que sea muy conveniente describir problemas de clusterización basados en los mismos. Nodos V de un grafo G se corresponden con puntos de datos en el espacio, mientras que los arcos E representan la proximidad entre cada par de puntos. Se pueden realizar clusterizaciones buscando los subgrafos máximamente conectados, o buscando subgrafos completos (cliques). Esta representación encaja muy bien con el problema a resolver en la tesis, con lo que se analizarán diferentes algoritmos propuestos de cara a realizar una partición inicial de nuestros productos en sub-conjuntos de productos similares.

Existen varias técnicas en la literatura utilizadas para particionar / clusterizar grafos. Se resumen a continuación los enfoques principales, basados en la clasificación propuesta en

(Schulz, 2015). Se ha completado esta clasificación con otros ejemplos interesantes encontrados que realizan clusterizaciones para problemas que pueden presentar cierta similitud al planteado en la tesis en otros ámbitos.

Programación lineal entera

Existen ejemplos de particionado de grafos utilizando programación lineal entera, cuando el tamaño del grafo y el tiempo de computación lo permiten. Gorke presenta un algoritmo ILP, utilizando la modularidad como función de coste (Gorke, 2010). De la misma manera, Brillout presenta un algoritmo de partición para optimizar los arcos que deben eliminarse del clúster inicial para ir dividiéndolo en clústeres más pequeños acorde a algún objetivo (Brillout, 2009).

Para hacerse una idea del rendimiento computacional con programación lineal entera, una optimización del algoritmo de Gorke con un grafo de 300 vértices puede tardar hasta un día. Es necesario pues, buscar otro tipo de técnicas para el problema planteado en la tesis.

Agrupamiento espectral

Esta técnica parte de una matriz de similitud entre los diferentes datos (vértices del grafo), y hace uso de los valores propios de esta matriz para reducir la dimensionalidad de la misma. En contraste con las aproximaciones de búsqueda local, se centra en resolver el problema de manera global. El algoritmo clásico propuesto en (Fiedler, 1975), y en el que se basaron (Hendrickson & Leland, 1995) y (Simon, 1991) es el siguiente:

- Construir la matriz Laplaciana L del grafo G
- Calcular el vector propio x de los valores propios no triviales más pequeños
- Calcular la media de los valores del vector x
- Realizar una partición $(V_1, V \setminus V_1)$, de manera que

$$\forall v \in V: v \in V_1 \leftrightarrow \tilde{x}_v = 1$$

$$\tilde{x}_i := \begin{cases} 1, & x_i < \tilde{m} \\ -1, & x_i \geq \tilde{m} \end{cases}$$

El rendimiento y los resultados obtenidos con estos algoritmos suelen mejorar a los clásicos algoritmos como el k-means. Se comportan particularmente bien para dividir un grafo inicial en 2 subgrafos, pero a medida que el número de subgrafos aumenta el rendimiento empeora. El número de clústeres a considerar ha de ser fijo, con lo que para el problema que se plantea en la tesis se podría utilizar una técnica similar de manera recursiva.

Aglomeraciones voraces

Los algoritmos voraces para clusterizar se basan en ir agrupando los elementos en clústeres de manera iterativa, hasta que sólo se haya creado un único clúster con todos, para después repasar todos los clústeres generados a lo largo del proceso, y quedarse con los mejores grupos en función del objetivo. Para describir la estructura de vecinos de la red se utiliza un dendrograma, representando en las hojas los vértices de la red original, y en los nodos internos las uniones realizadas. Dos métodos son los que destacan: el de Louvain y el de árbol de mínimo corte.

El algoritmo de Louvain se basa en optimizar la modularidad, definida como un valor Q (-1, 1) que mide la densidad de enlaces dentro de un clúster, comparado con los enlaces entre clústers (Waltman & van Eck, 2013). Se define como:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i c_j), \text{ donde}$$

A_{ij} representa el peso (distancia) del arco entre los nodos i y j .

k_i y k_j son la suma de los pesos de los arcos de los nodos i y j , respectivamente.

m es la suma de todos los pesos de los arcos del grafo

c_i y c_j son los clústeres de los nodos,

δ : función delta

El algoritmo asigna cada nodo de la red a su propio clúster, y después se cruzan en orden aleatorio y se mueven al clúster vecino de manera que se incremente la modularidad, repitiendo este proceso hasta el final.

Los algoritmos basados en árboles de mínimo corte se basan en agrupar en clústeres elementos cercanos entre sí, tratando de que la distancia entre los diferentes clústeres sea grande (Boykov & Kolmogorov, 2004). Para ello, van buscando sucesivos árboles de mínimo coste, para elegir el arco por donde cortar. El algoritmo genérico tiene los siguientes pasos:

- Partir del grafo original G
- Añadir un nodo artificial t con arcos de capacidad α que se unen al resto de nodos
- Calcular el árbol de mínimo corte
- Borrar t y los arcos añadidos

Partiendo de esta idea, en (Jain & Dubes, 1988) se propone un algoritmo que busca componentes conectados como clústeres detectando y descartando arcos inconsistentes en el mínimo spanning tree. Por su parte, en (Hartuv & Shamir, 2000) los autores tratan los clústeres como sub-grafos de gran conectividad, donde gran conectividad significa que la conectividad (número mínimo de arcos necesarios para desconectar un grafo) del sub-grafo es como mucho la mitad que el número de vértices. El algoritmo aplica un procedimiento de mínimo corte para separar un grafo con un número mínimo de arcos de manera recursiva.

Otras técnicas

Se incluyen en este apartado otros algoritmos basados en grafos que no encajan en ninguno de los tipos analizados, pero que se consideran interesantes por su enfoque y por la cierta similitud del problema con el propuesto en la tesis, aunque en ámbitos totalmente diferentes.

Un ejemplo de estos algoritmos es el propuesto por Kawaji et al. para la caracterización de proteínas, siendo necesario agrupar aquellas que tienen similitud regional a lo largo de sus secuencias (Kawaji, et al., 2004). Para ello los autores se apoyan en una representación de grafo, donde sus vértices representan las proteínas, y los arcos la similitud entre las mismas. El algoritmo particiona el grafo de manera repetitiva eliminando arcos con pesos pequeños, que se corresponden a aquellos que tienen poca similitud. Esta aproximación parece una buena base para proponer un algoritmo que según el artículo proporciona buenos resultados (76% de proteínas agrupadas en clústeres). Para evitar grandes tiempos

computacionales, los autores incluyen un límite por encima del cual se considera que las proteínas son suficientemente parecidas.

En otro ámbito diferente, en (Alvisi & Franchini, 2014) se propone un heurístico para la creación de distritos de áreas de medidas en un sistema de distribución de agua. Para ello, primero generan una serie de soluciones del problema, basándose en recorrer los diferentes nodos de la red de agua, para después elegir de estas soluciones aquellas que presentan mejores resultados desde el punto de vista hidráulico, y finalmente realizan una simulación y selección final de la mejor solución encontrada.

Como conclusión del estudio de posibles técnicas a aplicar para realizar una primera clusterización de elementos basándose en la homogeneidad de los mismos, se observa que el enfoque en grafos parece una buena opción por la facilidad de representación del problema, y por la rapidez a la hora de aplicar esta técnica para la generación de clústeres de componentes similares. En concreto, las ideas de Kawaji (Kawaji, et al., 2004) para la agrupación de las proteínas, y de Zhou (Zhou, et al., 2016) para particionar un grafo en función de atributos homogéneos con diferentes pesos parecen buenas alternativas. Se ha decidido plantear una solución para el problema a resolver en esta tesis basándose tanto en estas ideas de clusterización como en los algoritmos utilizados para resolver el BPP y el BPPC analizados en los apartados 4.3 y 4.4. En el siguiente apartado se desarrolla en profundidad la solución y los algoritmos propuestos.

En el caso real a resolver en la tesis, centrado en el mundo del acero, los recipientes se denominan lotes, con lo que también se utilizará esta nomenclatura en los siguientes apartados.

6 Solución propuesta

En este apartado se desarrolla la solución propuesta para resolver el problema planteado en la tesis. Se ha desarrollado un algoritmo consistente en varias fases, cada una de las cuales se explica a continuación. Previamente se define el problema a analizar desde el punto de vista matemático.

6.1 Problema de Empaquetado de Lotes Homogéneos con Mínimo Peso

El problema real que se desea resolver es bastante más complejo que los analizados y desarrollados hasta ahora en la literatura. Las variantes que incorpora con respecto a la descripción clásica del “Bin packing problem” son:

- No se pueden combinar en un lote productos que tengan diferentes parámetros de planta, localización, forma, categoría, familia (con las excepciones comentadas en el capítulo 2), y número de lados de recubrimiento. Se denominarán a estos parámetros “parámetros globales” de aquí en adelante.
- El tamaño máximo de los lotes depende de la planta de donde se haya fabricado el producto.
- Existe un tamaño mínimo (toneladas) que deben de cumplir todos los lotes.
- Dentro de un lote, los productos deben de ser lo más homogéneos posibles. Para determinar la homogeneidad, no existe un único parámetro como se aborda en casos de la literatura, sino que en este caso la homogeneidad entre productos viene definida por los parámetros: subfamilia, grado de acero, aceitado, peso, ancho, espesor y espesor de recubrimiento. Estos parámetros se denominarán “parámetros locales”.

- Es necesario permitir dar más prioridad a unos parámetros que a otros a la hora de agrupar.

La restricción correspondiente a que los productos tengan los mismos valores para los parámetros globales dentro de un lote se puede resolver realizando una división previa del conjunto inicial de productos, dividiendo el problema en n sub-problemas más pequeños. Cada uno de estos sub-problemas puede verse como un BPP con conflictos (el conflicto entre productos vendrá dado por su heterogeneidad), incluyendo además un tamaño mínimo en cada lote. Por otro lado, el objetivo no es simplemente minimizar el número de lotes, sino que también es que los lotes obtenidos sean lo más homogéneos posibles. Se ha decidido denominar a este problema como Problema de Empaquetado de Lotes Homogéneos con Mínimo Peso – Homogeneous Bin Packing Problem with Minimum Weight (HBPPMW). La descripción matemática del problema se muestra a continuación.

El HBPPMW puede ser modelado introduciendo dos conjuntos de variables binarias:

y_h , con valor 1 si el lote h se usa, y 0 sino

x_{ih} , con valor 1 si el artículo i se asigna al lote h , y 0 sino $| (i= 1, 2, \dots, n) (h= 1, 2, \dots, n)$

De igual manera, se definen:

d_h como la distancia máxima entre los elementos dentro de un lote

D_{max} , como la distancia máxima permitida entre cualquier par de elemento de un lote

w_{min} como el peso mínimo que deben tener los lotes,

u , como el número de elementos que quedan sin asignar

Matemáticamente el problema puede ser expresado de la siguiente forma:

$$\min(\sum_{h=1}^n y_h + u + \sum_{h=1}^n d_h) \quad (1)$$

$$\sum_{h=1}^n x_{ih} = 1 \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n w_i x_{ih} \leq C y_h \quad h = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^n w_i x_{ih} \geq w_{min} \quad h = 1, 2, \dots, n \quad (4)$$

$$d_h \leq Dmax \quad h = 1, 2, \dots, n \quad (5)$$

$$y_h \in \{0, 1\} \quad h = 1, 2, \dots, n \quad (6)$$

$$x_{ih} \in \{0, 1\} \quad i = 1, 2, \dots, n \quad h = 1, 2, \dots, n \quad (7)$$

La función de coste (1) intenta minimizar tanto el número de lotes como la heterogeneidad de los mismos, y dejar el menor número posible de elementos sin agrupar. La restricción (2) garantiza que cada artículo sea asignado a un único lote. Con la restricción (3) se garantiza que no se supere la capacidad máxima de los lotes, mientras que con la (4) se garantiza el peso mínimo necesario que se debe de alcanzar en los mismos. La restricción (5) obliga a que los elementos del lote sean similares (menor que una distancia máxima permitida por lote). Las restricciones (6) y (7) garantizan que las variables x e y tiene valores binarios.

Como se comentaba en el apartado anterior, la versión clásica del problema simple del “Bin Packing Problem” es NP-Hard, con lo que es improbable encontrar algoritmos de optimización que encuentren la solución óptima en tiempos razonables. Este hecho, unido a la complejidad que incorporan las peculiaridades que se abordan en la tesis, junto con el alto número de elementos y restricciones que debe de manejar el algoritmo hacen necesario pensar en alguna metodología que permita obtener soluciones lo más cercanas posibles al óptimo con un buen rendimiento computacional. El número de elementos que el algoritmo debe de agrupar en lotes ronda los 5.000 por semana, siendo la media por planta de unos 200 productos, lo cual es un número muy elevado para el escaso tiempo requerido para ejecutar el algoritmo.

Debido a estos factores, se ha decidido utilizar una estrategia para resolver el problema siguiendo una metodología basada en 3 fases bien diferenciadas, como se muestra en la Figura 12:

- 1) Pre-procesado de datos y división inicial de todos los productos en conjuntos con las mismas características
- 2) Clusterización de productos basándose en la similitud entre los mismos
- 3) Asignación final de productos a lotes

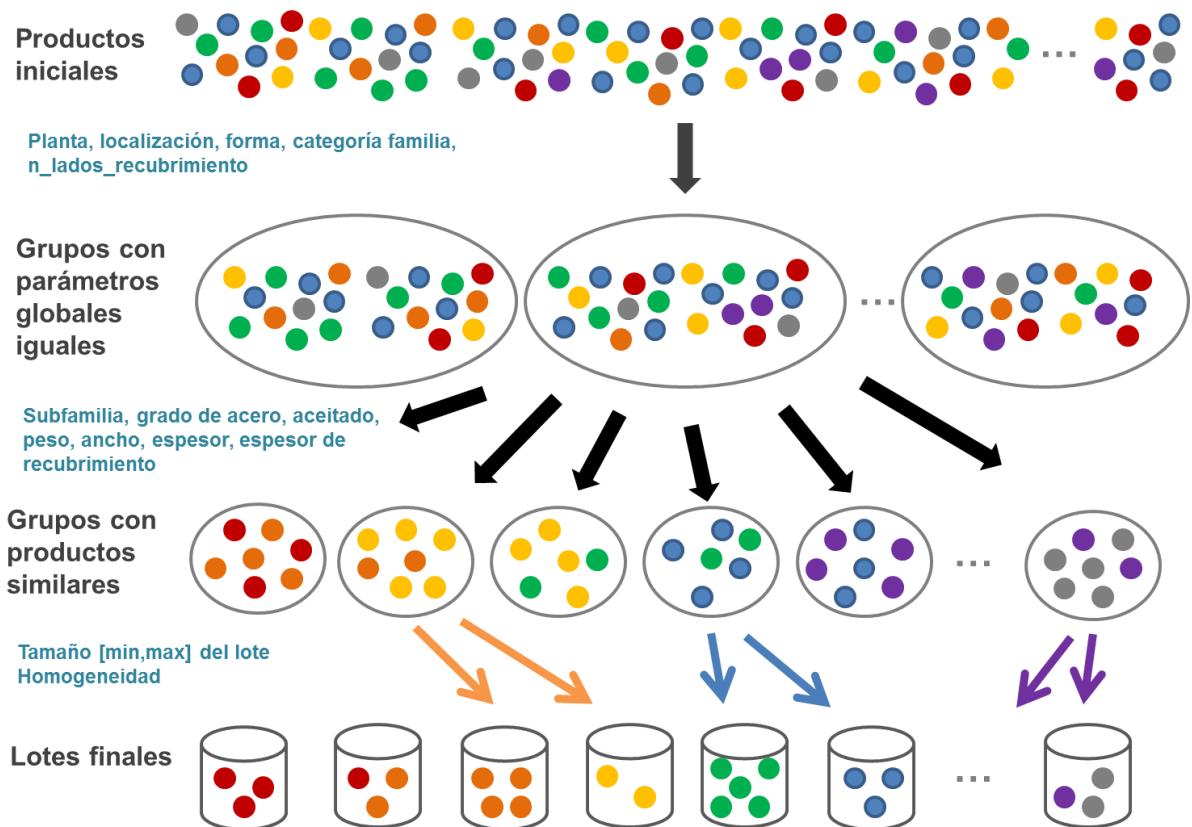


Figura 12. Fases de resolución del problema

Cada uno de los pasos de la estrategia seguida se desarrolla a continuación en profundidad.

6.2 Pre-procesado de datos y división inicial en conjuntos

Los datos de todos los productos de cada planta que se sacan a subasta actualmente se pueden exportar desde la aplicación interna de la empresa a formato Excel. Así pues, para facilitar el uso del algoritmo propuesto, se ha definido el formato Excel como formato de entrada para obtener los datos.

Es necesario realizar un pre-procesado previo de los datos a analizar. Si bien no es objeto de la tesis, es muy importante de cara a garantizar la calidad de los datos a emplear. Es necesario mantener una coherencia al leer la información, para asegurar que algún error tipográfico no se entienda como una nueva característica. Por poner un ejemplo sin revelar datos de las plantas por la confidencialidad de las mismas, puede que el campo "Planta" aparezca como "Planta1", "PLANTA1", o "planta1". Lo mismo puede suceder con las diferentes localizaciones, familias, subfamilias, grados de acero, etc.

Para garantizar la robustez de los datos se ha elaborado un diccionario histórico con todas las posibles variantes que han aparecido en cada campo en los últimos 2 años, y se ha adoptado una normalización interna para evitar que caracteres como mayúsculas o espacios presenten problemas. De igual manera, se ha incluido una parte de comprobación de datos erróneos, para descartar aquellos elementos que debido a un error incumplan alguna de estas características:

- Categoría: ha de ser 11, 1, 2, 3 o 4.
- Número de lados de recubrimiento: el valor ha de ser 0, 1, 2.
- Espesor de recubrimiento: ha de venir expresada la unidad para poder comparar, y la unidad deben de ser g/m² o µm. Todas las medidas se transforman internamente a la misma unidad para que sean comparables, para ello, dado que el recubrimiento se realiza con Zinc, utilizando su densidad (7.14g/cm³) se tiene que:

$$\text{espesor} \left(\frac{g}{m^2} \right) = 7.14 * \text{espesor}(\mu\text{m})$$

- Datos vacíos: todos los campos de cada producto han de tener un valor, exceptuando:
 - Número de lados de recubrimiento: si el campo viene vacío es porque no tiene sentido que ese producto se haya sometido a ningún recubrimiento, por lo que se asume que es cero.
 - Espesor de recubrimiento: si el número de lados de recubrimiento es 0, éste campo podrá ser 0 o vacío.
 - Aceitado: de igual manera, si el campo viene vacío se asume que el producto es “no aceitado”.

Tras leer los datos y comprobar su coherencia, se procede a realizar una división inicial en conjuntos, de manera que todos los elementos de cada conjunto tengan los mismos parámetros globales (y considerando aquellas sub-familias que no se pueden mezclar entre sí). Este primer paso garantiza que estas restricciones se cumplan en los lotes finales, y elimina complejidad algorítmica al disminuir el número de parámetros que se deben de considerar al asignar los productos a lotes. A pesar de que se está dividiendo el problema en varios subconjuntos que habrá que optimizar cada uno de ellos por separado se considera que dado el número de factores que se deben de tener en cuenta, ésta técnica de subdivisión de elementos en conjuntos más pequeños es muy apropiada al caso de uso ya que la complejidad de optimizar cada subconjunto generado es menor.

6.3 Clusterización por homogeneidad

Debido a la necesidad de intentar reducir la complejidad del problema, y a la vez intentar guiar al algoritmo hacia una mejor solución en términos de homogeneidad de los lotes, una vez que se han creado unos grupos iniciales de productos, todos ellos con las mismas características comunes, la siguiente fase de la estrategia propuesta consiste en realizar un clusterizado de los mismos utilizando la homogeneidad como parámetro para las agrupaciones. Basándose en los estudios analizados en el capítulo 5, para el problema a resolver en esta tesis se va a utilizar un algoritmo basado en grafos, con lo que se representará cada producto como un nodo, que estará unido a otros productos (nodos) a través de un arco no dirigido con una distancia d . Es necesario como primer paso, por lo tanto, definir una manera de medir cómo de similares son todos los productos entre sí. Se define a continuación el concepto de distancia utilizado para posteriormente explicar en detalle el algoritmo de clusterización desarrollado.

6.3.1 Distancia definida entre productos

Como se ha descrito en el apartado 5.1, el primer paso a la hora de aplicar alguna técnica de clúster es definir una distancia de medida. Para poder discernir y agrupar productos por similitud, es necesario establecer una manera de poder comparar dos productos, teniendo en cuenta los parámetros locales de cada uno (subfamilia, grado de acero, aceitado, peso, ancho, espesor y espesor de recubrimiento). Se ha decidido representar cada producto en un espacio N-dimensional, como se muestra en la Figura 13, siendo las dimensiones las variables locales, y utilizar la norma euclídea para calcular la distancia entre dos productos. De esta forma, la distancia entre el producto i y el producto j se calcularía como:

$$d_{ij} = \sqrt{\sum_{k=1}^N (param_k_i - param_k_j)^2}$$

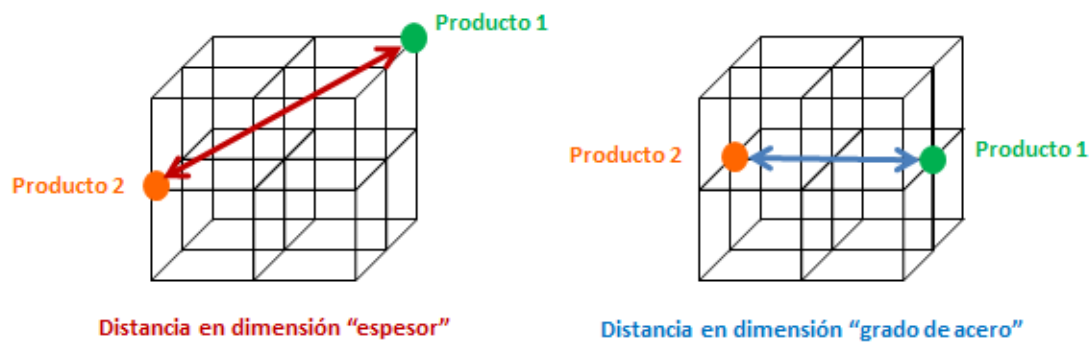


Figura 13. Ejemplo de representación en el espacio de las dimensiones “espesor” y “grado de acero”

6.3.1.1 Normalización y transformación de variables categóricas

Para poder comparar e incluir en la fórmula de la distancia los diferentes parámetros, es necesario por un lado realizar una normalización de las variables numéricas (peso, ancho, espesor y espesor de recubrimiento), y por otro una transformación de las variables categóricas (subfamilia, grado de acero, aceitado) a formato numérico. El procedimiento realizado es el siguiente:

Variables numéricas: todos los valores de cada variable numérica de los productos se normalizan dentro del rango [0, 1]. Para ello se utiliza la siguiente fórmula, siendo x_{\min} el menor de los valores del conjunto, y x_{\max} el mayor:

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Variables categóricas: Los valores numéricos asignados se han asignado con la validación de los expertos, considerando que 0, 0.1 y 1 eran una buena medida de la homogeneidad de los mismos. Las transformaciones realizadas para cada parámetro son:

- Subfamilia:
 - Para el caso de la familia “OC”, se permiten mezclar productos de sus subfamilias, pero se ha de priorizar que “OCR” y “OCH” vayan juntos, al

igual que "OAS" con "OAZ" o "OZ" o "OZA" o "OZE" o "OZO". Por ello, se define directamente la distancia entre grupos de la siguiente manera:

- OCR, OCH:
 - Distancia entre productos de igual subfamilia $\rightarrow 0$
 - Distancia entre productos de distinta subfamilia $\rightarrow 0.1$
- OOS, OAZ, OZ, OZA, OZE, OZE:
 - Distancia entre productos de igual subfamilia $\rightarrow 0$
 - Distancia entre productos de distinta subfamilia $\rightarrow 0.1$
- Distancia entre productos del grupo (OCR, OCH) con los productos del grupo (OOS, OAZ, OZ, OZA, OZE, OZE) $\rightarrow 1$

Expresándolo en formato de tabla, quedaría de la siguiente manera:

	OCR	OCH	OAS	OAZ	OZ	OZA	OZE	OZO
OCR	0	0.1	1	1	1	1	1	1
OCH	0.1	0	1	1	1	1	1	1
OAS	1	1	0	0.1	0.1	0.1	0.1	0.1
OAZ	1	1	0.1	0	0.1	0.1	0.1	0.1
OZ	1	1	0.1	0.1	0	0.1	0.1	0.1
OZA	1	1	0.1	0.1	0.1	0	0.1	0.1
OZE	1	1	0.1	0.1	0.1	0.1	0	0.1
OZO	1	1	0.1	0.1	0.1	0.1	0.1	0

Tabla 5. Distancias asignadas para productos de la familia OC

- Para el caso de la familia “HD”, los productos del tipo “HD-Z”, “HD-ZM”, “HD-ZMO” y “HD-ZA” pueden juntarse, priorizando que sean iguales, lo cual se traduce a:
 - Distancia entre productos de igual subfamilia $\rightarrow 0$
 - Distancia entre productos de distinta subfamilia $\rightarrow 0.1$

	HD-Z	HD-ZM	HD-ZMO	HD-ZA
HD-Z	0	0.1	0.1	0.1
HD-ZM	0.1	0	0.1	0.1
HD-ZMO	0.1	0.1	0	0.1
HD-ZA	0.1	0.1	0.1	0

Tabla 6. Distancias asignadas para productos de la familia HD

- Grado de acero: para favorecer la agrupación de los grados que combinan bien entre sí, se asignan las siguientes distancias:
 - Distancia entre productos del mismo grado de acero $\rightarrow 0$
 - Distancia entre productos de grados de acero del mismo grupo (aquellos que mezclan bien entre sí) $\rightarrow 0.1$
 - Distancia entre productos de distintos grupos (son grados de acero más diferentes) $\rightarrow 1$
 - En una misma familia, no existe ninguna prohibición de no mezclar ningún grado con otro, por lo que la mayor diferencia entre grados será el valor 1

Familia HR			
Grupo	Grados		
Grupo1	TYPE_DD11	TYPE_DD11ULC	TYPE_S185
Grupo2	TYPE_S235	TYPE_S275	
Grupo3	TYPE_HSS460	TYPE_HSS500	
Grupo4	TYPE_HSS660	TYPE_HSS_GT_600	
Grupo5	HIGH_CARBON_TYPE_C45_C60	HIGH_CARBON_TYPE_C25_C40	
Grupo6	TYPE_HSS_550	TYPE_HSS600	
Grupo7	TYPE_E295	TYPE_E335	TYPE_E360

Figura 14. Ejemplos de grupos de grado de acero para la familia HR

- Aceitado:
 - Productos con aceite → el campo aceitado se traduce a '1'
 - Productos sin aceite → el campo aceitado se traduce a '0'

6.3.1.2 Influencia dinámica de los parámetros sobre la distancia

Como se comentaba en la descripción del problema, uno de los requisitos del algoritmo es poder dar más importancia a unos factores que a otros, y el algoritmo ha de poder adaptarse dinámicamente y de permitir configurar este tipo de importancia relativa de las características.

Para poder permitir esta característica, se han incluido unos pesos que se aplican sobre la distancia en cada característica. El valor de los pesos es configurable en un rango que se ha definido de 0.1 a 10, para poder dar una diferencia de importancia de unos parámetros a otros de hasta 2 órdenes de magnitud. Estos factores se utilizan en la función del cálculo de la distancia entre cada par de productos, de esta manera se puede priorizar que a la hora de calcular la distancia tengan más importancia unas características del producto que otras. Para dar importancia al ratio entre unos pesos y otros, más que al propio valor asignado al peso, estos se normalizan utilizando la media geométrica, con lo que los valores de los pesos que se le pasan al algoritmo son:

$$\gamma'_i = \frac{\gamma_i}{\sqrt[N]{\prod_{i=1}^N \gamma_i}}$$

Teniendo en cuenta esto, la fórmula final con la que se calcula la distancia para cada par de productos es la siguiente:

$$d_{ij} = \sqrt{\sum_{k=1}^N \gamma_k \cdot (\text{param}_{k_i} - \text{param}_{k_j})^2}$$

En la Figura 15 se muestra un ejemplo de las distancias entre todos los pares de productos generados para un grupo de 36 elementos. Los pares con menores distancias están coloreados en verde, mientras que cuanto más separados están se van más al rojo.

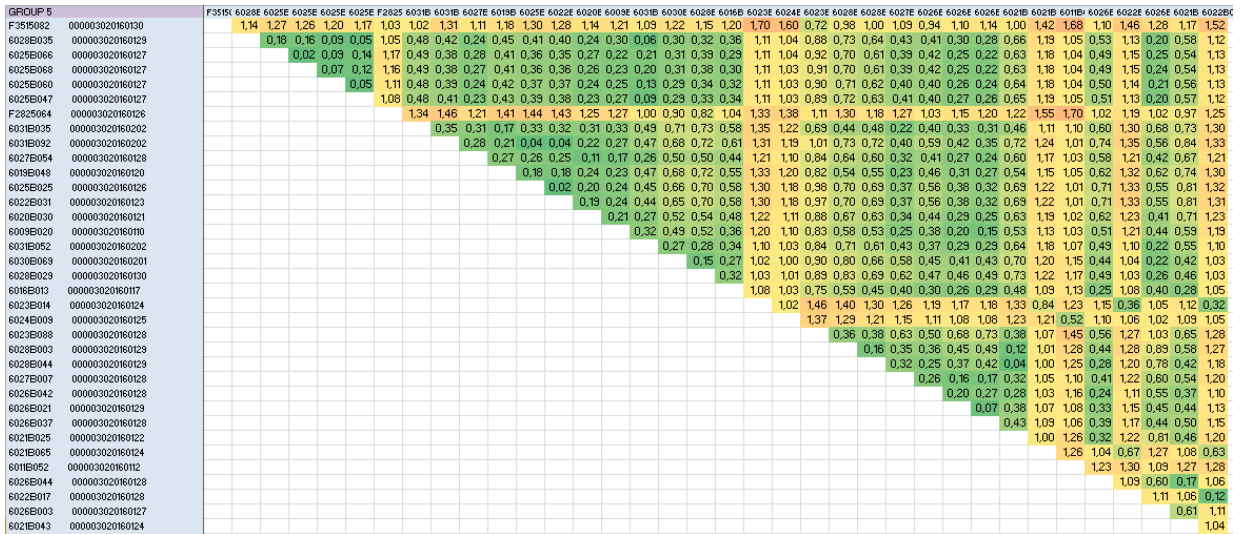


Figura 15. Distancias para un grupo de 36 elementos

6.3.2 Algoritmo de clustering

En este apartado se explica el algoritmo desarrollado en la tesis para realizar la agrupación inicial de productos similares. El algoritmo de clustering se aplica sobre cada uno de los N subconjuntos resultantes de la división inicial en función de los parámetros globales. Tras finalizar la ejecución del algoritmo de clustering sobre cada grupo el resultado es un número de grupos $M > N$, garantizando que todos los productos de cada grupo resultante son similares entre sí, de acuerdo al concepto de distancia anteriormente explicado y al valor de los pesos de los diferentes parámetros que se especifiquen.

Para el desarrollo del mismo, como se comentaba en el apartado anterior, se han tomado como base algoritmos basados en grafos. En concreto se han considerado los buenos resultados obtenidos por (Hendrickson & Leland, 1995) utilizando técnicas recursivas, la propuesta de Zhou et al. para realizar clústeres en función de la similitud de 2 atributos mediante una medida unificada ponderando dichos atributos (Zhou, et al., 2016); así como la partición repetitiva del grafo eliminando arcos de proteínas diferentes, utilizada en (Kawaji, et al., 2004). Se especifican a continuación los parámetros de entrada y la estructura del algoritmo utilizados en la tesis.

6.3.2.1 Parámetros de entrada

El algoritmo de clustering se lanza tantas veces como subconjuntos resultantes haya de la fase anterior; cada subconjunto garantiza que todos los productos tienen los mismos parámetros globales y que por lo tanto cualquier combinación de dichos productos satisfará las características obligatorias de cada lote. Para cada llamada al algoritmo se le pasan los siguientes parámetros y objetos:

Elementos del subconjunto: conjunto de elementos a agrupar

Distancia entre elementos: distancia entre cada par de elementos, calculada de la manera explicada anteriormente.

Threshold: distancia máxima deseada entre cada par de bobinas. Este Threshold es configurable para cada familia de productos. Por defecto el valor utilizado es 1, que es un valor apropiado para utilizar teniendo en cuenta que en la función del cálculo de distancia entre dos productos el peso de asociado a cada parámetro está comprendido entre 0.1 y 10, y los valores de las variables correspondientes a dichos parámetros también están normalizados en el rango $[0, 1]$. No obstante, es posible modificar este valor para afinar el algoritmo manualmente por cada familia de productos si fuera necesario.

MinSize: tamaño mínimo inicial de los clusters resultantes

MaxSize: tamaño máximo de los clusters resultantes

6.3.2.2 Estructura del algoritmo

Se ha denominado a la función de división por grafos Split. La estructura de dicho algoritmo sigue los siguientes pasos, ejecutados de manera recursiva para cada sub-grafo originado:

Paso 1 → Generación del grafo: se genera un grafo inicial no dirigido, considerando como nodos los productos, y generando un arco entre cada par de nodos i, j siempre que $d_{ij} < \text{Threshold}$.

Paso 2 → Generación de componentes conexas: separar las diferentes componentes conexas del grafo. Esto conlleva hacer varios sub-grafos a partir del grafo inicial.

Paso 3 → división subconjuntos más pequeños formado por productos más cercanos entre sí: para cada conjunto generado, se comprueba si su número de elementos está dentro del rango (MinSize, MaxSize). Si el número de elementos es mayor que MaxSize se vuelve a realizar una llamada recursiva a la función, reduciendo el threshold para conseguir subconjuntos más pequeños con productos más cercanos. En cada llamada a la función de nuevo para cada par de nodos cuya distancia sea menor al threshold se genera un arco que los une. En cada llamada recursiva el Threshold se actualiza de la siguiente forma:

$$\text{Threshold}' = \text{Threshold} * \text{Step}$$

El valor del Step se ha fijado a 0.8, de manera que se vaya disminuyendo en cada iteración. Esto se produce un número N determinado de veces, hasta que el tamaño del subconjunto está dentro del límite superior MaxSize, o en el caso de que no sea posible rebajar su tamaño, para evitar un número infinito de llamadas en el caso de que no se encuentren grupos de tamaño menor que MaxSize, se comprueba que el nuevo valor de Threshold' no sea menor que 10^{-6} , en cuyo caso no se realizan más llamadas recursivas. En cada llamada el algoritmo intentará juntar bobinas con Threshold más pequeño, o dicho de otra manera, más similares entre sí.

Si en cualquiera de las llamadas recursivas el tamaño del conjunto es menor que MinSize, se deshace el subconjunto y los productos se dejan cada uno independiente ("single").

Paso 4 → reagrupación de las singles en clusters: a medida que se realiza el proceso de agrupación, por el camino van quedando productos sueltos que no tienen la distancia suficientemente pequeña entre ellos como para formar parte de una componente conexas con el Threshold actual, pero puede que sí con algún threshold intermedio. En el proceso de vuelta del algoritmo se realiza una comprobación para ver los elementos que han quedado sueltos, y comprobar su posible agrupación en clústeres con un threshold intermedio. Para ello, el nuevo threshold se genera de la siguiente manera:

$$\text{Threshold}' = (\text{Threshold}' * \text{Threshold})^{1/2}$$

Esta comprobación se realiza hasta que el valor de la diferencia entre ambos thresholds sea de 10^{-6} , de esta manera se realizan varias comprobaciones para agrupar productos cercanos entre sí.

En la Figura 16 se observa la división de una de las componentes conexas (cluster inicial). En este ejemplo los grupos se crean de la siguiente manera:

1) División:

- Partiendo del conjunto inicial, se generan aquellas componentes conexas del grafo de tal manera que la distancia entre elementos del mismo sea < 1 , resultando tras esta llamada inicial a la función Split varios sub-grafos, junto con productos sueltos cuya distancia al resto de elementos es > 1 .
- Para cada sub-grafo generado, se repite el proceso, generando un nuevo Threshold multiplicando el anterior por 0.8 . Como resultado, tendremos una serie de sub-grafos cuyos elementos tienen una distancia entre ellos < 0.8 , y unos elementos sueltos cuya distancia es > 0.8 .
- Este proceso se repite varias veces, como se indicaba anteriormente, hasta que todos los productos han sido agrupados o hasta que el valor del threshold es demasiado pequeño como para poder agrupar productos. Como resultado final, tendremos varios clústeres generados a lo largo de todo el proceso, con diferentes thresholds.

2) Reagrupación:

- En el proceso de vuelta de la función recursiva, lo que se hace es comprobar si los elementos sueltos que han quedado por el camino se pueden agrupar en clústeres con thresholds intermedios a los utilizados hasta ahora (siempre y cuando el número de elementos sea mayor que MinSize); en el fondo, este proceso busca reagrupar los elementos sueltos en clústeres de distancia un poco mayor, pero siempre dentro de la máxima distancia permitida. Por ejemplo:
 - Para aquellos productos sueltos generados con $\text{threshold} = 0.64$, se comprueba si existe algún threshold intermedio entre 0.64 y 0.8 que los

permita agrupar, generando los thresholds intermedios como se explicaba anteriormente, utilizando la raíz cuadrada de los thresholds anterior y posterior, y realizando esta comprobación un número limitado de veces. En el ejemplo de la figura, se generan clústeres con threshold 0'71; y a su vez, al repetir el proceso entre 0'71 y 0'64 se generarían clústeres con threshold 0'67.

- El mismo ejemplo se puede aplicar cuando se vuelve un nivel más hacia arriba, con aquellos elementos que no estaban suficientemente cercanos como para agruparse con un threshold = 0'8, pero que por ejemplo sí que algunos de ellos se agruparían con threshold = 0'89.

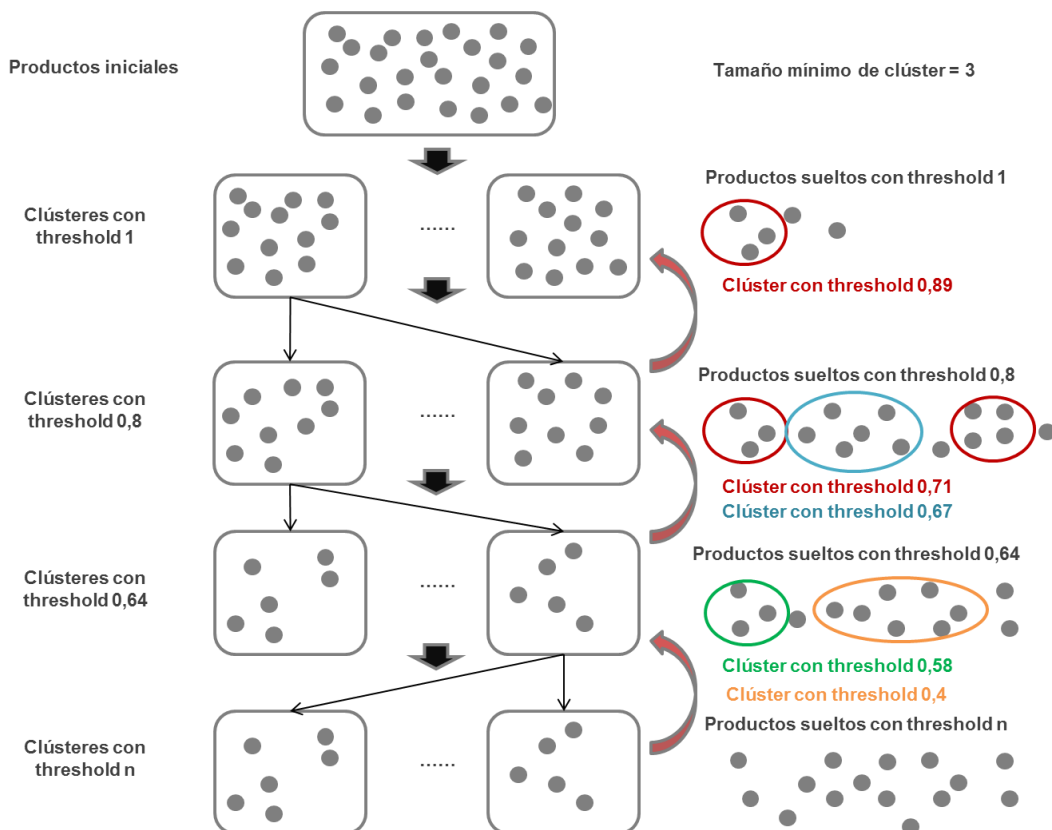


Figura 16. Proceso de clustering de un conjunto

En la Figura 17 se muestra el diagrama de flujo, y a continuación en la Figura 18 se muestra el pseudocódigo del algoritmo.

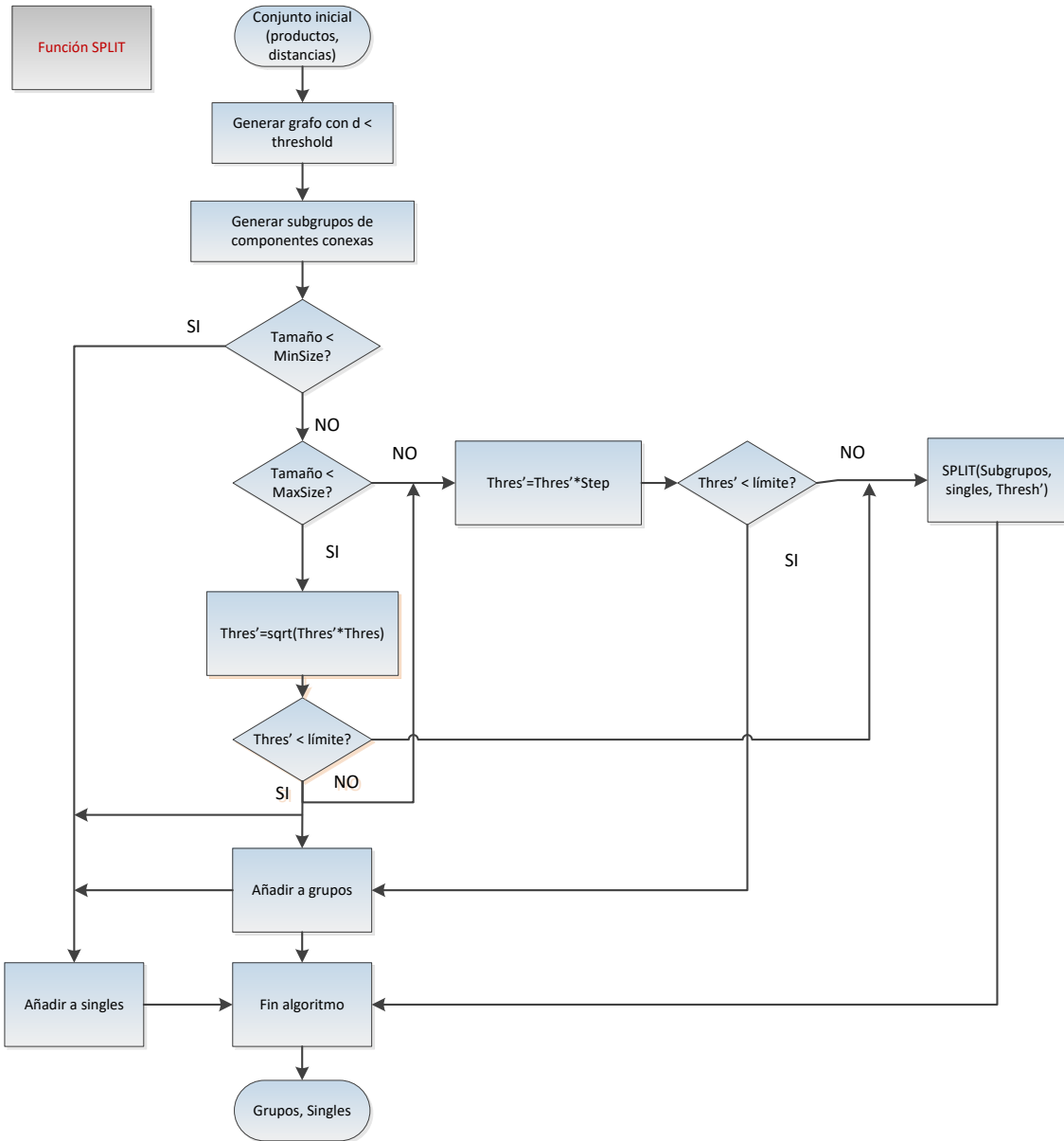


Figura 17. Esquema del proceso de clustering

```

SPLIT_GROUP (GR)
%Inputs: grupo, Threshold, MinSize, MaxSize

Graph = crear_grafo (grupo, Threshold)
    Nodos=Productos
    Crear arco si la distancia entre el producto i y el j es menor que el Threshold

%Sacar componentes conexas y las que quedan sueltas (singles)
Subgrupos, Singles = sacar componentes conexas de Graph

NumGroups = Tamaño|Subgrupos|

Para todo subgrupo i :=1, ..., NumGroups
    Si longitud(subgrupo) < MinSize
        Asignar los productos como singles %No llegan al tamaño mínimo

    Si longitud(subgrupo) > MaxSize:
        %Seguir dividiendo
        Threshold' = Threshold * Step
        Si Threshold' > limite
            SPLIT_GROUP (Threshold')
        Si no
            %Tamaño de grupo permitido → Refinar
            NewThreshold = sqrt (Threshold' * Threshold)
            Si NewThreshold > limite
                SPLIT_GROUP (NewThreshold)
                %Llamada recursiva a la función para intentar agrupar las
                singles generadas
            Si no
                Continuar

    Si no %Tamaño de grupo permitido → Refinar
        NewThreshold = sqrt (Threshold' * Threshold)
        Si NewThreshold > limite
            SPLIT_GROUP (NewThreshold)
            %Llamada recursiva a la función para intentar agrupar las
            singles generadas

Return singles, groups

```

Figura 18. Estructura del algoritmo de clustering

6.3.2.3 Ejemplo de influencia de los parámetros

Se muestran a continuación ejemplos de ejecución del algoritmo de clustering variando los parámetros minsize, maxsize y threshold, para aclarar su comportamiento. Para

todos los experimentos el valor de los pesos de cada parámetro será 1, de manera que los experimentos sean comparables. Para cada ejecución que se muestra en cada ejemplo se ha lanzado el algoritmo de clustering 10 veces, mostrando finalmente la media de los valores obtenidos en las 10 ejecuciones.

Se parte de una entrada con 284 productos. Tras hacer la división inicial por parámetros globales, quedan 5 grupos que se le pasarían al algoritmo de clustering:

- Grupo 1: 112 productos
- Grupo 2: 55 productos
- Grupo 3: 80 productos
- Grupo 4: 36 productos
- Grupo 5: 1 producto

Ejemplo base: MinSize = 5, MaxSize = 100, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños. [10, 5, 6, 13, 10, 5, 7, 5, 6, 8, 7, 8, 6, 9]
 - 7 productos sin agrupar
- Grupo 2:
 - 7 clústeres con los siguientes tamaños: [5, 11, 5, 11, 5, 5, 6]
 - 7 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [6, 6, 9, 21, 10, 8, 12, 5]
 - 3 productos sin agrupar
- Grupo 4:
 - 5 clústeres con los siguientes tamaños: [10, 6, 5, 5, 5]
 - 5 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar

- Total:
 - 34 clústeres (media de 7,6 productos por clúster) 23 productos sin agrupar

EFFECTO EN VARIACIÓN DE THRESHOLD

Para observar el efecto de la variación del threshold se realizarán 3 ejecuciones más, con valores de 1'5, 2 y 3.

Incremento del Threshold inicial. MinSize = 5, MaxSize = 100, Threshold = 1'5.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños: [6, 15, 6, 13, 10, 5, 7, 5, 6, 8, 7, 8, 6, 9]
 - 1 producto sin agrupar
- Grupo 2:
 - 7 clústeres con los siguientes tamaños: [8, 13, 8, 9, 7, 5, 5]
 - 0 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños:[9, 6, 9, 21, 10, 8, 12, 5]
 - 0 productos sin agrupar
- Grupo 4: 6 clústeres con los siguientes tamaños: [5, 10, 6, 5, 5, 5]
 - 0 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 35 clústeres (media de 8 productos por clúster), 2 productos sin agrupar

Incremento del Threshold inicial. MinSize = 5, MaxSize = 100, Threshold = 2.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños: [7, 7, 11, 6, 13, 10, 5, 7, 5, 11, 7, 8, 6, 9]
 - 0 productos sin agrupar
- Grupo 2:
 - 8 clústeres con los siguientes tamaños:[7, 5, 11, 5, 11, 5, 5, 6]
 - 0 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [6, 6, 9, 21, 10, 8, 12, 5]
 - 3 productos sin agrupar
- Grupo 4:
 - 6 clústeres con los siguientes tamaños: [5, 10, 6, 5, 5, 5]
 - 0 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 36 clústeres (media de 7'8 productos por clúster), 4 productos sin agrupar

Incremento del Threshold inicial. MinSize = 5, MaxSize = 100, Threshold = 3.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños: [7, 15, 6, 13, 10, 5, 7, 5, 6, 8, 7, 8, 6, 9]
 - 0 productos sin agrupar
- Grupo 2:
 - 7 clústeres con los siguientes tamaños: [8, 13, 8, 9, 7, 5, 5]
 - 0 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [9, 6, 9, 21, 10, 8, 12, 5]

- 0 productos sin agrupar
- Grupo 4:
 - 6 clústeres con los siguientes tamaños: [5, 10, 6, 5, 5, 5]
 - 0 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 35 clústeres (media de 8'1 productos por clúster), 1 producto sin agrupar

La Tabla 7 resume los resultados de las ejecuciones con los 4 valores de threshold, pudiendo comparar su evolución en la Figura 19. :

	1	1'5	2	3
Clusters	34	35	36	35
Productos / clúster	7'6	8	7'8	8'1
Productos sin agrupar	23	2	4	1

Tabla 7. Efecto de variación de threshold en el algoritmo de clustering

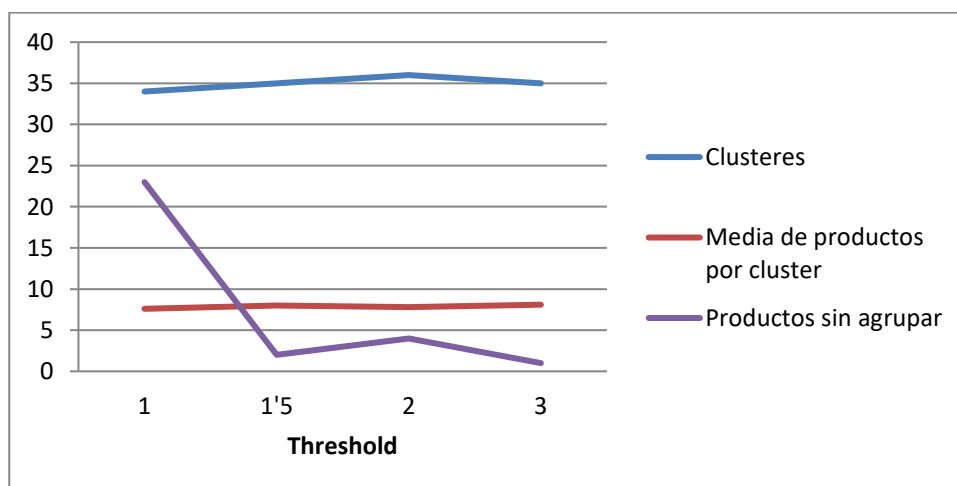


Figura 19. Evolución de la salida del algoritmo de clustering al variar el threshold

Como se puede observar, la tendencia general al aumentar el threshold es a agrupar más elementos en clústeres (se permite juntar elementos más dispersos), siendo esta tendencia muy grande al principio. El número de clústeres se mantiene más o menos parecido.

EFFECTO EN INCREMENTO DE MINSIZE

Para observar el efecto de la variación del MinSize se realizarán 3 ejecuciones más, con valores de 20, 30 y 35.

Incremento del MinSize: MinSize = 20, MaxSize = 100, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 4 clústeres con los siguientes tamaños: [22, 24, 23, 21]
 - 22 productos sin agrupar
- Grupo 2:
 - 1 clúster con el siguiente tamaño: [42]
 - 13 productos sin agrupar
- Grupo 3:
 - 2 clústeres con los siguientes tamaños: [26, 39]
 - 15 productos sin agrupar
- Grupo 4:
 - 1 clúster con el siguiente tamaño: [24]
 - 12 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 8 clústeres (media de 27'6 productos por clúster), 63 productos sin agrupar

Incremento del MinSize: MinSize = 30, MaxSize = 100, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 2 clústeres con los siguiente tamaños: [47, 30]
 - 35 productos sin agrupar
- Grupo 2:
 - 1 clúster con el siguiente tamaño: [42]
 - 13 productos sin agrupar
- Grupo 3:
 - 2 clústeres con los siguientes tamaños: [32, 39]
 - 9 productos sin agrupar
- Grupo 4:
 - 36 elementos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 5 clústeres (media de 38 productos por clúster), 94 productos sin agrupar

incremento del MinSize: MinSize = 35, MaxSize = 100, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 2 clústeres con los siguientes tamaños: [43, 47]
 - 22 productos sin agrupar
- Grupo 2:
 - 1 clúster con el siguiente tamaño: [42]
 - 13 productos sin agrupar
- Grupo 3:
 - 1 clúster con el siguiente tamaño: [39]

- 41 productos sin agrupar
- Grupo 4:
 - 36 elementos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 4 clústeres (media de 42'7 productos por clúster), 113 productos sin agrupar

La Tabla 8 resume los resultados de las ejecuciones con los 4 valores de MinSize, pudiendo comparar su evolución en la Figura 20:

	10	20	30	35
Clusters	34	8	5	4
Productos / clúster	7'6	27'6	38	42'7
Productos sin agrupar	23	63	94	113

Tabla 8. Efecto de variación de MinSize en el algoritmo de clustering

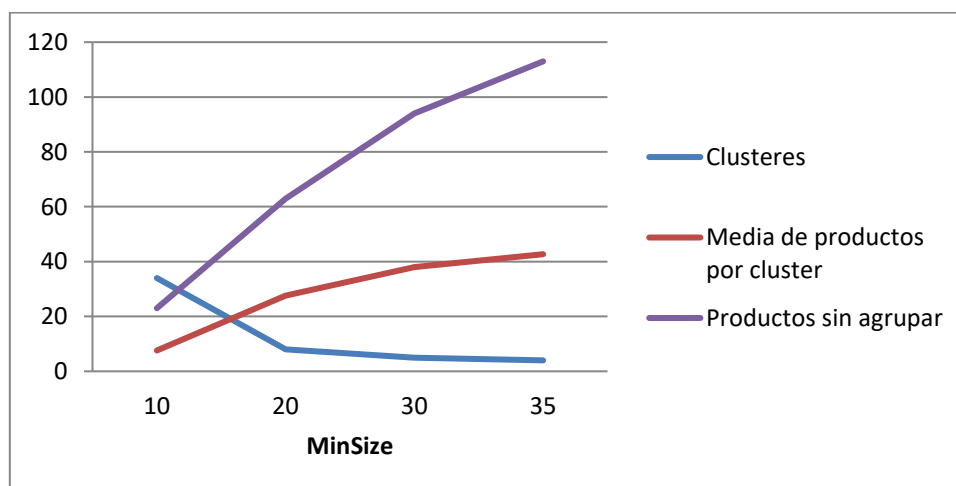


Figura 20. Evolución de la salida del algoritmo de clustering al variar el MinSize

Como se puede observar, el comportamiento es el esperado a priori: a medida que se aumenta el tamaño mínimo de los clústeres, es más difícil encontrar tantos productos suficientemente parecidos como para poder agruparse, de ahí que el número de clústeres sea inversamente proporcional al MinSize, mientras que el número de productos sin agrupar sea directamente proporcional. Al aumentar el tamaño mínimo de clúster es lógico también que la media de productos por clúster aumente.

EFFECTO EN DECREMENTO DE MAXSIZE

Para observar el efecto de la variación del MaxSize se realizarán 3 ejecuciones más, con valores de 50, 25 y 10.

MinSize = 5, MaxSize = 50, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños: [7, 5, 6, 6, 13, 10, 5, 7, 5, 11, 7, 8, 6, 9]
 - 7 productos sin agrupar
- Grupo 2:
 - 7 clústeres con los siguientes tamaños: [5, 11, 5, 11, 5, 5, 6]
 - 7 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [6, 6, 9, 21, 10, 8, 12, 5]
 - 3 productos sin agrupar
- Grupo 4:
 - 5 clústeres con los siguientes tamaños: [10, 6, 5, 5, 5]
 - 5 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:

- 34 clústeres (media de 7'67 productos por clúster), 23 productos sin agrupar

MinSize = 5, MaxSize = 25, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:
 - 14 clústeres con los siguientes tamaños: [7, 5, 5, 6, 7, 16, 5, 8, 5, 11, 7, 8, 6, 9]
 - 7 productos sin agrupar
- Grupo 2:
 - 8 clústeres con los siguientes tamaños: [5, 6, 5, 8, 8, 5, 5, 6]
 - 7 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [6, 6, 9, 21, 10, 8, 12, 5]
 - 3 productos sin agrupar
- Grupo 4:
 - 5 clústeres con los siguientes tamaños: [10, 6, 5, 5, 5]
 - 5 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 35 clústeres (media de 7'45 productos por clúster), 23 productos sin agrupar

MinSize = 5, MaxSize = 10, Threshold = 1.

El resultado de lanzar el algoritmo de clustering con estos parámetros es el siguiente:

- Grupo 1:

- 15 clústeres con los siguientes tamaños: [7, 8, 6, 7, 6, 10, 5, 8, 5, 5, 8, 7, 8, 6, 9]
- 7 productos sin agrupar
- Grupo 2:
 - 8 clústeres con los siguientes tamaños: [5, 5, 6, 6, 8, 5, 5, 6]
 - 9 productos sin agrupar
- Grupo 3:
 - 8 clústeres con los siguientes tamaños: [6, 6, 9, 21, 10, 8, 12, 5]
 - 3 productos sin agrupar
- Grupo 4:
 - 5 clústeres con los siguientes tamaños: [10, 5, 5, 5, 5]
 - 6 productos sin agrupar
- Grupo 5:
 - 1 producto sin agrupar
- Total:
 - 36 clústeres (media de 7'1 productos por clúster), 26 productos sin agrupar

La Tabla 9 resume los resultados de las ejecuciones con los 4 valores de MaxSize, pudiendo comparar su evolución en la Figura 21. Evolución de la salida del algoritmo de clustering al variar el MaxSize:

	100	50	25	10
Clusters	34	34	35	36
Productos / clúster	7'6	7'6	7'4	7'1
Productos sin agrupar	23	23	23	26

Tabla 9. Efecto de variación de MaxSize en el algoritmo de clustering

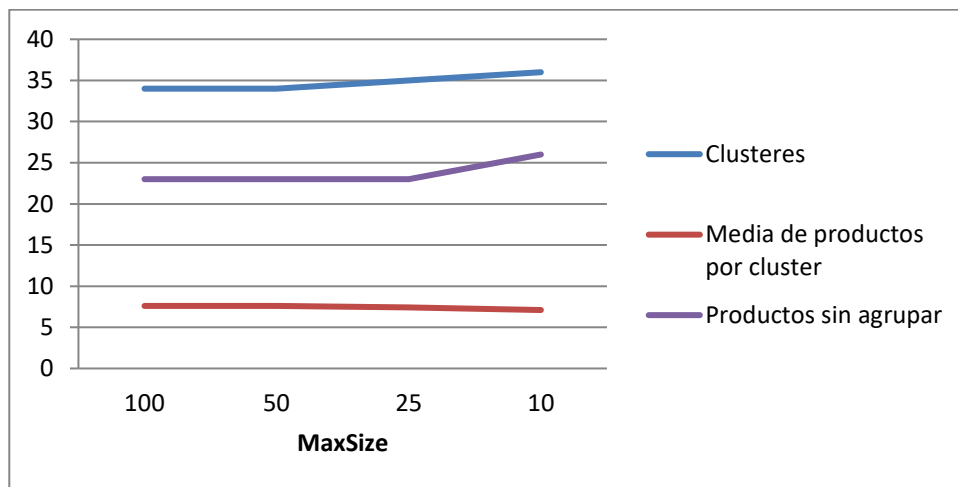


Figura 21. Evolución de la salida del algoritmo de clustering al variar el MaxSize

La gráfica muestra como la salida del algoritmo permanece más o menos constante en el rango de MaxSize [50, 100]. Sin embargo, a partir de valores de MaxSize menores de 50 se observa como el número de clústeres aumenta, pero también el número de productos que quedan sin agrupar en ningún clúster, si bien esta variación no es fuertemente significativa.

Como conclusión del apartado se puede concluir que tras analizar diversas técnicas de clustering existentes en la literatura, se ha propuesto un algoritmo para realizar unas primeras agrupaciones de productos en función de su similitud, basándose en ideas encontradas en la literatura: utilización de distancia euclídea para conseguir 1 único valor de distancia entre productos, representación mediante grafos y uso de un algoritmo recursivo basado en la distancia entre elementos para crear los clústeres. Las pruebas realizadas demuestran que el algoritmo se comporta según lo esperado y que presenta una buena solución intermedia. La salida de este algoritmo será utilizada en la última fase para crear los lotes finales, como se explicará en el siguiente apartado. Se ha optado por centrar el esfuerzo de mejora en la parte final de bin packing, debido a que es la parte donde se pueden conseguir mejores ahorros.

Como ejemplo real del número de grupos y tamaños de los mismos, en una ejecución para una semana estándar con todas las plantas, utilizando como parámetros $MinSize=5$, $MaxSize=100$, $Threshold=1.5$ se obtienen los siguientes resultados:

- Productos: 2771
- Productos sin agrupar: 62
- Número de clústeres: 136

En la Figura 22 se puede ver el número de productos asignados a cada clúster.

Cluster	Elementos	Cluster	Elementos	Cluster	Elementos	Cluster	Elementos
Cluster 1	5	Cluster 35	8	Cluster 69	11	Cluster 103	23
Cluster 2	5	Cluster 36	8	Cluster 70	11	Cluster 104	24
Cluster 3	5	Cluster 37	8	Cluster 71	11	Cluster 105	25
Cluster 4	5	Cluster 38	8	Cluster 72	11	Cluster 106	25
Cluster 5	5	Cluster 39	8	Cluster 73	12	Cluster 107	26
Cluster 6	5	Cluster 40	8	Cluster 74	12	Cluster 108	26
Cluster 7	5	Cluster 41	8	Cluster 75	12	Cluster 109	26
Cluster 8	5	Cluster 42	8	Cluster 76	12	Cluster 110	27
Cluster 9	5	Cluster 43	8	Cluster 77	12	Cluster 111	28
Cluster 10	5	Cluster 44	9	Cluster 78	13	Cluster 112	30
Cluster 11	5	Cluster 45	9	Cluster 79	13	Cluster 113	30
Cluster 12	5	Cluster 46	9	Cluster 80	13	Cluster 114	32
Cluster 13	5	Cluster 47	9	Cluster 81	14	Cluster 115	35
Cluster 14	5	Cluster 48	9	Cluster 82	14	Cluster 116	36
Cluster 15	6	Cluster 49	9	Cluster 83	14	Cluster 117	37
Cluster 16	6	Cluster 50	9	Cluster 84	14	Cluster 118	38
Cluster 17	6	Cluster 51	9	Cluster 85	15	Cluster 119	38
Cluster 18	6	Cluster 52	9	Cluster 86	15	Cluster 120	40
Cluster 19	6	Cluster 53	10	Cluster 87	15	Cluster 121	41
Cluster 20	6	Cluster 54	10	Cluster 88	16	Cluster 122	44
Cluster 21	6	Cluster 55	10	Cluster 89	16	Cluster 123	48
Cluster 22	6	Cluster 56	10	Cluster 90	16	Cluster 124	50
Cluster 23	6	Cluster 57	10	Cluster 91	16	Cluster 125	50
Cluster 24	7	Cluster 58	10	Cluster 92	17	Cluster 126	55
Cluster 25	7	Cluster 59	10	Cluster 93	19	Cluster 127	55
Cluster 26	7	Cluster 60	10	Cluster 94	19	Cluster 128	56
Cluster 27	7	Cluster 61	10	Cluster 95	20	Cluster 129	57
Cluster 28	7	Cluster 62	10	Cluster 96	20	Cluster 130	65
Cluster 29	7	Cluster 63	11	Cluster 97	21	Cluster 131	65
Cluster 30	7	Cluster 64	11	Cluster 98	21	Cluster 132	66
Cluster 31	7	Cluster 65	11	Cluster 99	22	Cluster 133	66
Cluster 32	8	Cluster 66	11	Cluster 100	22	Cluster 134	78
Cluster 33	8	Cluster 67	11	Cluster 101	22	Cluster 135	78
Cluster 34	8	Cluster 68	11	Cluster 102	23	Cluster 136	135

Figura 22. Salida del algoritmo de clustering en una ejecución con datos reales de 1 semana

6.4 Asignación de productos a lotes

Como se explicaba en el apartado 6.1, la estrategia de resolución consiste en una metodología basada en 3 fases bien diferenciadas:

- 1) Pre-procesado de datos y división inicial de todos los productos en conjuntos con las mismas características
- 2) Clusterización de productos basándose en la similitud entre los mismos
- 3) Asignación final de productos a lotes

Llegados a este punto, y tras haber realizado las dos primeras fase, la última fase parte de los grupos de productos generados de la fase anterior, siendo cada uno de esos grupos un conjunto de productos similares entre sí. Para cada uno de esos conjuntos se ha de realizar una agrupación de los productos en los lotes ya definitivos, manteniendo el tamaño mínimo y máximo de los lotes, y procurando que los productos de un lote sean lo más homogéneos posibles.

6.4.1 Pruebas previas

En un primer intento de resolución del problema se planteó un modelo lineal para resolver el problema de manera exacta. El modelo planteado se describe a continuación:

Conjuntos:

C : productos a agrupar.

B : conjunto de lotes potenciales, de cardinalidad $|C|$ (el máximo de lotes se daría en el caso de que cada lote sólo contuviera un producto).

Variables:

a_{cb} , $(c, b) \in C \times B$: Variable binaria para representar la asignación de un producto c a un lote b .

h : heterogeneidad de los lotes, correspondiendo a la máxima distancia entre los productos del lote.

$r_c, c \in C$: variable binaria para representar si un producto c no está asignado a ningún lote.

$s_{cc'}, (c, c') \in C \times C$: variable binaria para representar si el producto c y c' están en el mismo lote.

$s_{cc'b}, (c, c', b) \in C \times C \times B$: variable binaria para representar si los productos c y c' están ambos en el lote b .

$u_b, b \in B$: variable binaria para representar si el lote b se usa (tiene algún producto asignado) o no.

Parámetros:

$d_{cc'}$: distancia entre los productos c y c' , definida sobre $c' > c$ por simetría.

w_c : peso del producto c

B_{\max} : máximo peso de un lote

B_{\min} : mínimo peso de un lote

Modelo:

La función objetivo es una combinación lineal del número de lotes, la heterogeneidad de los mismos, y del número de productos que quedan sin asignar.

$$\text{Minimizar } (h + \sum_{b \in B} u_b + \sum_{c \in C} r_c) \quad (1)$$

Las principales variables de decisión son a_{cb} , que indican la asignación de los productos a los lotes. De estas se pueden derivar qué pares de productos están en el mismo lote ($s_{cc'}$), calculado a su vez utilizando $s_{cc'b}$ (representando si los dos productos pertenecen al lote b en particular).

Cada $s_{cc'b}$ es equivalente a la función lógica a_{cb} AND $a_{c'b}$, y cada $s_{cc'}$ es equivalente a la función lógica OR de $s_{cc'b}$ en todo los lotes, lo cual se representa linealmente en las ecuaciones 2-6 como:

$$s_{cc'b} \leq a_{cb} \quad \forall (c, c', b) \in C \times C \times B, c' > c \quad (2)$$

$$s_{cc'b} \leq a_{c'b} \quad \forall (c, c', b) \in C \times C \times B, c' > c \quad (3)$$

$$s_{cc'b} \geq a_{cb} + a_{c'b} - 1 \quad \forall (c, c', b) \in C \times C \times B, c' > c \quad (4)$$

$$s_{cc} \geq s_{cc'b} \quad \forall (c, c', b) \in C \times C \times B, c' > c \quad (5)$$

$$s_{cc'b} \leq \sum_{b \in B} s_{cc'rb} \quad \forall (c, c') \in C \times C, c' > c \quad (6)$$

Se ha de garantizar que cada producto se asigne sólo a un lote:

$$\sum_{b \in B} a_{cb} \leq 1 \quad \forall (c) \in C \quad (7)$$

La heterogeneidad de los lotes debe de ser positiva, y mayor o igual que la distancia entre productos del lote:

$$h \geq s_{cc'} d_{cc'} \quad \forall (c, c') \in C \times C, c' > c \quad (8)$$

Las restricciones de peso mínimo y máximo se representan como:

$$u_b \text{Bmin} \leq \sum_{c \in C} a_{cb} w_c \leq u_b \text{Bax} \quad \forall b \in B \quad (9)$$

El uso de los lotes se describe con las 2 siguientes ecuaciones:

$$u_b \geq a_{cb} \quad \forall (c, b) \in C \times B \quad (10)$$

$$u_b \leq \sum_{c \in C} a_{cb} \quad \forall b \in B \quad (11)$$

Finalmente, el número de productos sin asignar se calcula como:

$$r_c = 1 - \sum_{b \in B} a_{cb} \quad \forall c \in C \quad (12)$$

Para probar el algoritmo se generaron 5 instancias de prueba con productos reales, de tamaños 5, 15, 25, 35 y 40 elementos. Las pruebas se realizaron en un ordenador con un procesador Intel Xeon X5675 @ 3GHz, con 16 GB de RAM. El algoritmo se codificó en AMPL y el solver utilizado para resolver el problema fue CPLEX. Los resultados obtenidos se muestran a continuación, y su evolución gráfica en minutos en la Figura 23:

- 5 elementos: 0 segundos
- 15 elementos: 4 segundos
- 25 elementos: 150 segundos
- 35 elementos: 900 segundos
- 40 elementos: 182300 segundos

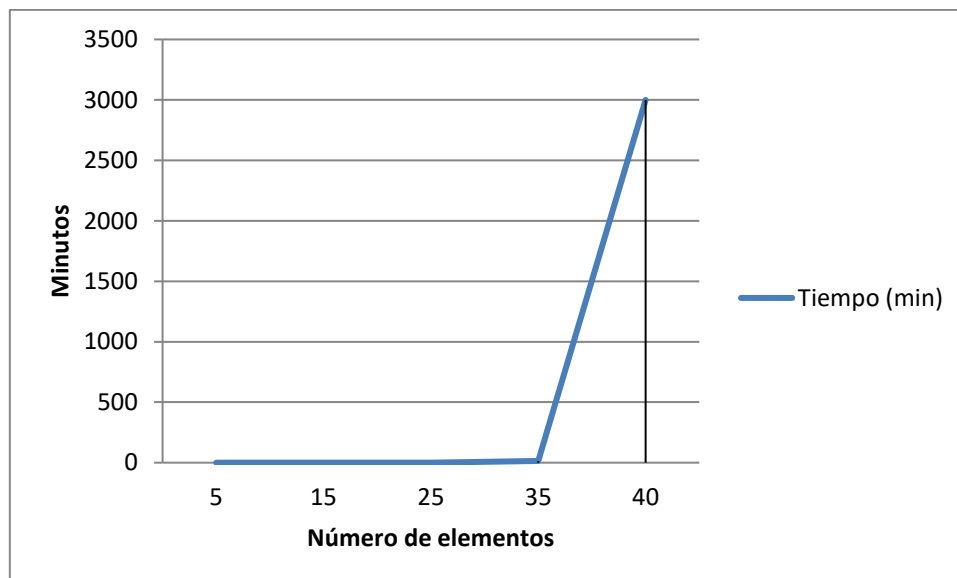


Figura 23. Evolución temporal con el número de elementos para la solución exacta

En el apéndice A del anexo se muestran las instancias utilizadas para realizar estas pruebas previas.

Como se analizó en el apartado 4 en el estado del arte de las soluciones existentes en los problemas de Bin Packing Problem, al hacer la prueba empírica se corroboró que el tiempo de ejecución para encontrar la solución óptima para un número de lotes y tamaños como los

que se han de resolver en el caso de esta tesis es inviable. El tiempo de ejecución no depende además solamente del número de elementos, sino de las distancias entre ellos, ya que si existen muchos elementos muy similares, el algoritmo branch-and-bound que utiliza el solver para resolver el problema tarda más en encontrar la solución. Es necesario, pues, buscar algoritmos que sean capaces de encontrar buenas soluciones en tiempos cortos. En el siguiente apartado se explica la estructura del algoritmo para la tercera fase planteado en la tesis.

6.4.2 Estructura del algoritmo

Teniendo en cuenta el número de grupos y el tamaño de los mismos que pueden resultar en una ejecución normal, es necesario que se desarrolle un algoritmo de generación de lotes rápido, ya que se ha de ejecutar un número elevado de veces. El seguimiento de la metodología en las 3 fases definidas garantiza que los productos que forman parte de los grupos iniciales de entrada al algoritmo de la última fase son similares entre sí. El objetivo es por lo tanto minimizar el número de lotes y elementos sueltos, y a la vez agrupar aún más si cabe elementos por similitud. Para resolver el problema se ha creado el algoritmo CREATE_BUNDLES. Los pasos que se realizan para buscar la mejor solución son los siguientes:

- 1) **Ordenación de los elementos:** el listado de productos que se deben asignar a lotes finales se ordenan en función de algún criterio. Estos elementos se irán recorriendo después, asignándose a un lote final definitivo. Las diferentes ordenaciones que se realizan son:
 - 1) De mayor a menor tamaño (largest first): ordenación de los elementos según peso decreciente, siendo el primero el de mayor peso y el último el de menor peso.
 - 2) De menor a mayor tamaño (smallest first): ordenación de los elementos según peso creciente, siendo el primero el de menor peso y el último el de mayor peso.
 - 3) Aleatorio

- 2) **Asignación de elementos a lotes finales:** se han definido 3 heurísticas basadas en las heurísticas clásicas de mejores resultados, adaptándolas para incluir la homogeneidad de los diferentes elementos (FIFO, Greedy y basada en distancias), junto con un meta-heurístico GRASP. Cada una de ellas se explicará en detalle en el apartado siguiente. Para cada una de ellas en cada repetición se ordenan de manera aleatorio los elementos; y además se realizan un par de repeticiones más, una ordenando los elementos según “largest first”, y otra “smaller first”.
- 3) **Evaluación de la solución:** tras finalizar la asignación de todos los productos a lotes se evalúa la función de coste correspondiente a la solución. De todas las múltiples soluciones analizadas el algoritmo se quedará con aquella asignación de productos que tenga el menor coste.

La Figura 24 representa gráficamente el proceso de asignación de productos a los lotes finales. Es un esquema que representa el proceso genérico, para el caso del GRASP existe además una parte de búsqueda local para mejorar la solución encontrada.

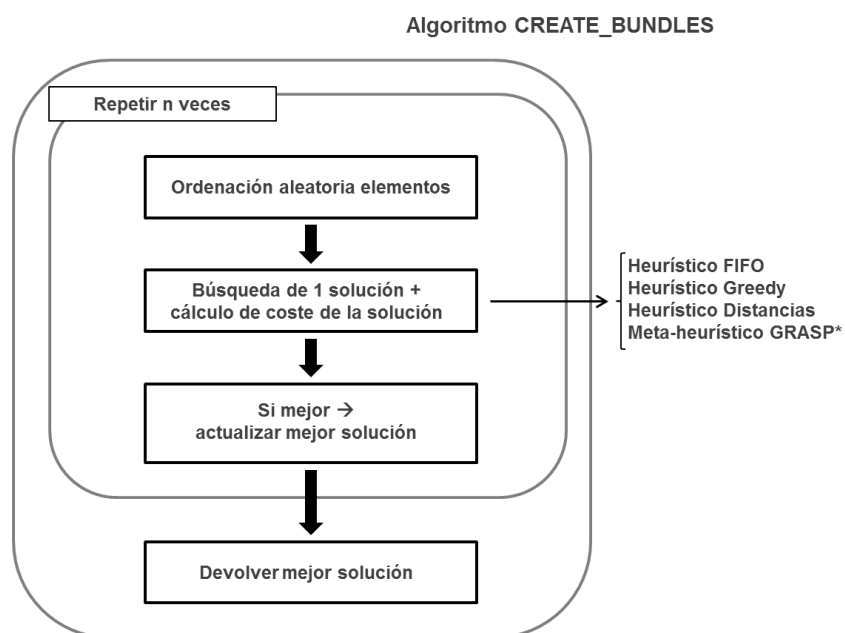


Figura 24. Estructura del algoritmo CREATE_BUNDLES

Todas las heurísticas hacen uso de la función “can_add” que comprueba si en un momento determinado un producto puede ser añadido a un lote, teniendo en cuenta por un lado la capacidad disponible en el lote y por otro garantizando que la distancia del producto a todos los del lote sea menor que el Threshold. La función retorna 2 booleanos indicando si el producto puede ser incluido en el lote, tanto por distancia a los productos ya asignados al lote, como por capacidad disponible del mismo. El esquema de la función es el siguiente:

```
CAN_ADD(Lote, producto)
%Inputs: lote, product, Threshold, lote_max_size
%Comprueba si es posible añadir un producto a un lote tanto por el tamaño del lote
como por la distancia entre productos < Threshold

Si tamaño_lote + tamaño_producto <= tamaño_max_lote
    size_condition = Verdadero
Si no
    size_condition = Falso

Para cada producto product del lote Si distancia(producto, product) < Threshold
    dist_condition = Verdadero
Si no
    dist_condition = Falso
Fin %Devolver size_condition (V/F), dist_condition (V/F)
```

Figura 25. Pseudocódigo de la función CAN_ADD

En los apartados siguientes se explican en detalle los parámetros de entrada, la función de coste y las heurísticas y el meta-heurístico utilizados.

6.4.3 Parámetros de entrada

El algoritmo de asignación de productos a lotes finales se lanza para cada subconjunto resultante de la fase anterior. Para cada llamada al algoritmo se le pasan los siguientes parámetros y objetos:

Elementos del subconjunto: productos del subconjunto que serán asignados a lotes finales.

Distancia entre cada par de elementos del subconjunto: calculada con el concepto de distancia anteriormente explicado en el apartado 6.3.1.

Pesos de los diferentes componentes de la función de coste: la función de coste que el algoritmo utilizará para determinar la mejor solución dependerá de diferentes factores que se explicarán en detalle en el apartado siguiente. Para cada uno de esos factores es posible asignar un peso diferente para ponderar más unas características u otras.

Pesos mínimo y máximo de los lotes: vienen determinados por la limitación física derivada de enviar los lotes en diferentes medios de transporte: el valor máximo se corresponde con el máximo tamaño que un medio de transporte puede enviar, mientras que el valor mínimo es aquel para el que la empresa considera que no es rentable alquilar un medio de transporte. Por defecto estos valores son los mismos para todas las plantas (16 toneladas de límite mínimo y 24 toneladas de límite máximo), aunque al ser valores parametrizables se pueden modificar para adaptar los valores a un caso específico diferente.

Número de repeticiones: número de veces que se ejecutará cada heurístico.

Threshold: el límite utilizado para agrupar cada producto en el apartado anterior se utiliza de nuevo para garantizar que en los lotes finales no solamente la distancia entre cada par de productos sean menor que ese límite, sino entre todos los elementos que conforman el lote final.

6.4.4 Función de coste

El objetivo del algoritmo es minimizar el número de lotes, a la vez que se reducen el número de bobinas que quedan sueltas sin incluir en ningún lote y procurando que las bobinas que pertenecen al mismo lote sean lo más parecido posible. En función de estos criterios se ha diseñado una función de coste dependiente de los siguientes factores:

- 1) **Número de lotes (num_lotes):** número de lotes finales generados.
- 2) **Número de elementos sueltos (num_singles):** número de productos no asignados a ningún lote.

- 3) Media de las distancias máximas entre productos de un lote (dist):** para cada lote se calcula la distancia máxima entre los elementos del mismo, y luego se realiza la media de todas las distancias máximas de los lotes generados. A pesar de que cada elemento del subconjunto inicial ya es similar al resto de los productos del mismo conjunto gracias a todos los pasos realizados hasta este momento, al añadir de nuevo el concepto de distancia en la función de coste se intenta reforzar y mejorar la igualdad entre productos en los lotes finales. El coste de todos los lotes es:

$$dist_{cost} = media(max_{dist}(lote))$$

En caso de que el algoritmo no haya generado ningún lote, el valor que se introduce por defecto en la función de coste es de 100. De otra forma, soluciones sin ningún lote podrían parecer buenas a priori, ya que el valor de su función de coste si son pocos elementos sería muy bajo.

Asociado a cada factor existe un parámetro para ponderar más una característica u otra. La función de coste se puede expresar por lo tanto de la siguiente manera:

$$Total_{cost} = coef_{lotes} * num_{lotes} + coef_{singles} * num_{singles} + coef_{dist} * dist_{cost}$$

El algoritmo intentará encontrar la solución que minimice esta función de coste. Para resolver el problema de la tesis y comprobar el rendimiento del algoritmo todos los coeficientes tendrán valor 1, para que la parte principal de la solución sean el número de lotes y de elementos sueltos, y a igualdad de los mismos la media de las distancias máximas decida. Se propone como futuras líneas de investigación un estudio detallado de la influencia de estos parámetros, de cara a una posible optimización de los mismos en función de los resultados requeridos (más homogeneidad con más lotes / menos homogeneidad con menos lotes, etc.).

Se explican a continuación en detalle las 3 heurísticas y el meta-heurístico desarrollado para tratar de encontrar la mejor solución posible, atendiendo a la función de coste explicada.

6.4.5 Heurístico FIFO

El primer heurístico desarrollado sigue la metodología FIFO (first in – first out). La idea general del heurístico es ir recorriendo los elementos asignándolos a lote. La peculiaridad de este heurístico es que en el momento en el que un elemento no entra en el lote por su capacidad éste se cierra. Si el producto no cumple la condición de distancia con el resto de elementos del lote el algoritmo continúa buscando, ya que de otra manera serían muchos los casos que obligarían a cerrar los lotes “antes de tiempo” con tamaños lejanos a su capacidad máxima.

El funcionamiento del heurístico es el siguiente:

1. Se abre un lote
2. Se van recorriendo los diferentes productos en orden: para cada producto se comprueba si se puede añadir al lote tanto por la condición de tamaño (capacidad en el lote) como por la de distancia.
 - Si el producto cumple ambas condiciones se añade al lote, y el algoritmo continúa con el siguiente producto.
 - Si el producto cumple la condición de tamaño pero no de distancia se siguen comprobando los siguientes productos de la lista
 - Si el producto no cumple la condición de tamaño el lote se cierra.

El pseudocódigo del mismo sería:

```
HEURISTICO_FIFO (productos, distancias)
%Generación de un lote a partir de un listado de productos

Lote = [] %Se crea un nuevo lote

Para cada producto de productos
    size_condition, dist_condition = CAN_ADD(Lote, producto)
    % Comprobar si cumplen las condiciones para añadir el producto al lote

    Si size_condition = Verdadero
        Si dist_condition = Verdadero
            Añadir product a Lote
            Quitar product de la lista de elementos disponibles
        Si no
            Continuar %Comprobar siguiente elemento
    Si no
        Cerrar Lote

Fin %Devuelve el lote
```

Figura 26. Pseudocódigo del heurístico FIFO

6.4.6 Heurístico Greedy

La principal diferencia de este heurístico con respecto al anterior es que si un producto no entra por tamaño el lote no se cierra, sino que se sigue comprobando con el siguiente producto de la lista, y así sucesivamente hasta llegar al final de los productos.

El funcionamiento del heurístico es el siguiente:

1. Se abre un lote
2. Se van recorriendo los diferentes productos en orden: para cada producto se comprueba si se puede añadir al lote tanto por la condición de tamaño (capacidad en el lote) como por la de distancia.
 - Si el producto cumple ambas condiciones se añade al lote, y el algoritmo continúa con el siguiente producto.

- Si no el algoritmo continúa comprobando los siguientes elementos hasta que encuentre uno que se pueda añadir o, si no hay ninguno, se cierra el lote.

Debido a que los productos se ordenan de manera creciente en peso, decreciente y aleatorio, en el heurístico Greedy se realizan una serie de comprobaciones adicionales para evitar consumir tiempo innecesario, como son:

- 4) Si un elemento no cumple la condición de tamaño y están ordenados por peso creciente → no es necesario seguir comprobando el resto de elementos.
- 5) Si un elemento no cumple la condición de tamaño y están ordenados por peso decreciente → se comprueba que el tamaño del último elemento encajaría, si esto es así se sabe que al menos hay un elemento que cumple la condición de peso, con lo que el algoritmo seguiría.
 - Si el último elemento no cumple la condición de tamaño → ningún elemento intermedio lo cumplirá y podemos finalizar.

El pseudocódigo del heurístico sería:

```

HEURISTICO_GREEDY( productos, distancias )
%Generación de un lote a partir de un listado de productos

Lote = [] %Se crea un nuevo lote

Para cada producto de productos
size_condition, dist_condition = CAN_ADD(Lote, producto)
% Comprobar si cumplen las condiciones para añadir el producto al lote

Si size_condition = Verdadero y dist_condition = Verdadero
    Añadir product a Lote
    Quitar product de la lista de elementos disponibles
    % Si no cumple alguna condición se comprueba en función del orden si el
    algoritmo debe seguir
Si no
    Si order = none %Si están ordenados aleatoriamente
        Continuar %Se comprueba el siguiente elemento
    Si order = "smaller_first" y size_condition = Falso
        Cerrar Lote
    Si no mirar el último producto de la lista
        Si no se puede añadir al lote
            Cerrar lote

Fin %Devuelve el lote

```

Figura 27. Pseudocódigo del heurístico Greedy

6.4.7 Heurístico basado en distancias

El heurístico basado en distancias toma decisiones teniendo en cuenta la similitud de los elementos con aquellos que ya se encuentran en un lote, para intentar juntar elementos más homogéneos. El funcionamiento del heurístico es el siguiente:

1. Se abre un lote.
2. Se mete el primer elemento en el lote.
3. Se ordenan el resto de elemento por distancia creciente a los elementos del lote (cuando haya más de un producto en el lote se cogerá la máxima distancia de cada elemento restante a cada uno de los elementos del lote).
4. Se va recorriendo el conjunto ordenado de elementos comprobando para cada uno de ellos si se puede añadir al lote (tanto por distancia como por tamaño):

- 1) Si no se puede añadir → se busca el siguiente elemento hasta que no se pueda añadir ninguno → en ese caso se se cierra el lote
- 2) Si cumple las condiciones → se añade el producto al lote y se vuelve al paso anterior (nueva reorganización de elementos por distancias a los elementos actuales del lote y comprobación de si se puede meter alguno más).

El pseudocódigo del heurístico sería:

```
HEURISTICO_DISTANCIAS ( productos, distancias )
%Generación de un lote a partir de un listado de productos

Lote = [] %Se crea un nuevo lote

Mientras haya productos sin asignar
  Seleccionar el primer producto
  Asignar producto al lote
  Eliminar producto de products
  Para cada producto restante
    Ordenar los productos por distancia creciente a los productos del lote
    size_condition, dist_condition = CAN_ADD ( Lote, producto )
    % Comprobar si cumplen las condiciones para añadir el producto al lote
    Si dist_condition = Falso
      Cerrar lote
      Fin
    Si size_condition = Verdadero
      Añadir producto al lote
      Quitar producto de productos
  Fin Mientras
Fin %Devuelve el lote
```

Figura 28. Pseudocódigo del heurístico basado en distancias

6.4.8 Meta-heurístico GRASP

Se ha desarrollado un meta-heurístico para tratar de encontrar mejores soluciones. Por la naturaleza del problema, se ha decidido implementar un algoritmo GRASP (Greedy Randomized Adaptive Search Procedure). Es un meta-heurístico en dos fases utilizado para resolver problemas combinatorios complejos, como se muestra en (Alvarez-Valdés, et al., 2013), (Trivella & Pisinger, 2016) y (Rajkumar, et al., 2011). La primera consiste en la construcción de soluciones iniciales, mientras que la segunda fase es generalmente un proceso de búsqueda local para mejorar las soluciones de la fase anterior.

La Figura 29 muestra el esquema general del algoritmo GRASP:

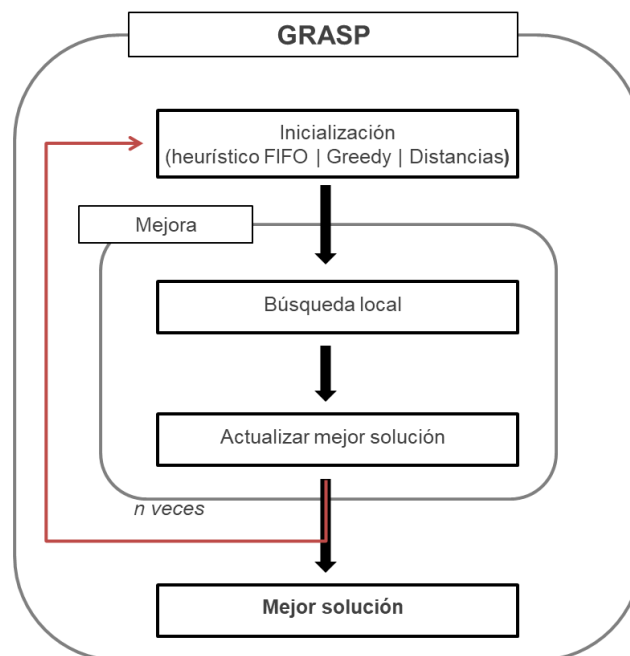


Figura 29. Esquema del algoritmo GRASP

Para la fase de construcción inicial de soluciones se pueden utilizar cualquiera de los heurísticos desarrollados en los apartados anteriores, para buscar buenas soluciones iniciales. Al final de cada iteración la solución del algoritmo se intentará mejorar con la búsqueda local de la siguiente fase.

Una vez obtenida una solución inicial, la fase de mejora realiza una búsqueda local y comprueba si las soluciones obtenidas explorando el vecindario de soluciones cercano mejoran a la mejor solución obtenida hasta ahora. Para la parte de búsqueda en el espacio de soluciones se han implementado la función “best_neighbour_fn”, que a su vez utiliza las funciones “intercambiar_productos” e “asignar_individuales”. Los esquemas de dichas funciones se pueden ver a continuación:

```

LOCAL_SEARCH (best_sol, cost_fn, best_neighbour_fn, num_runs)

%Almacenar mejor valor función objetivo
best_val = cost_fn(best_sol)

Mientras num_runs
    new_sol = best_neighbour_fn(best_sol)
    Si cost_fn(new_sol) < cost_fn(best_sol) %Si la nueva solución mejora la actual
        best_sol = new_sol % la nueva pasa a ser la mejor
        best_val = new_val
    Fin Si

Fin num_runs

Fin %Devuelve la mejor solución

```

Figura 30. Esquema de la búsqueda local del GRASP

```

best_neighbour_fn (sol)

%Llamada a las funciones “intercambiar_productos” e “asignar_individuales”

sol = intercambiar_productos (sol)
sol = asignar_individuales (sol)

Fin %Devuelve la solución

```

Figura 31. Función de búsqueda en vecindario

Dentro de la función “intercambiar_productos” se eligen elementos de lotes al azar y se intentan colocar en otros lotes, siempre y cuando se pueda realizar esta acción, teniendo en cuenta las restricciones de peso y distancia utilizadas anteriormente en los heurísticos. El esquema de la función es el siguiente:

- 1) Elegir un lote aleatorio
- 2) Elegir un producto aleatorio del lote
- 3) Recorrer el resto de lotes existentes, para cada lote, elegir un producto aleatorio y ver si ambos productos pueden ser intercambiables entre los lotes.
 - Si es posible → intercambiarlos
 - Si no es posible → seguir comprobando con el resto de lotes

El pseudocódigo es el siguiente:

```
intercambiar_productos (sol)

%Elegir un lote aleatorio
lote1 = random(sol.lotes)

%Elegir un producto aleatorio del lote
producto1 = random(lote.productos)

Para el resto de lotes de la solución
  %Elegir otro producto del lote
  producto2 = random (lotes.productos)

  size_cond_1, dist_cond_1 = CAN_ADD(lote1, producto2)
  size_cond_2, dist_cond_2 = CAN_ADD(lote2, producto1)

  % Comprobar si cumplen las condiciones para intercambiar los productos

  Si size_cond_1 = V y dist_cond_1 = V y size_cond_2 = V y dist_cond_2 = V
    Intercambiar productos entre lotes
  Fin Para
Fin %Devuelve la solución
```

Figura 32. Función para intercambiar productos entre lotes

Por último, el último paso es llamar a la función “asignar_individuales”, que intenta asignar aquellos elementos que han quedado sin asignar a alguno de los lotes de la nueva solución. El esquema de la función es el siguiente:

- 1) Recorrer todos los elementos que no están asignados a un lote
- 2) Ordenar los lotes aleatoriamente
- 3) Recorrer todos los lotes y comprobar si el elemento sin agrupar se puede incluir en algún lote

El pseudocódigo es el siguiente:

```
asignar_individuales(sol)
Para cada producto en productos_individuales
  Ordenar lotes aleatoriamente
  Para cada lote en lotes
    size_cond, dist_cond = CAN_ADD(lote, producto)
    Si size_cond= V y dist_cond = V
      Añadir producto al lote
  Fin Para
```

Figura 33. Función de asignación de elementos sueltos a lotes

En el siguiente capítulo se mostrarán una comparativa de resultados obtenidos con el algoritmo, tanto con las 3 heurísticas como con el metaheurístico GRASP. El modelo se probará tanto con instancias de prueba generadas a partir de las más comunes en la literatura, como con una instancia de una semana real.

7 Diseño de los experimentos de prueba

Debido a que no existe un problema similar al planteado en esta tesis, en este capítulo se proponen unas instancias de prueba a partir de las instancias de prueba del Bin Packing Problem genérico y del Bin Packing Problem con conflictos. El capítulo comienza precisamente con el análisis de las instancias utilizadas en la literatura para estos problemas, y a continuación se describe el procedimiento realizado para generar las instancias de prueba del HBPPMW.

7.1 Instancias de prueba en la literatura

En este apartado se resumen las principales instancias de prueba utilizadas en la literatura. Se especifican primero las instancias del problema general de empaquetado en recipientes, y a continuación se muestran las utilizadas para el caso con conflictos.

7.1.1 Instancias del problema de empaquetado en recipientes

En este apartado se recogen las principales instancias de prueba utilizadas en la versión clásica del problema. Tradicionalmente se han utilizado tres grupos de instancias (Bai, et al., 2007). El primero de ellos es propuesto en (Falkenauer, 1996), está disponible en la Operational Research Library (<http://people.brunel.ac.uk/~mastjib/jeb/orlib/binpackinfo.html>). Este grupo de conjuntos de datos consiste en 2 clases de problemas: uniforme y triple. En las clases uniformes, el número de productos es 120, 250, 500 y 1000, y sus tamaños están uniformemente distribuidos en el rango de [20, 100]. El tamaño de lote es de 150. Para cada tamaño de problema hay 20 instancias, siendo 80 instancias en total. En los conjuntos triples, la capacidad de los lotes es 1000, y los tamaños de productos están generados deliberadamente de manera que, en la solución óptima, cada lote contiene exactamente 3 productos (uno “grande” y dos “pequeños”), sin capacidad residual. El número de los productos es 60, 120, 249 y 501, y cada uno de ellos contiene 20 instancias. Este conjunto de datos es más complicado de resolver debido al hecho de que no se permite ninguna capacidad residual en la solución óptima del problema.

El segundo conjunto de instancias fue generado y estudiado por Scholl et al., y está disponible en <http://or.dei.unibo.it/library/bpplib>. Contiene 3 grupos (denominados Sch_Set1, Sch_Set2 y Sch_Set3), y un total de 1210 instancias. El número de productos varía de 50 a 500. Sch_Set1 son 720 instancias donde el número esperado medio de productos por lote no es mayor de 3; sin embargo, en Sch_Set2, el número medio de productos por lote varía entre 3, 5, 7 y 9. Los conjuntos de datos Sch_Set3 están considerados más complejos de resolver debido a que no existen dos artículos con el mismo tamaño (como sucede normalmente en el problema a resolver en esta tesis) (Scholl, et al., 1997).

El tercer grupo de conjuntos de instancias provienen del Euro Special Interest Group on Cutting and Packing, estando disponibles para descargar del siguiente enlace: http://paginas.fe.up.pt/~esicup/problem_generators. Este grupo de instancias representan una colección de instancias difíciles sacadas de un gran número de instancias de las publicaciones de (Wascher & Gau, 1996), (Schwerin & Wascher, 1997) y (Belov & Scheithauer, 2007).

7.1.2 Instancias del problema de empaquetado en recipientes con conflictos

En este apartado se enumeran las instancias de pruebas utilizadas por los artículos más relevantes hasta la fecha del problema de empaquetado en recipientes con conflictos.

En (Gendreau, et al., 2004), los autores parten de las instancias para BPP generadas por Falkenauer (Falkenauer, 1996). En concreto, seleccionan las 10 primeras instancias uniformes para cada valor de $n = 120, 250, 500$ y 1000 . Estas instancias contienen artículos con pesos discretos, y uniformemente distribuidos en $[20, 100]$, y capacidades de lote $c = 150$. Otro conjunto de instancias utilizadas parte también de las instancias triples de Falkenauer, generadas de manera que en la solución óptima cada lote contenga exactamente 3 artículos. Gendreau parte de las 10 primeras instancias para cada valor de $n = 60, 120, 249$ y 501 y multiplica el peso de los artículos por 10 para conseguir valores enteros, siendo la capacidad del lote $c = 1000$ (Gendreau, et al., 2004).

Muritiba et al. también parten de las instancias de Gendreau et al., y generan 10 grafos de conflictos aleatorios, caracterizados por diferentes valores de densidad entre 0 y 0.9 (Muritiba, et al., 2010). Para ello se asigna un valor p a cada vértice, según una distribución uniforme continua en $[0,1]$, generando un conflicto entre dos elementos i y j si $(p_i+p_j)/2 > \delta$. Esta misma estrategia e instancias son las utilizadas en (Maiza, et al., 2016).

En (Sadykov & Vanderbeck, 2013) realizan pruebas con las mismas instancias expuestas hasta ahora, y también generan unas propias, generando el grafo de conflictos aleatoriamente, empezando por un grafo vacío y seleccionando iterativamente un par aleatorio (i, j) hasta que se alcanza la densidad del grafo.

Yuan et al. utilizan directamente las instancias de Muritiba et al (Yuan, et al., 2014).

7.2 Generación de instancias de prueba para el HBPPMW

Para la generación de instancias de prueba para el problema propuesto, se tomarán como base las instancias uniformes de Falkenauer, ya que son las utilizadas por la mayoría de autores tanto en el BPP como en el HBPP. En dichas instancias el número de productos considerados son 120, 250, 500 y 1000, con los pesos uniformemente distribuidos en el rango de [20, 100], y un tamaño de lote de 150. Para cada tamaño de problema hay 20 instancias, siendo 80 instancias en total.

Para nuestro problema, además de los pesos de los productos y el tamaño de lote, son necesarios además valores para las siguientes características: subfamilia, grado de acero, espesor, ancho, espesor de recubrimiento (de ambos lados de la bobina) y aceitado. Se ha elegido como base la familia de bobinas galvanizadas (HD) ya que es muy representativa. Se asumen que todos los productos de las instancias tienen la misma familia (HD), forma (bobina), categoría y localización, y que están recubiertos por ambos lados. De esta forma, todos los productos de las instancias puedan tenerse en cuenta para realizar los lotes ya que todos comparten los mismos parámetros globales. Se detalla a continuación la manera de generar las características para las instancias:

- Peso: se cogen los pesos de las instancias de Falkenauer, que habían sido generados con una distribución uniforme en el rango [20, 100].
- Subfamilia: se han tomado como referencia 2 subfamilias, denominadas Z y ZM. Analizando datos de bobinas reales, se observa que un 92,27% de bobinas son de tipo Z, y el 7,73% correspondiente de tipo ZM. Tras analizar estos datos y consultar con expertos en el proceso, los valores correspondientes a la subfamilia se han generado aleatoriamente con valores enteros $\epsilon [0, 1]$, con las siguientes probabilidades:
 - 0 \rightarrow subfamilia tipo Z \rightarrow 90%
 - 1 \rightarrow subfamilia tipo ZM \rightarrow 10%

- Grado de acero: se han elegido 12 grados que representan la mayoría de grados dentro de esta subfamilia, denominados TYPE_DX51, TYPE_DX52, TYPE_DX53, TYPE_DX54, TYPE_DX56, TYPE_HCT590X, TYPE_HX700LAD, TYPE_S220GD, TYPE_S350GD, TYPE_S390GD, TYPE_S450GD, TYPE_S550GD. Analizando los grados de acero de bobinas reales non-prime para la familia tomada como representativa, se observa que el porcentaje de grados está dividido de la siguiente forma:

- TYPE_DX51 → 24,31%
- TYPE_DX52 → 0,55%
- TYPE_DX53 → 16,02%
- TYPE_DX54 → 19,34%
- TYPE_DX56 → 5,52%
- TYPE_HCT590X → 4,42%
- TYPE_HX700LAD → 2,21%
- TYPE_S220GD → 12,15%
- TYPE_S350GD → 4,97%
- TYPE_S390GD → 1,10%
- TYPE_S450GD → 3,87%
- TYPE_S550GD → 5,52%

Tras consultar con los expertos del proceso y confirmar la representatividad de estos valores, finalmente los grados de acero de las instancias se han generado aleatoriamente con valores enteros $\epsilon [0, 11]$, asignando a cada grado los siguientes valores con las siguientes probabilidades:

- 0 → TYPE_DX51 → 24 %
- 1 → TYPE_DX52 → 1%
- 2 → TYPE_DX53 → 16%
- 3 → TYPE_DX54 → 19%
- 4 → TYPE_DX56 → 6%

- 5 → TYPE_HCT590X → 4%
 - 6 → TYPE_HX700LAD → 2%
 - 7 → TYPE_S220GD → 12%
 - 8 → TYPE_S350GD → 5%
 - 9 → TYPE_S390GD → 1%
 - 10 → TYPE_S450GD → 4%
 - 11 → TYPE_S550GD → 6%
- **Espesor:** la Figura 34 muestra el espesor real de las bobinas sacadas a subasta para una planta concreta una semana (180 bobinas de la familia tomada como base). Se comprobó con otras 4 muestras de otras tantas semanas que el patrón observado era representativo:

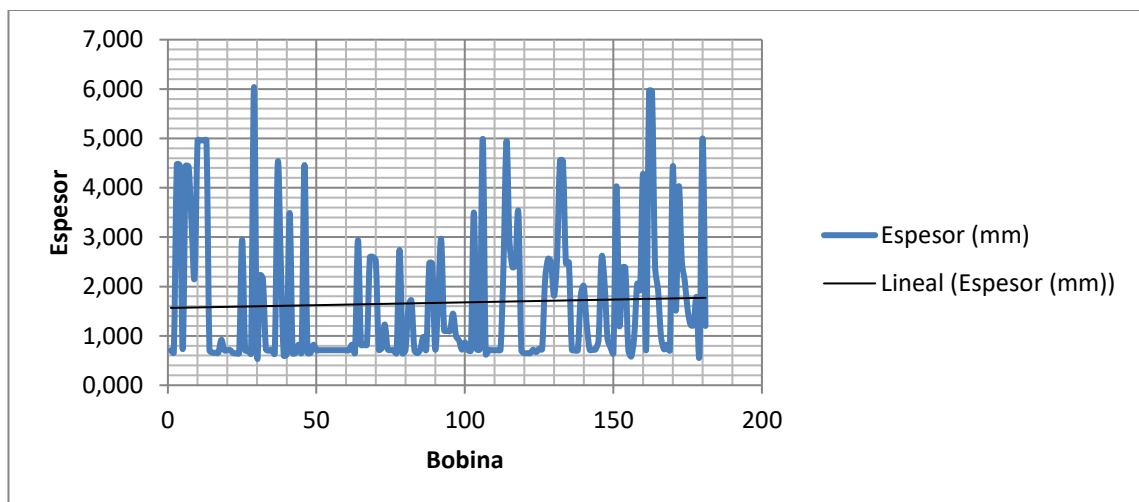


Figura 34. Valores de espesor de bobinas reales

Se ha utilizado un software (EasyFit) para tratar de determinar a qué tipo de distribución se ajustan más estos valores. El resultado obtenido es que la distribución de probabilidad que mejor se ajusta es una Kumaraswamy limitada entre $[0.56, 6]$ y con los parámetros $\alpha_1 = 0,44024$ y $\alpha_2 = 1,5529$. El estadístico de Kolmogorov Smirnov muestra una bondad de ajuste de $0,18729$. En la Figura 35 se representan la distribución de probabilidad de los datos y la de la distribución Kumaraswamy. Será esta distribución por lo tanto la que se utilice para generar los valores correspondientes al espesor de las instancias de prueba.

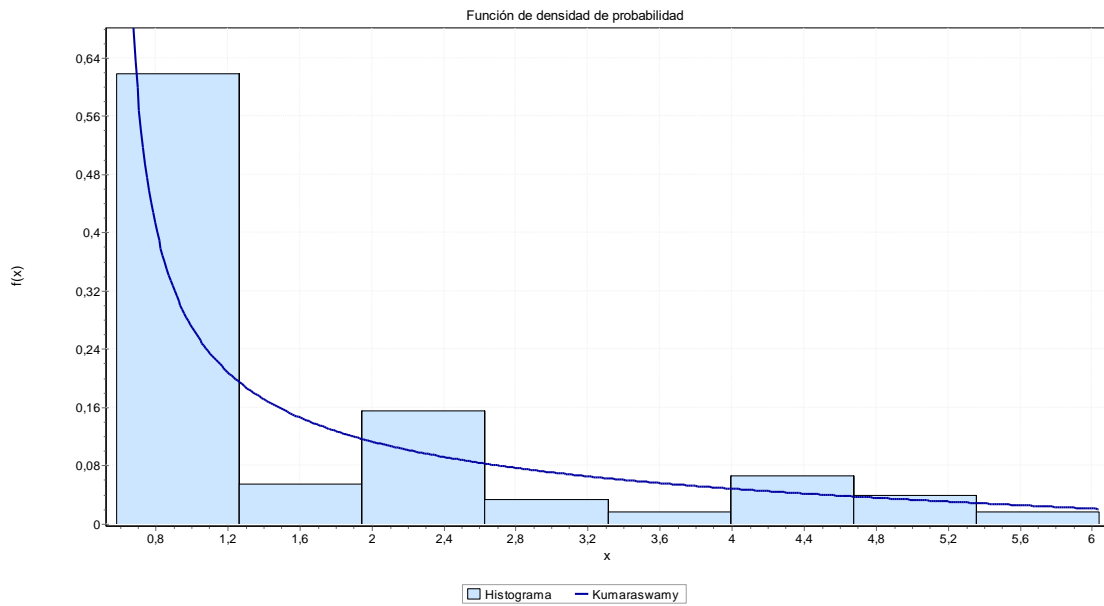


Figura 35. Función de densidad de probabilidad del espesor

- Ancho: la Figura 36 muestra el ancho de las bobinas de muestra utilizadas para tener una idea representativa de los valores mínimos y máximos a considerar:

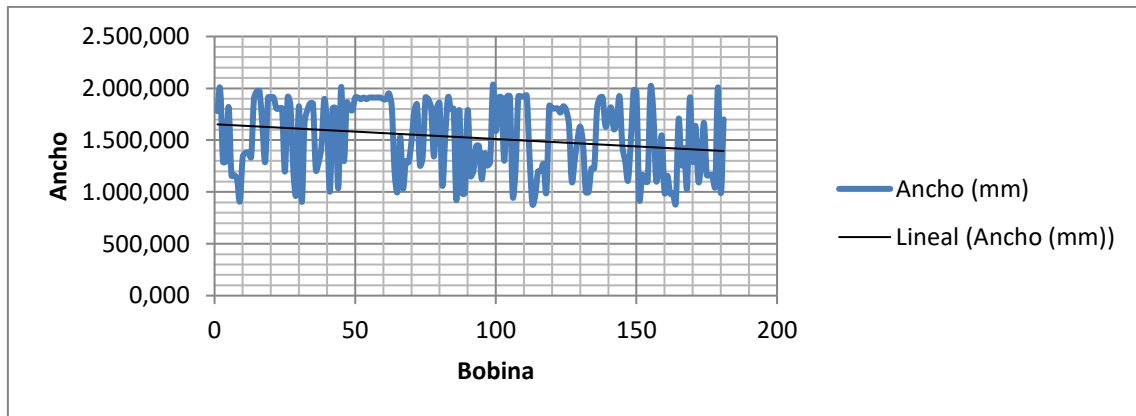


Figura 36. Valores de ancho de bobinas reales

De nuevo, aproximando los valores con una función de distribución se observa que la que mejor representa el comportamiento es una distribución de Pareto generalizada, con los parámetros $k=-1,4106$ $\sigma=1651,9$ y $\mu=838,65$. La bondad de ajuste de la distribución según

Kolmogorov Smirnov es de 0,10474. En la Figura 37 se representan la distribución de probabilidad de los datos y la de la distribución de Pareto generalizada. Se utilizará esta distribución por tanto para genera los valores correspondientes de las instancias de prueba.

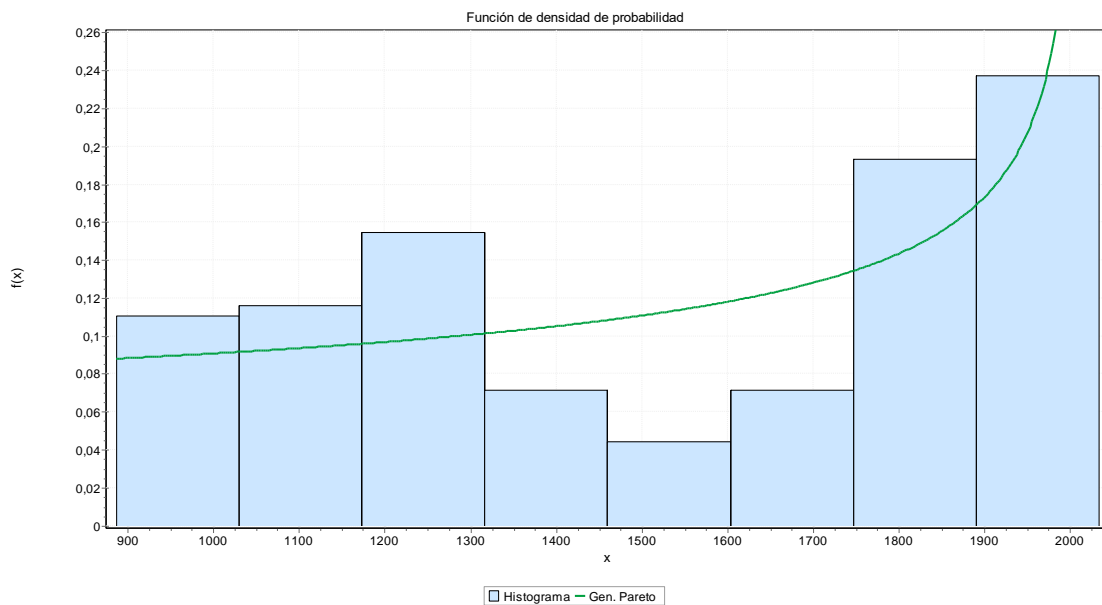


Figura 37. Función de densidad de probabilidad del ancho

- **Espesor de recubrimiento:** la Figura 38 y la Figura 39 muestran el espesor de recubrimiento de las bobinas de muestra utilizadas, para los lados 1 y 2 respectivamente, para tener una idea representativa de los valores mínimos y máximos a considerar:

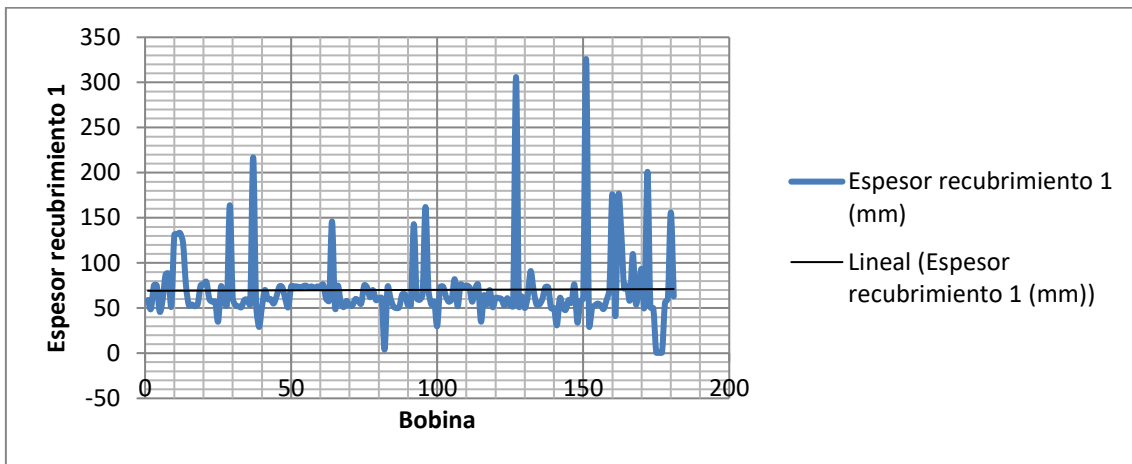


Figura 38. Valores de espesor de recubrimiento (lado 1) de bobinas reales

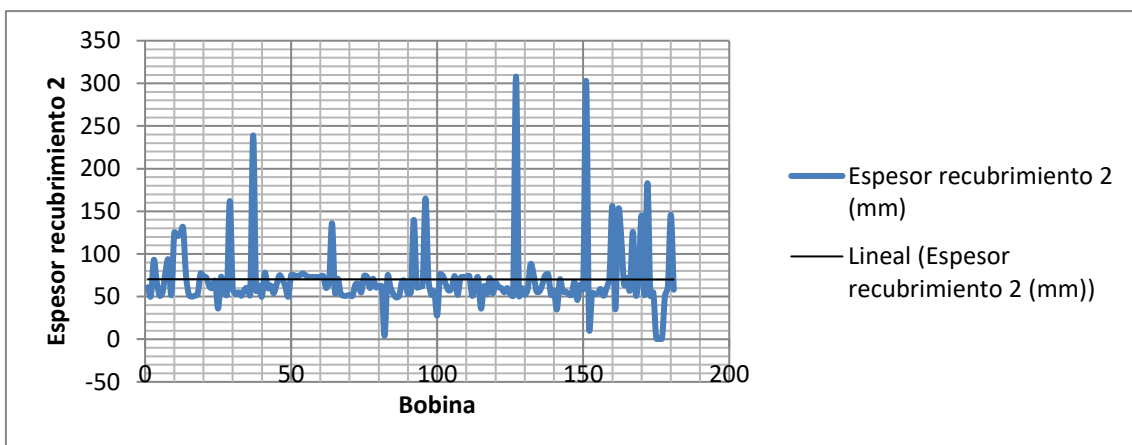


Figura 39. Valores de espesor de recubrimiento (lado 2) de bobinas reales

En ambos casos, se han ajustado los valores a distribuciones Dagum, con los siguientes parámetros:

- Espesor de recubrimiento 1 $\rightarrow k=1,208, \alpha=6,8828, \beta=90,655, \gamma=-31,139$
- Espesor de recubrimiento 2 $\rightarrow k=1,1793, \alpha=7,2491, \beta=94,253, \gamma=-33,785$

La bondad de ajuste según Kolmogorov Smirnov es de 0,162 para el lado 1, y de 0,175 para el lado 2. En la Figura 40 se representan la distribución de probabilidad de los datos y la

de la distribución de Dagum para el espesor de recubrimiento del lado 1, mientras que en la Figura 41 se muestra lo mismo para el espesor de recubrimiento del lado 2. Se utilizarán estas distribuciones por tanto para genera los valores correspondientes de las instancias.

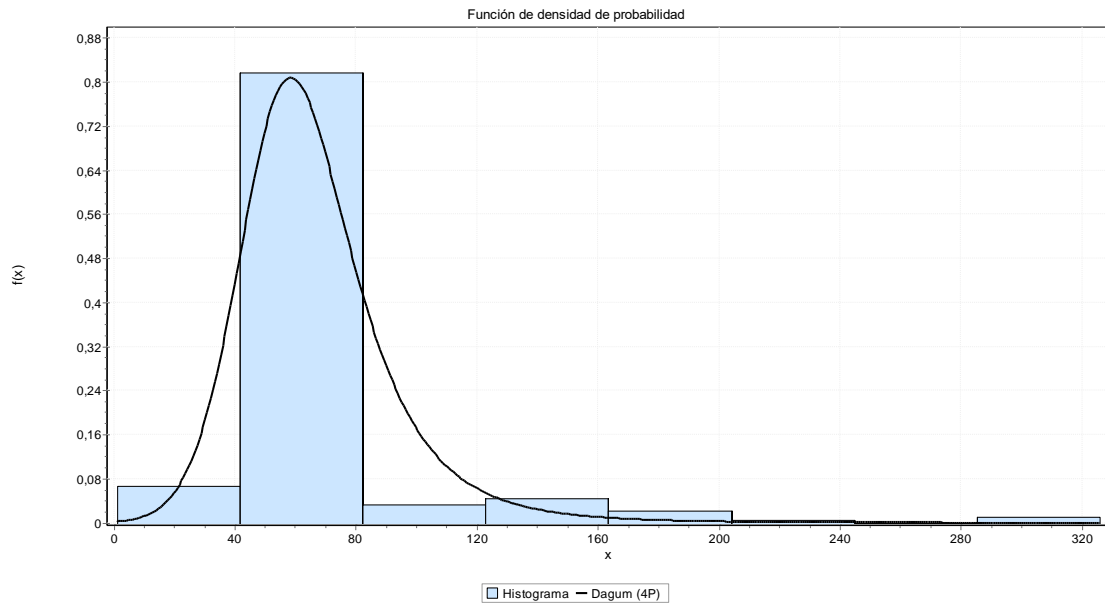


Figura 40. Función de densidad de probabilidad del espesor de recubrimiento lado 1

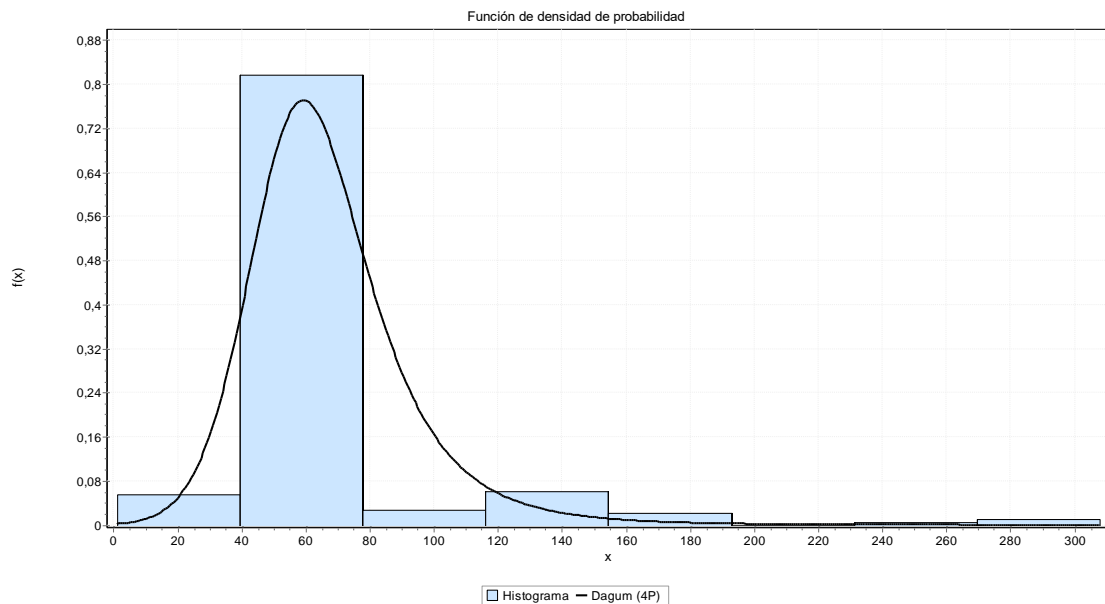


Figura 41. Función de densidad de probabilidad del espesor de recubrimiento lado 2

- Aceitado: la bobina puede estar con o sin aceitado. Del mismo conjunto de datos analizados, se observa que un 18,23% de las bobinas son sin aceitar, y el resto llevan algún tipo de aceitado. Tras confirmar con expertos del proceso la representatividad de estos porcentajes, finalmente los valores correspondientes al aceitado se han generado aleatoriamente con valores enteros $\in [0, 1]$, y con las siguientes probabilidades:
 - 0 \rightarrow sin aceitado \rightarrow 20%
 - 1 \rightarrow con aceitado \rightarrow 80%

Se han generado un total de 80 instancias \rightarrow 20 instancias con cada número de productos: 120, 250, 500 y 1000. En el Apéndice B del Anexo se muestra 1 ejemplo de instancia de prueba con 120 productos y otro con 250 productos. Todas las instancias de prueba están disponibles para descargar en

<https://www.dropbox.com/s/m1oo0jgzg9w358a/Instancias%20prueba%20HBBPMW.zip?dl=0>

8 Pruebas realizadas y validación de la solución

En este capítulo se analiza el comportamiento de la solución propuesta, con cada uno de los algoritmos explicados en el capítulo 6 para las instancias desarrolladas. Se evaluarán en primer lugar las soluciones obtenidas con los algoritmos FIFO, Greedy y basado en Distancias, y a continuación se compararán dichas soluciones con las obtenidas con el meta-heurístico GRASP.

Finalmente, se demostrará la utilidad de aplicación del algoritmo en un caso real, tomando todos los artículos de todas las plantas que deben de ser subastados en 1 semana, para comprobar el rendimiento y la calidad de las soluciones obtenidas con la solución propuesta.

8.1 Análisis del número de repeticiones de los heurísticos

Debido a la limitación de tiempo de ejecución del problema menor a 5 minutos para todas las plantas, previamente a realizar las pruebas con los heurísticos se determinará el número de repeticiones con las que se lanzarán los mismos, comparando el rendimiento de la función de coste con la evolución de tiempo. Para ello, se ha hecho una prueba inicial utilizando el algoritmo Greedy, por ser el más común en la literatura, junto con las instancias de prueba de 500 productos, con un número de iteraciones de 100, 200, 500 y 1000.

En la Tabla 10 se muestran los resultados obtenidos, mientras que en la Figura 42 se muestra la evolución del tiempo con el incremento de ejecuciones. Observando los valores de la función de coste, se observa que a medida que aumentan las repeticiones se obtienen mejores soluciones, sin embargo, la mejora de las soluciones es muy pequeña comparándola con el tiempo de ejecución; es decir, mientras que el tiempo de ejecución aumenta exponencialmente, el valor de la función objetivo mejora en torno a un 1%. Considerando

esto, se ha decidido que el número de repeticiones de los heurísticos con las que se realizarán las pruebas serán de 200, para permitir explorar en un espacio de soluciones que parece adecuado en un tiempo de computación razonable.

Instancia	100 iteraciones	200 iteraciones	500 iteraciones	1000 iteraciones
500_00.csv	449	450	447	446
500_01.csv	766	767	766	765
500_02.csv	245	247	243	243
500_03.csv	662	660	659	659
500_04.csv	356	355	354	353
500_05.csv	364	363	361	361
500_06.csv	550	548	548	547
500_07.csv	773	771	770	769
500_08.csv	442	441	439	439
500_09.csv	653	651	651	650
500_10.csv	660	659	658	657
500_11.csv	652	652	651	650
500_12.csv	349	348	346	346
500_13.csv	758	757	755	755
500_14.csv	558	553	555	553
500_15.csv	253	252	251	250
500_16.csv	560	557	556	556
500_17.csv	554	552	551	550
500_18.csv	775	773	770	770
500_19.csv	344	344	342	341

Tabla 10. Valores de la función de coste con 50, 100, 200 y 500 repeticiones del heurístico Greedy

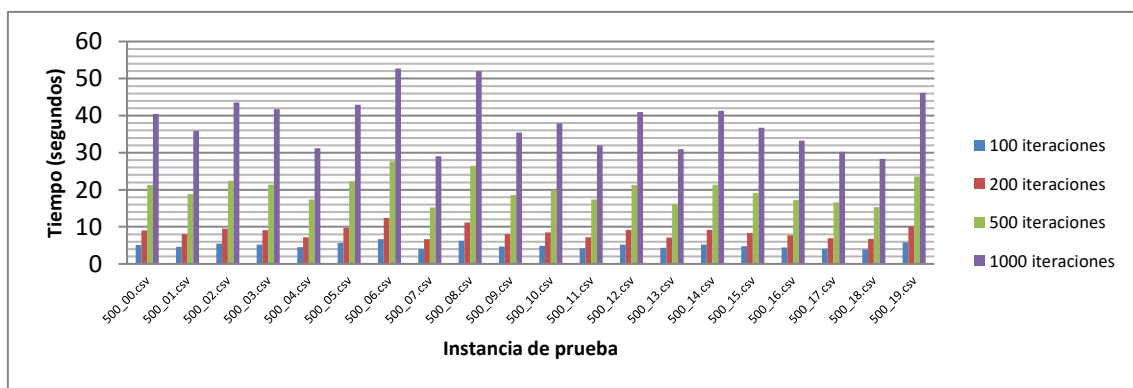


Figura 42. Evolución del tiempo de computación con 50, 100, 200 y 500 repeticiones

8.2 Experimentos realizados con las instancias de prueba

En este apartado se muestran los experimentos realizados con el algoritmo propuesto para cada una de las instancias de prueba generadas, y se muestran los resultados obtenidos con los mismos. En primer lugar se compararán tanto los resultados como el rendimiento del algoritmo con las 3 heurísticas desarrolladas, para posteriormente realizar las mismas pruebas con el meta-heurístico GRASP desarrollado, y poder dar unas conclusiones sobre el comportamiento del algoritmo. Los valores de los parámetros utilizados en las pruebas son los siguientes: Tamaño mínimo de lote: 100, Tamaño máximo de lote: 150, Threshold (distancia máxima permitida entre elementos del lote) = 1, MinSize = 5, MaxSize = 100. Pesos de los parámetros para calcular la distancia entre bobinas todos a 1.

8.2.1 Comportamiento de las 3 heurísticas

Se ha lanzado el algoritmo con cada una de las 80 instancias de prueba generadas, y a su vez para cada una de ellas el algoritmo se ha lanzado 3 veces, una con cada heurística generada para la tercera fase del algoritmo.

El número de veces que se lanza cada heurística ordenando los elementos aleatoriamente es de 200. Como se vio en el apartado anterior, con este número de iteraciones se conseguían buenos resultados tanto desde el punto de vista de la solución como de tiempo empleado. Debido al carácter aleatorio de las heurísticas, en realidad cada instancia de prueba no se ha ejecutado una vez con cada heurística, sino que se han realizado 10 ejecuciones. Los resultados que se presentan en el apartado siguiente para cada instancia de prueba y heurística se corresponden con la media de estas 10 ejecuciones.

En la Figura 43 se recuerda el funcionamiento del algoritmo global, con el número de veces que se ha lanzado cada parte en cada instancia de prueba:

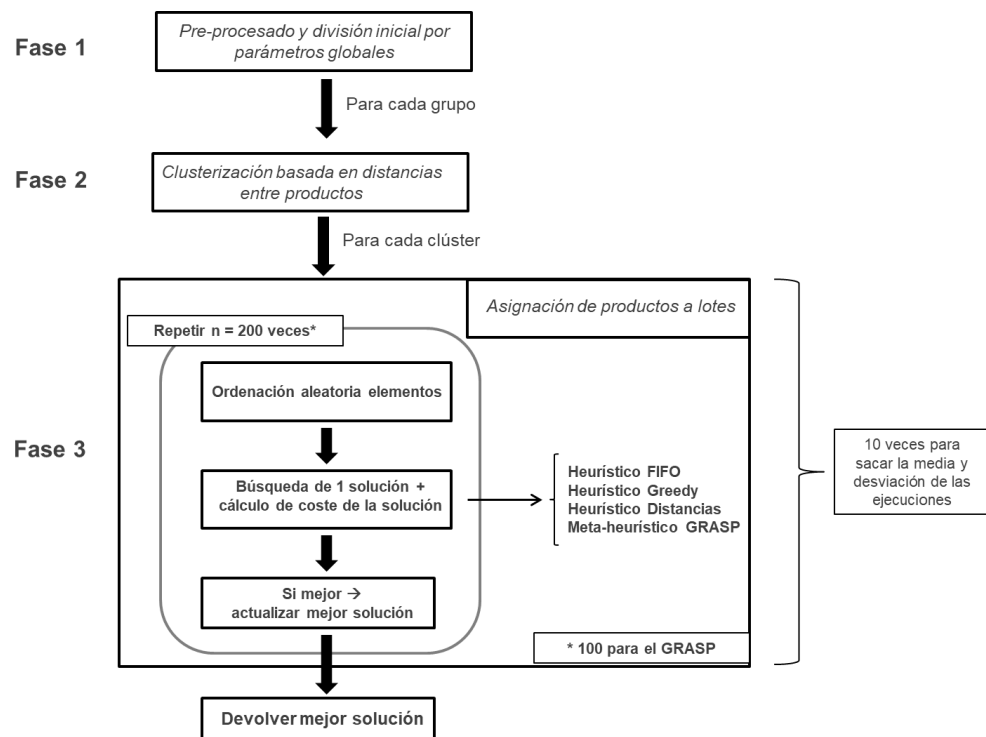


Figura 43. Esquema del algoritmo y número de repeticiones empleadas en las instancias de prueba

Las características utilizadas para comparar el rendimiento y la bondad de la solución son:

- Número de lotes
- Número de elementos sin asignar a lotes: hay que considerar aquí que, antes de lanzar cada heurística, ya hay algunas bobinas sin asignar de la fase anterior. Estas bobinas no son consideradas ni siquiera por los heurísticos, ya que la distancia que las separa con el resto ya las impide que sean asignadas a algún lote. Por supuesto, estas bobinas se contabilizan para dar el número de elementos sin asignar.
- Homogeneidad de los lotes: calculada como la media de las distancias máximas entre elementos de cada lote
- Resultado de la función de coste explicada en el apartado anterior. De nuevo aquí es importante destacar el hecho de que hay algunas bobinas que ya no son consideradas por los heurísticos. Estas bobinas no se considerarán por tanto para la función de coste, pero como se comentaba anteriormente, por supuesto se reportarán al final como elementos sin asignar a lotes.

- Tiempo empleado (segundos)

En el Apéndice C del Anexo se muestran los resultados de todas las ejecuciones, con la media de cada característica (lotes, elementos sin asignar, homogeneidad, función de coste, tiempo) y su desviación estándar.

8.2.1.1 Función de coste

En primer lugar se analizará la característica principal que define la bondad de la solución, que es el valor de la función de coste. En las siguientes tablas se muestra una comparativa del valor obtenido con cada heurístico, para cada una de las instancias de prueba.

120 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
120_00.csv	66.75	62.58	64.59	GREEDY
120_01.csv	66.36	63.35	62.50	DISTANCIAS
120_02.csv	64.08	59.99	59.45	DISTANCIAS
120_03.csv	67.10	61.93	64.36	GREEDY
120_04.csv	69.76	65.97	66.95	GREEDY
120_05.csv	74.18	70.92	67.37	DISTANCIAS
120_06.csv	167.63	165.12	164.35	DISTANCIAS
120_07.csv	67.99	64.77	64.58	DISTANCIAS
120_08.csv	67.07	62.60	64.40	GREEDY
120_09.csv	63.22	59.08	59.53	GREEDY
120_10.csv	72.56	68.52	70.40	GREEDY
120_11.csv	65.44	62.02	62.45	GREEDY
120_12.csv	65.35	60.77	60.33	DISTANCIAS
120_13.csv	64.33	59.77	62.36	GREEDY
120_14.csv	67.54	63.13	63.85	GREEDY
120_15.csv	70.56	66.21	65.85	DISTANCIAS
120_16.csv	71.13	67.59	69.77	GREEDY
120_17.csv	69.66	64.74	66.31	GREEDY
120_18.csv	68.47	65.18	65.83	GREEDY
120_19.csv	172.09	168.27	168.49	GREEDY

Tabla 11. Valores de la función de coste para cada heurístico con las instancias de 120 productos

250 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
250_00.csv	232.57	222.34	225.79	GREEDY
250_01.csv	141.07	132.54	134.24	GREEDY
250_02.csv	339.47	328.59	331.51	GREEDY
250_03.csv	237.60	228.71	232.06	GREEDY
250_04.csv	137.75	130.78	131.09	GREEDY
250_05.csv	235.98	224.77	227.19	GREEDY
250_06.csv	134.52	123.00	124.33	GREEDY
250_07.csv	134.44	121.29	126.54	GREEDY
250_08.csv	238.07	223.66	232.09	GREEDY
250_09.csv	135.14	120.35	130.80	GREEDY
250_10.csv	137.25	123.45	132.05	GREEDY
250_11.csv	134.20	119.85	127.55	GREEDY
250_12.csv	143.94	129.95	136.23	GREEDY
250_13.csv	234.24	221.12	227.46	GREEDY
250_14.csv	232.80	220.71	119.89	DISTANCIAS
250_15.csv	140.42	126.50	133.23	GREEDY
250_16.csv	128.76	116.62	121.78	GREEDY
250_17.csv	130.72	119.69	122.36	GREEDY
250_18.csv	131.72	119.00	124.00	GREEDY
250_19.csv	135.86	124.92	128.42	GREEDY

Tabla 12. Valores de la función de coste para cada heurístico con las instancias de 250 productos

500 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
500_00.csv	371.21	345.09	352.74	GREEDY
500_01.csv	678.03	657.38	659.63	GREEDY
500_02.csv	578.49	553.95	457.64	DISTANCIAS
500_03.csv	480.80	453.92	456.43	GREEDY
500_04.csv	587.55	567.04	569.80	GREEDY
500_05.csv	382.26	354.74	363.98	GREEDY
500_06.csv	580.10	554.49	564.00	GREEDY
500_07.csv	790.63	768.52	466.53	DISTANCIAS
500_08.csv	368.80	349.04	346.97	DISTANCIAS
500_09.csv	472.58	446.49	450.88	GREEDY
500_10.csv	676.39	652.97	557.47	DISTANCIAS
500_11.csv	674.65	654.77	656.89	GREEDY
500_12.csv	371.15	344.47	250.32	DISTANCIAS
500_13.csv	467.75	443.53	446.08	GREEDY
500_14.csv	376.38	350.69	358.02	GREEDY
500_15.csv	371.09	342.78	352.58	GREEDY
500_16.csv	472.02	450.36	458.29	GREEDY
500_17.csv	673.81	654.70	551.17	DISTANCIAS
500_18.csv	892.98	875.27	873.19	DISTANCIAS
500_19.csv	467.26	441.01	452.18	GREEDY

Tabla 13. Valores de la función de coste para cada heurístico con las instancias de 500 productos

1000 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
1000_00.csv	1262.28	1213.43	1123.97	DISTANCIAS
1000_01.csv	1268.40	1217.99	1124.37	DISTANCIAS
1000_02.csv	1166.12	1112.38	1036.23	DISTANCIAS
1000_03.csv	858.57	796.97	819.89	GREEDY
1000_04.csv	945.11	894.19	899.18	GREEDY
1000_05.csv	944.50	891.81	905.46	GREEDY
1000_06.csv	1055.24	1002.74	802.45	DISTANCIAS
1000_07.csv	1051.62	995.79	915.25	DISTANCIAS
1000_08.csv	956.44	908.37	818.54	DISTANCIAS
1000_09.csv	740.12	684.07	603.46	DISTANCIAS
1000_10.csv	1051.91	1003.27	919.71	DISTANCIAS
1000_11.csv	1053.58	1002.96	909.94	DISTANCIAS
1000_12.csv	1250.29	1197.85	797.69	DISTANCIAS
1000_13.csv	939.41	880.74	901.22	GREEDY
1000_14.csv	931.75	881.68	795.76	DISTANCIAS
1000_15.csv	1052.29	999.42	915.76	DISTANCIAS
1000_16.csv	946.19	892.28	907.88	GREEDY
1000_17.csv	955.86	898.24	819.27	DISTANCIAS
1000_18.csv	1863.33	1811.75	1312.46	DISTANCIAS
1000_19.csv	956.43	905.72	818.87	DISTANCIAS

Tabla 14. Valores de la función de coste para cada heurístico con las instancias de 1000 productos

El principal resultado llamativo es que el algoritmo FIFO no ha conseguido la mejor solución en ninguna de las instancias de prueba. El algoritmo GREEDY alcanzó la mejor solución en 50 de las 80 instancias, mientras que el algoritmo de Distancias consigue las mejores soluciones para las otras 30 instancias. La Figura 44 muestra la distribución de mejores soluciones obtenidas para cada uno de los tamaños de instancias:

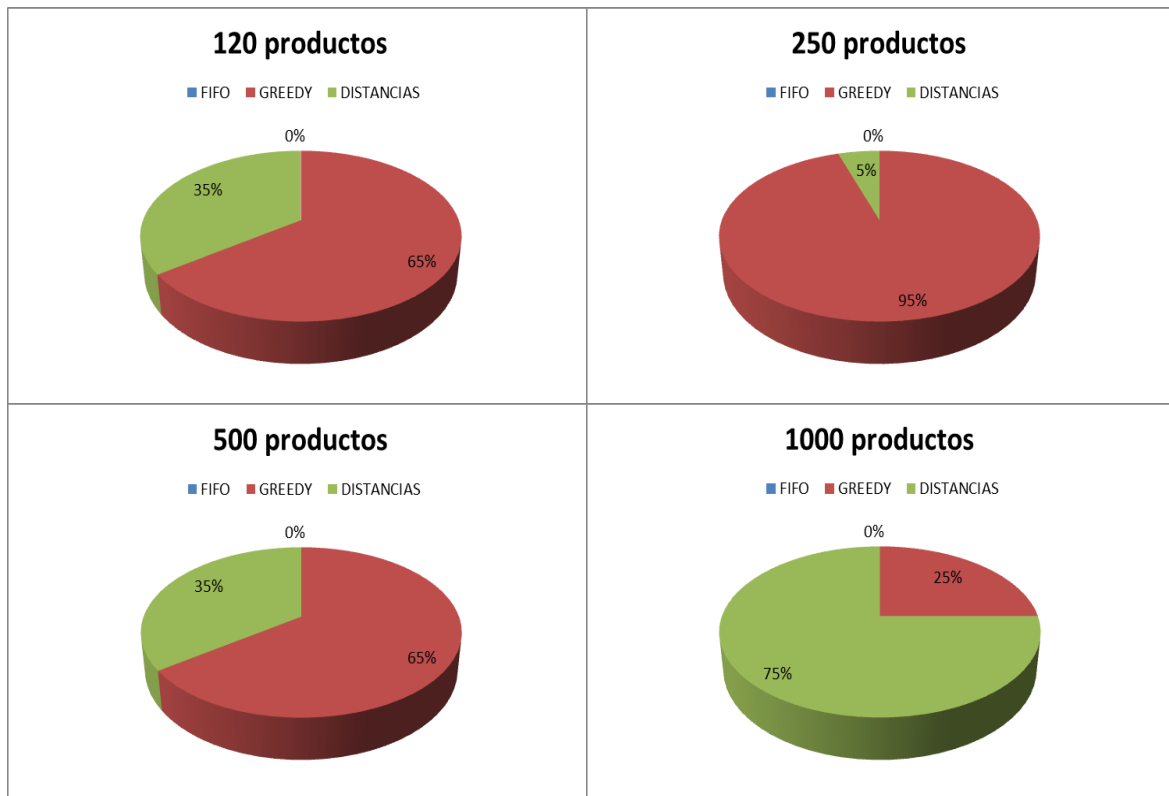


Figura 44. Distribución de mejores soluciones para cada tamaño de instancias

Ya que la función de coste depende por un lado del número de lotes y de productos que han quedado sin asignar, y por otro de la homogeneidad de los lotes generados, se realiza a continuación un estudio desglosado de los mismos, analizando el comportamiento de las heurísticas en cada instancia.

8.2.1.2 Número de lotes y productos sin asignar

A continuación se presentan unas tablas comparativas del número de lotes generados para cada heurístico, y del número de productos que no se han asignado a ningún lote, para cada grupo de instancias.

Resultados en cuanto a lotes y productos sueltos con las instancias de 120 productos:

	FIFO		GREEDY		DISTANCIAS	
	# sin asignar	# lotes	# sin asignar	# lotes	# sin asignar	# lotes
120_00.csv	18.9	43.7	14.20	45.10	16.10	42.90
120_01.csv	18.8	43.3	17.30	42.30	15.90	41.20
120_02.csv	16.8	42.5	13.20	42.30	12.00	42.00
120_03.csv	20	43.4	12.50	46.50	15.60	43.40
120_04.csv	22.6	42.2	18.70	43.30	21.00	41.00
120_05.csv	28.9	40	24.30	41.60	15.00	44.00
120_06.csv	20	42.5	18.30	42.40	11.20	44.90
120_07.csv	18.7	44.4	16.70	44.40	13.00	45.00
120_08.csv	19.9	44.6	15.00	46.00	19.00	42.00
120_09.csv	17.6	41.6	13.00	42.70	11.00	43.00
120_10.csv	22.6	44.1	17.30	45.70	18.50	44.50
120_11.csv	22.6	42.4	20.80	41.60	18.00	42.00
120_12.csv	13.5	47.2	10.10	46.80	9.50	46.50
120_13.csv	12.6	47.3	10.10	46.60	16.60	41.60
120_14.csv	17.4	45.6	13.70	45.90	14.90	44.10
120_15.csv	25.2	39.5	18.70	42.30	19.00	39.00
120_16.csv	22.3	43.8	17.40	45.60	21.00	43.00
120_17.csv	18.7	46.8	12.90	48.10	15.00	46.00
120_18.csv	17.9	45	14.30	45.60	16.00	43.00
120_19.csv	27.4	39.9	24.00	41.00	23.00	40.00

Tabla 15. Lotes y productos sin asignar para cada heurístico con las instancias de 120 productos

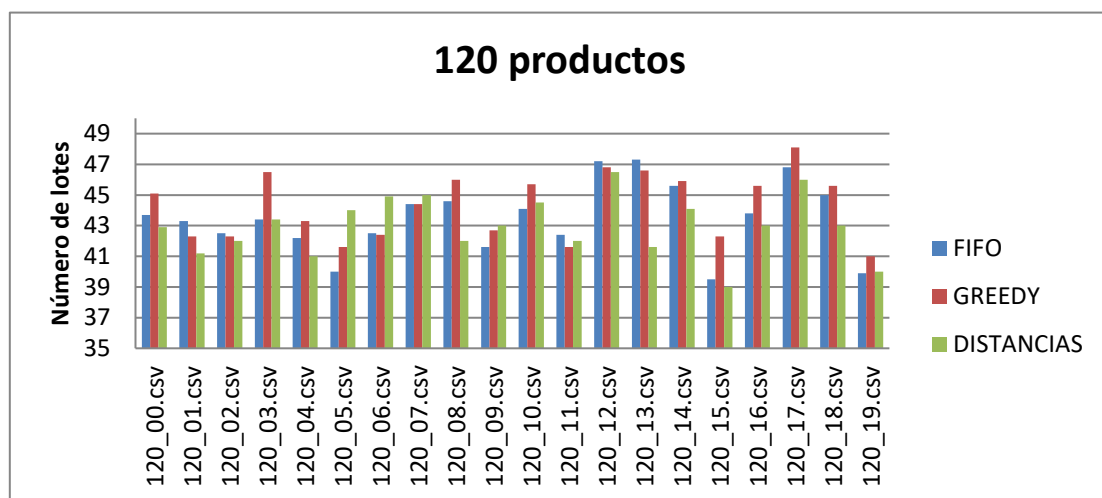


Figura 45. Comparativa de lotes generados para las instancias de 120 productos

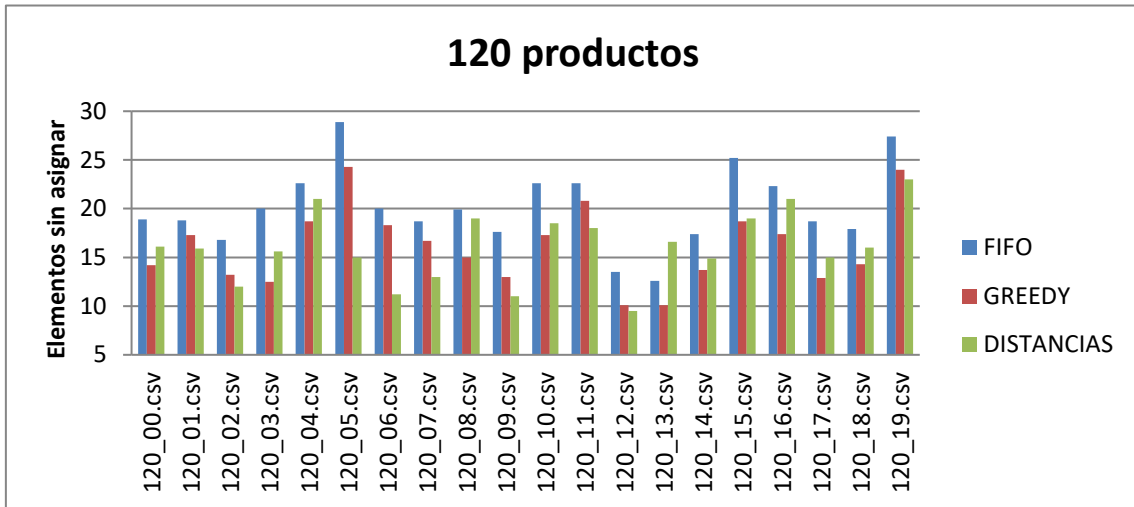


Figura 46. Comparativa de elementos sin asignar para las instancias de 120 productos

Si agrupamos tanto el número de lotes como el número de elementos sueltos, se obtiene la siguiente gráfica:

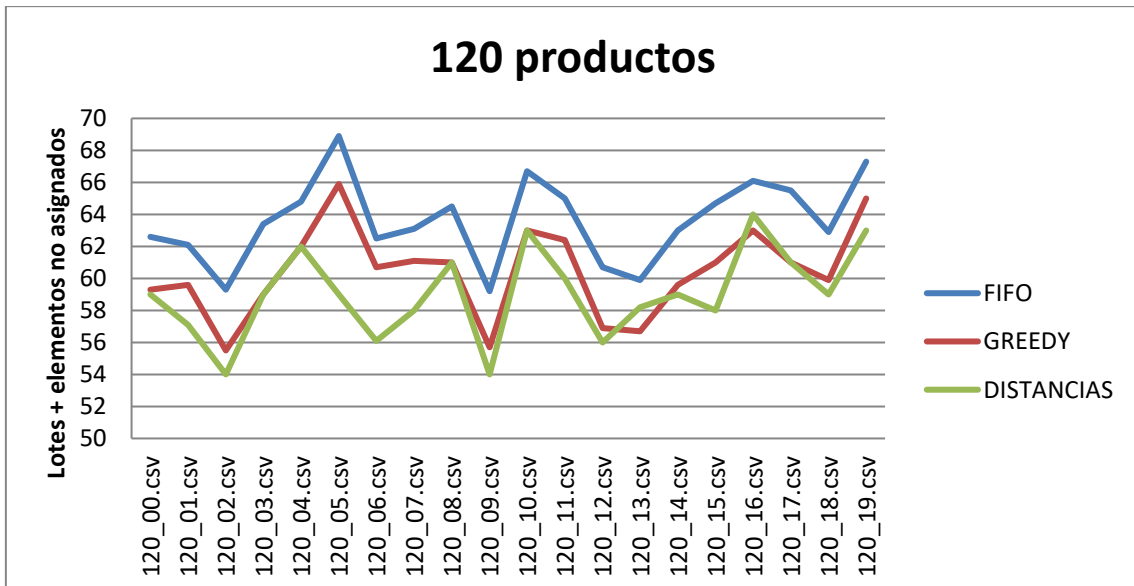


Figura 47. Comparativa de lotes + elementos sin asignar para las instancias de 120 productos

Atendiendo solamente al número de lotes, el algoritmo de Distancias es el que mejores resultados presenta, seguido del FIFO. Sin embargo, el FIFO deja muchos elementos sin asignar. Considerando los 2 factores, el algoritmo de distancias es el mejor, seguido de cerca del algoritmo Greedy.

Resultados en cuanto a lotes y productos sueltos con las instancias de 250 productos:

	FIFO		GREEDY		DISTANCIAS	
	# sin asignar	# lotes	# sin asignar	# lotes	# sin asignar	# lotes
250_00.csv	33.9	90.9	24.20	91.60	27.00	89.00
250_01.csv	41	91.3	33.60	91.10	36.30	85.80
250_02.csv	46.6	86.9	36.60	87.20	40.30	83.70
250_03.csv	40.2	91	32.20	91.40	37.20	84.20
250_04.csv	33.2	95.7	26.00	97.30	26.00	93.00
250_05.csv	39.3	91.3	28.20	93.60	32.00	89.00
250_06.csv	40.1	90.8	27.00	93.70	29.10	89.90
250_07.csv	33.5	96.8	18.70	100.20	31.00	91.00
250_08.csv	43.4	92.3	23.30	99.70	41.00	87.00
250_09.csv	38.6	90.7	17.50	97.60	41.00	82.00
250_10.csv	37.3	95.7	19.10	101.50	43.40	83.60
250_11.csv	33.4	94.6	16.50	98.10	35.20	84.80
250_12.csv	48.5	89.9	29.00	96.70	46.40	83.20
250_13.csv	38.7	93.6	24.60	96.30	41.40	83.60
250_14.csv	38.6	91.8	25.20	94.00	26.50	88.70
250_15.csv	50.9	87.7	29.30	95.90	47.20	81.90
250_16.csv	36.6	89.7	25.50	90.50	37.80	80.20
250_17.csv	30.4	94.4	21.50	94.00	27.00	89.00
250_18.csv	39.3	91.1	24.90	93.80	35.30	85.80
250_19.csv	37.1	93.5	23.60	97.10	29.70	91.30

Tabla 16. Lotes y productos sin asignar para cada heurístico con las instancias de 250 productos

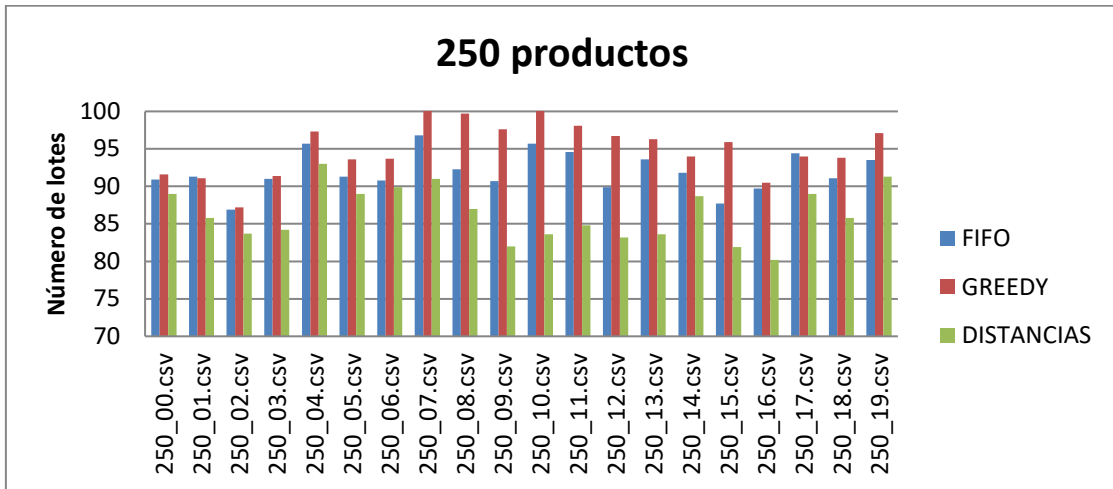


Figura 48. Comparativa de lotes generados para las instancias de 250 productos

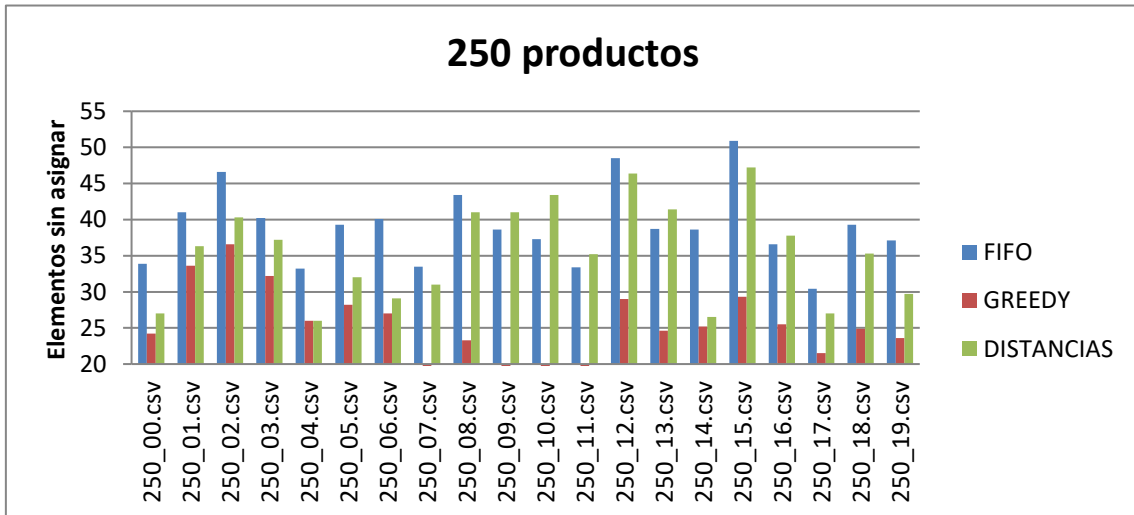


Figura 49. Comparativa de elementos sin asignar para las instancias de 250 productos

Como en el caso de 120 productos, si agrupamos tanto el número de lotes como el número de elementos sueltos, se obtiene la siguiente gráfica:

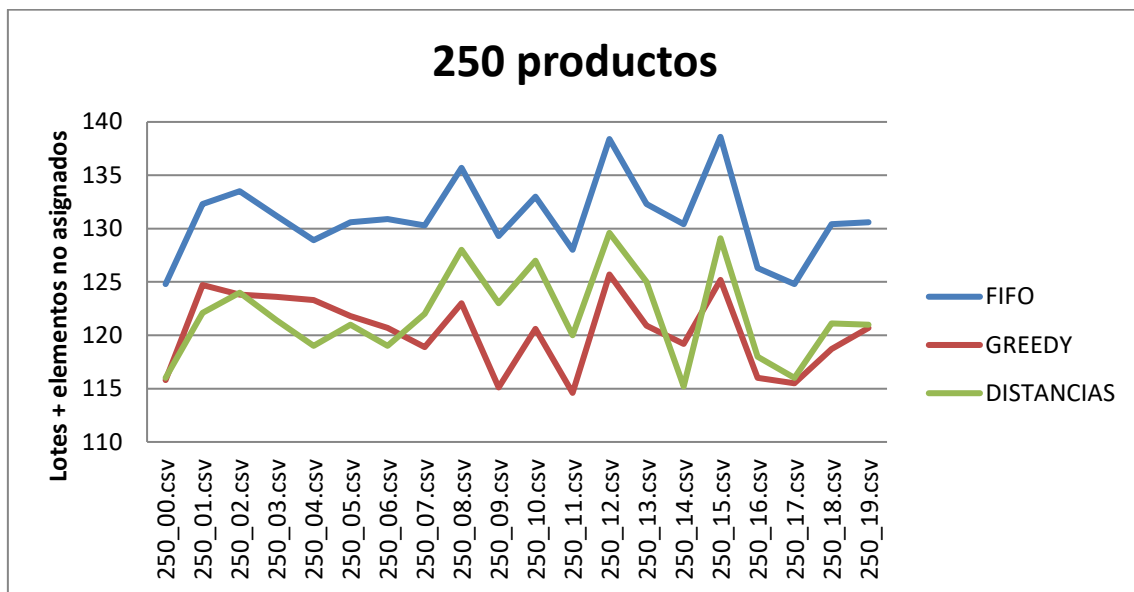


Figura 50. Comparativa de lotes + elementos sin asignar para las instancias de 250 productos

De igual manera que en las instancias de 120 productos, con las instancias de 250 productos se comprueba que el algoritmo de Distancias presenta menor número de lotes, con un comportamiento también bueno del Greedy. Atendiendo solamente al número de elementos sin asignar el Greedy destaca claramente. Si se consideran los dos factores, la mejor solución se obtiene con el algoritmo de Distancias, siendo muy bueno también el rendimiento del Greedy.

Resultados en cuanto a lotes y productos sueltos con las instancias de 500 productos:

	FIFO		GREEDY		DISTANCIAS	
	# sin asignar	# lotes	# sin asignar	# lotes	# sin asignar	# lotes
500_00.csv	82.7	177.1	49.30	187.50	79.30	162.00
500_01.csv	96	171.9	75.00	174.30	85.10	162.00
500_02.csv	90.7	177.7	59.80	185.90	80.20	166.10
500_03.csv	100.3	170.4	64.50	181.70	72.90	170.50
500_04.csv	112.3	167.1	84.20	177.20	102.40	159.70
500_05.csv	101.1	174.5	60.10	190.00	88.80	165.00
500_06.csv	86.9	182.8	52.40	194.00	84.60	167.60
500_07.csv	110.1	170.3	85.00	176.00	92.20	161.20
500_08.csv	91.4	168.9	73.90	169.20	71.30	164.30
500_09.csv	80.7	182.8	51.70	188.30	68.10	172.20
500_10.csv	94.6	172	66.10	179.70	80.80	165.20
500_11.csv	95.8	168.2	76.60	169.90	83.70	161.30
500_12.csv	89.3	176	53.20	187.80	77.30	163.80
500_13.csv	77.5	179.6	52.10	183.80	64.50	169.70
500_14.csv	100.4	170.3	68.60	178.70	91.80	159.30
500_15.csv	80.9	181.2	43.80	191.10	74.80	167.30
500_16.csv	85.8	179	59.60	186.10	80.00	168.30
500_17.csv	99.2	167.4	78.80	170.60	77.20	164.80
500_18.csv	116.7	164.8	97.40	168.90	109.00	151.00
500_19.csv	78.3	177.9	50.00	182.70	84.10	155.40

Tabla 17. Lotes y productos sin asignar para cada heurístico con las instancias de 500 productos

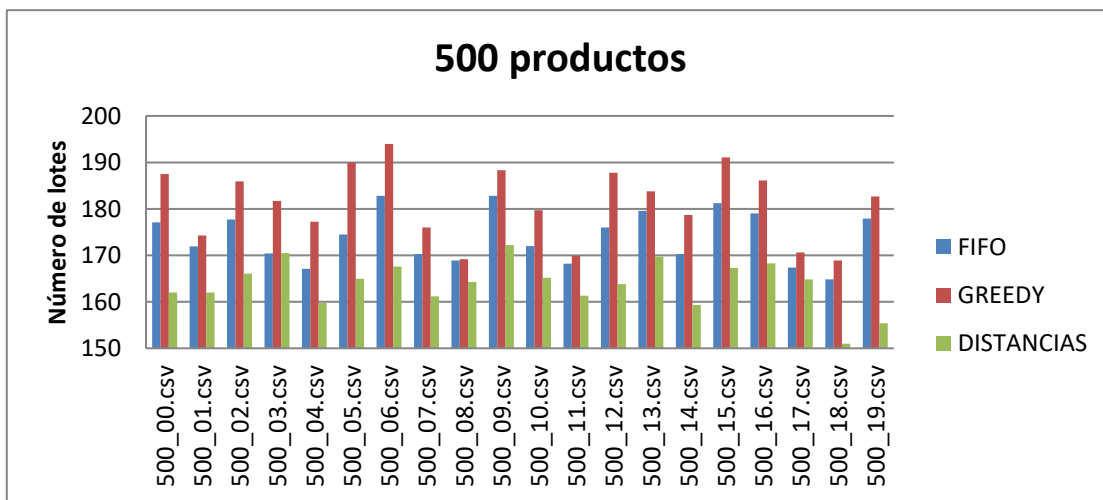


Figura 51. Comparativa de lotes generados para las instancias de 500 productos

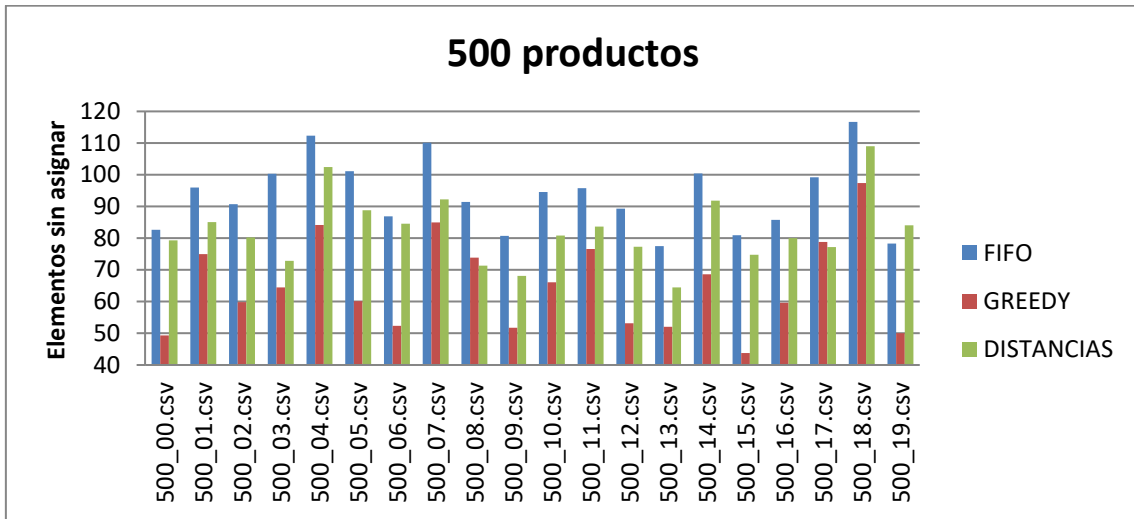


Figura 52. Comparativa de elementos sin asignar para las instancias de 500 productos

Agrupando el número de lotes como el número de elementos sueltos, se obtiene:

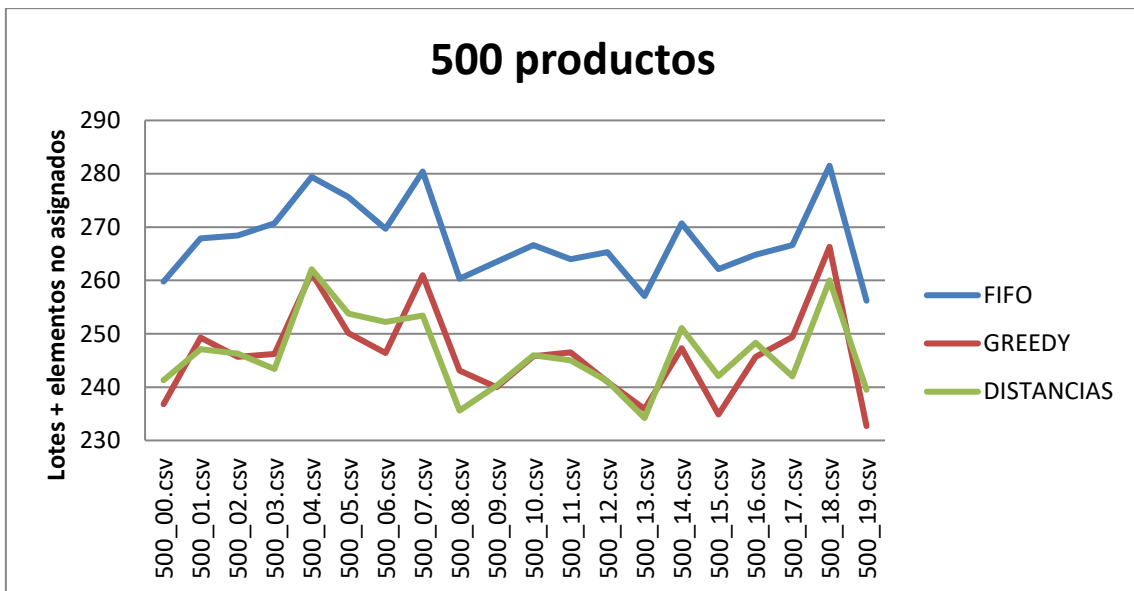


Figura 53. Comparativa de lotes + elementos sin asignar para las instancias de 500 productos

Con las instancias de 500 productos se repite la misma tendencia: algoritmo de Distancias produce menos lotes, algoritmo Greedy deja pocos elementos sin asignar.

Resultados en cuanto a lotes y productos sueltos con las instancias de 1000 productos:

	FIFO		GREEDY		DISTANCIAS	
	# sin asignar	# lotes	# sin asignar	# lotes	# sin asignar	# lotes
1000_00.csv	211	336.5	150.00	352.30	202.90	304.50
1000_01.csv	218.6	334	151.00	355.10	191.30	315.80
1000_02.csv	200.6	348.9	126.70	372.30	198.20	319.20
1000_03.csv	188.7	355.4	101.80	383.70	179.20	322.60
1000_04.csv	195.7	336.3	134.50	349.60	159.50	324.40
1000_05.csv	185.2	344	119.10	361.50	164.50	324.80
1000_06.csv	215.8	327.4	147.40	346.40	176.50	312.20
1000_07.csv	184.2	351.9	109.00	376.00	180.60	317.50
1000_08.csv	223	320.5	159.80	339.80	206.80	299.10
1000_09.csv	193.2	337	115.30	361.80	182.40	309.70
1000_10.csv	203.8	334.4	142.80	349.60	192.80	309.00
1000_11.csv	199.8	342.6	135.10	360.70	165.40	327.20
1000_12.csv	193.8	340.9	131.20	355.60	159.80	319.70
1000_13.csv	173.7	351.5	96.70	373.10	151.80	330.60
1000_14.csv	164.5	352.5	103.90	366.50	151.30	326.40
1000_15.csv	195.7	341.1	123.50	363.60	181.80	315.00
1000_16.csv	178.5	354.8	112.50	369.80	163.90	328.60
1000_17.csv	210.6	334.2	132.00	359.40	203.70	303.00
1000_18.csv	213.2	335.4	151.00	349.70	189.00	305.40
1000_19.csv	206.3	333.9	143.60	349.30	187.60	312.80

Tabla 18. Lotes y productos sin asignar para cada heurístico con las instancias de 1000 productos

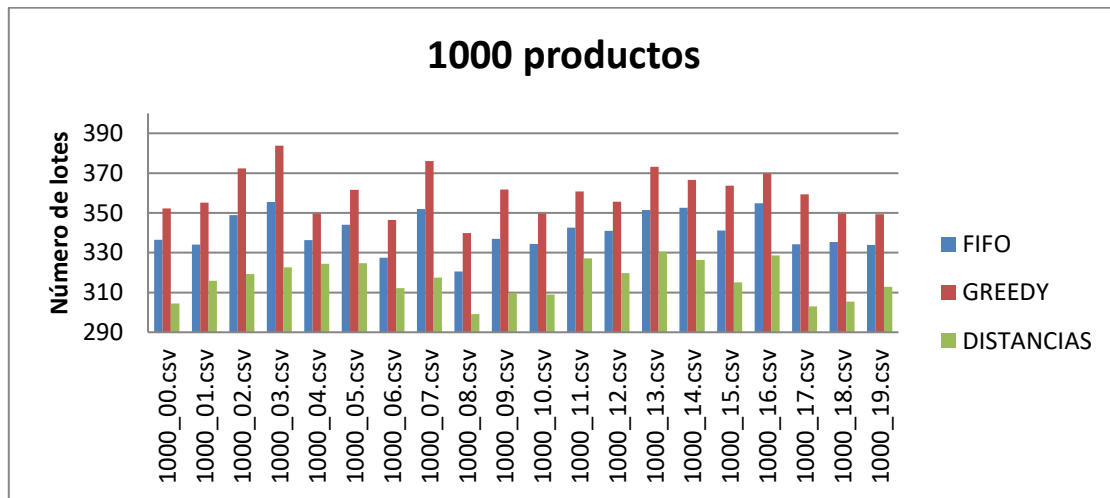


Figura 54. Comparativa de lotes generados para las instancias de 1000 productos

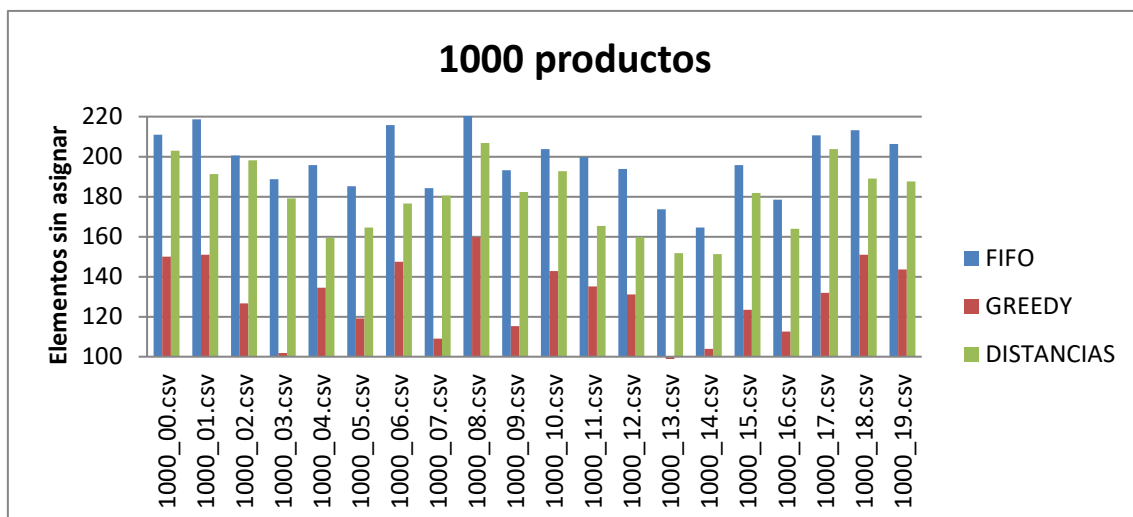


Figura 55. Comparativa de elementos sin asignar para las instancias de 1000 productos

Como en los casos anteriores, si se agrupa el número de lotes como el número de elementos sueltos, se obtiene la siguiente gráfica:

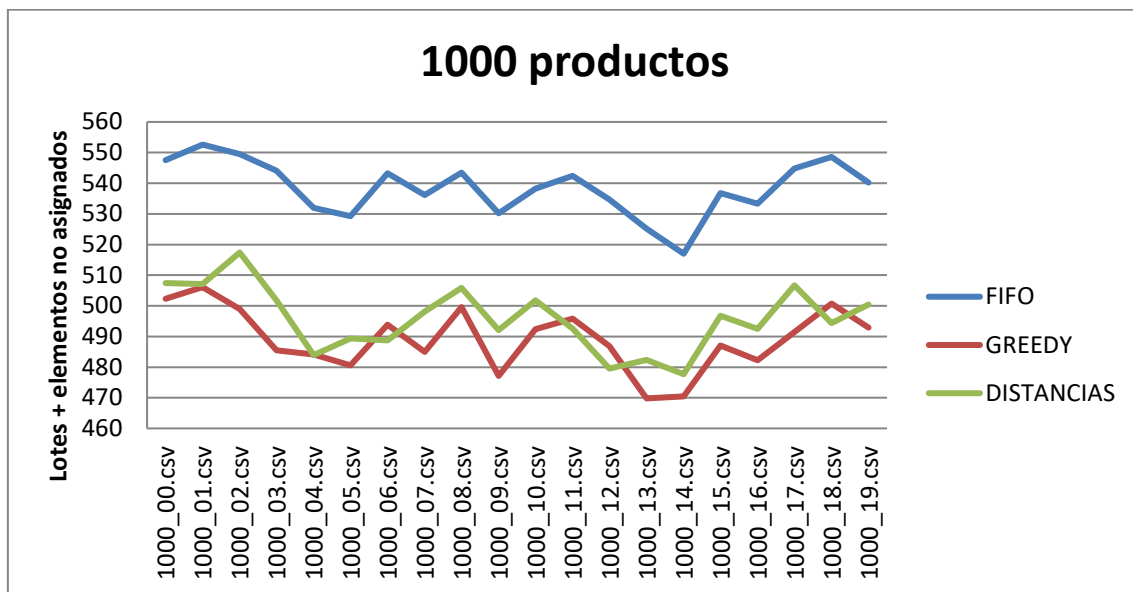


Figura 56. Comparativa de lotes + elementos sin asignar para las instancias de 1000 productos

De nuevo, el algoritmo Greedy deja pocos elementos sin asignar, mientras que el basado en Distancias produce muy pocos lotes. Para las instancias de 1000 productos la combinación de ambos hace que los pocos elementos sin asignar a lotes del algoritmo Greedy prevalezcan, siendo el mejor en la mayoría de instancias.

Como resultado general del estudio de las 80 instancias, se observa que atendiendo solamente al número de lotes, el algoritmo de Distancias consigue el 89% de las mejores soluciones, mientras que si se centra la atención solamente en el número de elementos sin asignar el algoritmo Greedy consigue la mejor solución en un 86% de las instancias. En la Figura 57 se puede ver la distribución de mejores soluciones atendiendo solamente a estos dos factores.

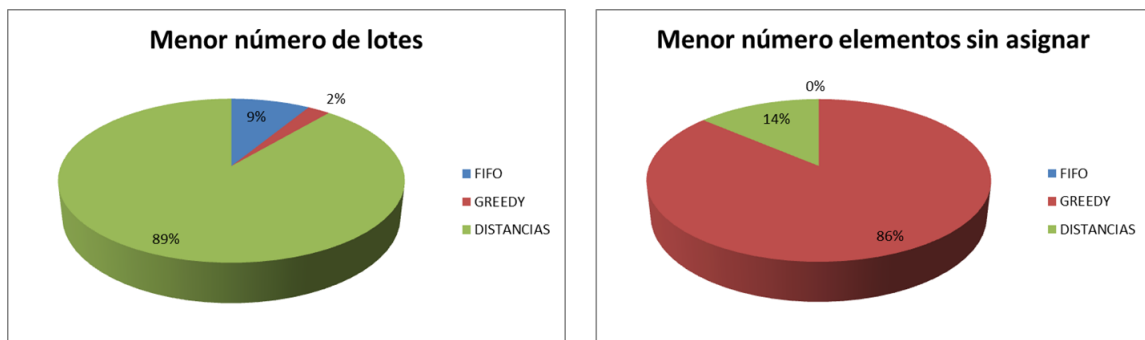


Figura 57. Heurísticos con mejores números de lotes y menor número de elementos sin asignar

Representando conjuntamente el número de lotes junto con el número de elementos sin asignar la distribución de mejores soluciones de las heurísticas es la siguiente:

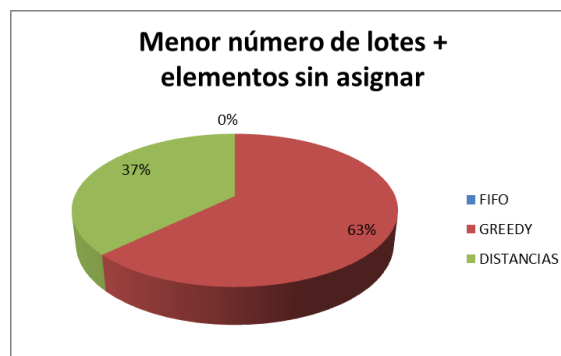


Figura 58. Heurísticos con mejor número de lotes + número de elementos sin asignar

El algoritmo de distancias es el que consigue generar menos número de lotes, pero a cambio de dejar más elementos sueltos. Con el algoritmo Greedy se da la situación inversa: consigue agrupar más elementos, lo que conlleva mayor número de lotes. Atendiendo a ambos parámetros en conjunto el algoritmo Greedy consigue el mejor comportamiento en un 63% de los casos, mientras que el algoritmo de Distancias lo hacen en el 37% restante.

Como se había visto en el apartado anterior, al analizar la función de coste el algoritmo Greedy conseguía mejores soluciones en 50 de las 80 instancias, es decir, un 62.5%. Estos valores se corresponden con el 63% de mejores soluciones atendiendo solamente al número de lotes y número de productos individuales, con lo que parece que la calidad de las soluciones en cuanto a la homogeneidad de los lotes debe de seguir una tendencia parecida. Para

constatarlo, se analizan a continuación las soluciones atendiendo a la homogeneidad de los lotes.

8.2.1.3 Homogeneidad de los lotes

Atendiendo solamente a la parte de la función de coste que tiene en cuenta la homogeneidad de los lotes generados, la distribución de mejores soluciones en los heurísticos se puede ver en la Figura 59:

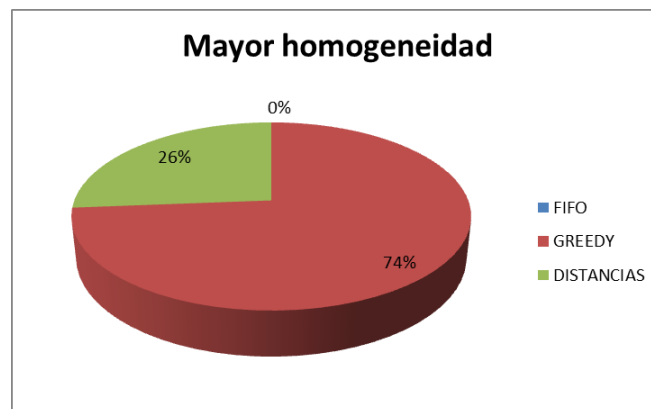


Figura 59. Heurísticos con mejor homogeneidad en las instancias de prueba

En línea con lo comentado en el apartado anterior, las mejores soluciones atendiendo solamente a la homogeneidad se consiguen con el algoritmo Greedy.

En la Tabla 19 se puede ver una comparativa de todas las instancias, atendiendo a la mejor solución global (función de coste), al menor número de lotes y elementos sueltos, y a la homogeneidad de los lotes:

Instancia	#lotes #no asig.	Homogeneo	Coste total	Instancia	#lotes #no asig.	Homogeneo	Coste total
120_00.csv	DISTANCIAS	GREEDY	GREEDY	250_00.csv	GREEDY	GREEDY	GREEDY
120_01.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_01.csv	DISTANCIAS	GREEDY	GREEDY
120_02.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_02.csv	GREEDY	GREEDY	GREEDY
120_03.csv	GREEDY	GREEDY	GREEDY	250_03.csv	DISTANCIAS	GREEDY	GREEDY
120_04.csv	GREEDY	GREEDY	GREEDY	250_04.csv	DISTANCIAS	GREEDY	GREEDY
120_05.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_05.csv	DISTANCIAS	GREEDY	GREEDY
120_06.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_06.csv	DISTANCIAS	GREEDY	GREEDY
120_07.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_07.csv	GREEDY	GREEDY	GREEDY
120_08.csv	GREEDY	GREEDY	GREEDY	250_08.csv	GREEDY	GREEDY	GREEDY
120_09.csv	DISTANCIAS	GREEDY	GREEDY	250_09.csv	GREEDY	GREEDY	GREEDY
120_10.csv	GREEDY	GREEDY	GREEDY	250_10.csv	GREEDY	GREEDY	GREEDY
120_11.csv	DISTANCIAS	GREEDY	GREEDY	250_11.csv	GREEDY	GREEDY	GREEDY
120_12.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_12.csv	GREEDY	GREEDY	GREEDY
120_13.csv	GREEDY	GREEDY	GREEDY	250_13.csv	GREEDY	GREEDY	GREEDY
120_14.csv	DISTANCIAS	GREEDY	GREEDY	250_14.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS
120_15.csv	DISTANCIAS	GREEDY	DISTANCIAS	250_15.csv	GREEDY	GREEDY	GREEDY
120_16.csv	GREEDY	GREEDY	GREEDY	250_16.csv	GREEDY	GREEDY	GREEDY
120_17.csv	GREEDY	GREEDY	GREEDY	250_17.csv	GREEDY	GREEDY	GREEDY
120_18.csv	DISTANCIAS	GREEDY	GREEDY	250_18.csv	GREEDY	GREEDY	GREEDY
120_19.csv	DISTANCIAS	GREEDY	GREEDY	250_19.csv	GREEDY	GREEDY	GREEDY
500_00.csv	GREEDY	GREEDY	GREEDY	1000_00.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_01.csv	DISTANCIAS	GREEDY	GREEDY	1000_01.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_02.csv	GREEDY	DISTANCIAS	DISTANCIAS	1000_02.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_03.csv	DISTANCIAS	GREEDY	GREEDY	1000_03.csv	GREEDY	GREEDY	GREEDY
500_04.csv	GREEDY	GREEDY	GREEDY	1000_04.csv	DISTANCIAS	GREEDY	GREEDY
500_05.csv	GREEDY	GREEDY	GREEDY	1000_05.csv	GREEDY	GREEDY	GREEDY
500_06.csv	GREEDY	GREEDY	GREEDY	1000_06.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS
500_07.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS	1000_07.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_08.csv	DISTANCIAS	GREEDY	DISTANCIAS	1000_08.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_09.csv	GREEDY	GREEDY	GREEDY	1000_09.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_10.csv	GREEDY	DISTANCIAS	DISTANCIAS	1000_10.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_11.csv	DISTANCIAS	GREEDY	GREEDY	1000_11.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS
500_12.csv	GREEDY	DISTANCIAS	DISTANCIAS	1000_12.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS
500_13.csv	DISTANCIAS	GREEDY	GREEDY	1000_13.csv	GREEDY	GREEDY	GREEDY
500_14.csv	GREEDY	GREEDY	GREEDY	1000_14.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_15.csv	GREEDY	GREEDY	GREEDY	1000_15.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_16.csv	GREEDY	GREEDY	GREEDY	1000_16.csv	GREEDY	GREEDY	GREEDY
500_17.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS	1000_17.csv	GREEDY	DISTANCIAS	DISTANCIAS
500_18.csv	DISTANCIAS	GREEDY	DISTANCIAS	1000_18.csv	DISTANCIAS	DISTANCIAS	DISTANCIAS
500_19.csv	GREEDY	GREEDY	GREEDY	1000_19.csv	GREEDY	DISTANCIAS	DISTANCIAS

Tabla 19. Comparativa de mejores soluciones frente a homogeneidad y número de lotes + elementos sueltos

Se puede observar que el factor homogeneidad influye decisivamente en el valor final de la función de coste en 20 de las 80 instancias, lo cual es un valor muy coherente, ya que se deseaba que el factor homogeneidad fuera utilizado para mejorar las soluciones, pero sin prevalecer frente al número de lotes y elementos sueltos.

8.2.1.4 Tiempo computacional

Es necesario analizar el tiempo empleado en el algoritmo para cada una de las heurísticas. Para comparar los tiempos se utilizará el tiempo medio para cada una de las instancias de 120, 250, 500 y 1000 productos. En la Tabla 20 se representa el resultado de las ejecuciones. El tiempo del algoritmo se mide en segundos.

	120 productos	250 productos	500 productos	1000 productos
Fifo	1.15	1.16	3.65	13.34
Greedy	1.32	2.21	6.07	20.06
Distancias	1.92	4.07	10.14	30.45

Tabla 20. Tiempos medios de ejecución de las instancias de prueba con cada heurística

Para todas las instancias, el algoritmo Fifo es el más rápido, seguido del Greedy, y por último del algoritmo basado en Distancias. La evolución del tiempo de ejecución para cada instancia se representa a continuación:

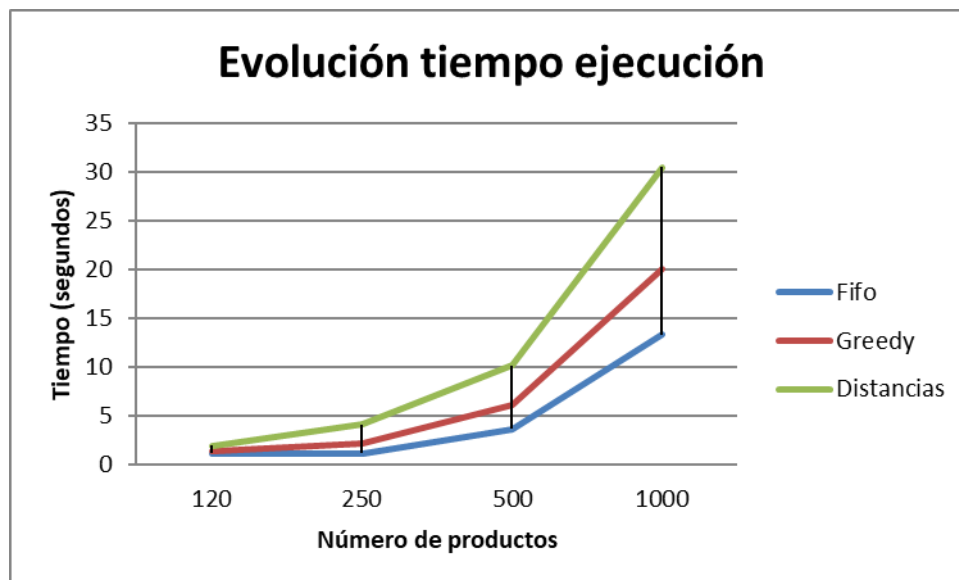


Figura 60. Evolución del tiempo de ejecución para cada heurística

8.2.1.5 Resumen de las pruebas con los heurísticos

Del análisis de los experimentos realizados con las instancias de prueba generadas se puede concluir que el algoritmo Greedy es el que presenta mejores resultados. Consigue la menor función de coste en 50 de las 80 instancias, destacando claramente tanto en la suma de número de lotes más número de elementos individuales, como por otro lado en la homogeneidad de los mismos, donde presenta mejores resultados para el 74% de las instancias.

El algoritmo basado en Distancias consigue las mejores soluciones en el resto de las 30 instancias. Este algoritmo destaca sobre todo por la generación de pocos lotes, aunque a veces a costa de dejar más elementos sueltos. Por su parte, el algoritmo FIFO no consigue ninguna mejor solución con ninguna de las instancias, debido al gran número de elementos que deja sin agrupar.

Analizando los tiempos de computación, se observa claramente el comportamiento exponencial de los mismos. Para todas las instancias, el menor tiempo lo emplea el Fifo, seguido del Greedy y finalmente del algoritmo de Distancias.

8.2.2 Comportamiento del GRASP

Para analizar la bondad del algoritmo GRASP generado se repetirán todos los experimentos con las 80 instancias. Como algoritmo base para encontrar una solución inicial se utilizarán de nuevo los heurísticos FIFO, Greedy y basado en Distancias.

El número de veces que se realiza el intento de mejora de la solución se ha determinado de tal manera que los experimentos sean comparables. Para ello, se ha tomado como referencia el tiempo de ejecución de los heurísticos, mostrado en la Tabla 20, y se ha elegido un número de repeticiones de forma que el tiempo con el GRASP sea similar. Con este criterio, el número de veces que se realizará la mejora en el GRASP será de 55 para los algoritmos Greedy y Distancias, mientras que sólo de 40 en el FIFO. Por su parte, el número de iteraciones de las heurísticas será de 100.

Primeramente, se analizará la mejor solución para cada instancia obtenida con el GRASP, frente a la mejor solución de los heurísticos, para poder analizar la capacidad de

mejora y el rendimiento del GRASP desarrollado. Después se analizará el comportamiento del GRASP junto con cada heurístico utilizado para encontrar la solución inicial, teniendo en cuenta su función de coste para determinar con qué heurístico se han encontrado las mejores soluciones, así como el tiempo computacional de los mismos.

8.2.2.1 Rendimiento del GRASP

Para poder comprobar la capacidad de mejora del algoritmo de búsqueda local desarrollado, se compararán las soluciones comentadas en el apartado anterior, obtenidas con cada heurístico, con las soluciones obtenidas utilizando como base cada heurístico para encontrar la solución inicial en cada iteración del GRASP.

Los resultados obtenidos son los siguientes:

Instancias de 120 productos:

	fifo			greedy			Distancias		
	sin grasp	grasp	mejora	sin grasp	grasp	mejora	sin grasp	grasp	mejora
120_00	66.75	22.85	SI	62.58	29.00	SI	64.59	30.15	SI
120_01	66.36	24.94	SI	63.35	33.09	SI	62.50	32.38	SI
120_02	64.08	23.72	SI	59.99	30.54	SI	59.45	32.31	SI
120_03	67.10	19.34	SI	61.93	27.01	SI	64.36	27.82	SI
120_04	69.76	21.91	SI	65.97	26.70	SI	66.95	29.38	SI
120_05	74.18	23.67	SI	70.92	28.90	SI	67.37	30.46	SI
120_06	167.63	233.10	NO	165.12	240.17	NO	164.35	241.30	NO
120_07	67.99	24.80	SI	64.77	31.63	SI	64.58	33.30	SI
120_08	67.07	26.49	SI	62.60	34.40	SI	64.40	33.97	SI
120_09	63.22	19.81	SI	59.08	28.17	SI	59.53	28.13	SI
120_10	72.56	29.96	SI	68.52	37.39	SI	70.40	33.43	SI
120_11	65.44	127.59	NO	62.02	134.10	NO	62.45	134.57	NO
120_12	65.35	24.33	SI	60.77	33.29	SI	60.33	32.18	SI
120_13	64.33	21.87	SI	59.77	30.08	SI	62.36	29.24	SI
120_14	67.54	126.24	NO	63.13	133.66	NO	63.85	138.16	NO
120_15	70.56	25.08	SI	66.21	32.79	SI	65.85	31.55	SI
120_16	71.13	127.52	NO	67.59	134.29	NO	69.77	134.70	NO
120_17	69.66	26.56	SI	64.74	34.53	SI	66.31	30.22	SI
120_18	68.47	27.13	SI	65.18	34.77	SI	65.83	36.25	SI
120_19	172.09	127.59	SI	168.27	135.38	SI	168.49	135.00	SI

Tabla 21. Comparativa de soluciones con los heurísticos frente al GRASP para 120 productos

Instancias de 250 productos:

	fifo			greedy			Distancias		
	sin grasp	grasp	mejora	sin grasp	grasp	mejora	sin grasp	grasp	mejora
250_00	232.57	265.12	NO	222.34	284.25	NO	225.79	174.96	SI
250_01	141.07	52.16	SI	132.54	69.74	SI	134.24	64.72	SI
250_02	339.47	259.37	SI	328.59	278.07	SI	331.51	266.87	SI
250_03	237.60	161.43	SI	228.71	176.58	SI	232.06	171.18	SI
250_04	137.75	53.03	SI	130.78	70.74	SI	131.09	71.82	SI
250_05	235.98	153.68	SI	224.77	175.55	SI	227.19	170.61	SI
250_06	134.52	155.46	NO	123.00	176.47	NO	124.33	174.58	NO
250_07	134.44	51.98	SI	121.29	73.93	SI	126.54	65.29	SI
250_08	238.07	159.80	SI	223.66	182.89	SI	232.09	175.25	SI
250_09	135.14	49.99	SI	120.35	74.09	SI	130.80	59.92	SI
250_10	137.25	53.78	SI	123.45	76.11	SI	132.05	65.12	SI
250_11	134.20	50.01	SI	119.85	70.66	SI	127.55	61.05	SI
250_12	143.94	56.63	SI	129.95	76.51	SI	136.23	68.06	SI
250_13	234.24	165.71	SI	221.12	188.38	SI	227.46	171.31	SI
250_14	232.80	260.17	NO	220.71	282.21	NO	119.89	166.40	NO
250_15	140.42	51.33	SI	126.50	73.09	SI	133.23	65.43	SI
250_16	128.76	56.53	SI	116.62	75.24	SI	121.78	64.99	SI
250_17	130.72	55.78	SI	119.69	75.08	SI	122.36	70.26	SI
250_18	131.72	52.47	SI	119.00	76.89	SI	124.00	68.44	SI
250_19	135.86	58.37	SI	124.92	74.32	SI	128.42	69.69	SI

Tabla 22. Comparativa de soluciones con los heurísticos frente al GRASP para 250 productos

Instancias de 500 productos:

	fifo			greedy			Distancias		
	sin grasp	grasp	mejora	sin grasp	grasp	mejora	sin grasp	grasp	mejora
500_00	371.21	320.62	SI	345.09	359.48	NO	352.74	341.23	SI
500_01	678.03	640.91	SI	657.38	676.01	NO	659.63	562.77	SI
500_02	578.49	103.81	SI	553.95	147.20	SI	457.64	128.33	SI
500_03	480.80	530.92	NO	453.92	572.46	NO	456.43	562.83	NO
500_04	587.55	206.17	SI	567.04	248.85	SI	569.80	236.84	SI
500_05	382.26	225.85	SI	354.74	264.66	SI	363.98	238.87	SI
500_06	580.10	428.08	SI	554.49	473.78	SI	564.00	343.91	SI
500_07	790.63	619.79	SI	768.52	654.52	SI	466.53	336.96	SI
500_08	368.80	342.83	SI	349.04	369.01	NO	346.97	347.01	NO
500_09	472.58	538.76	NO	446.49	577.27	NO	450.88	564.19	NO
500_10	676.39	536.94	SI	652.97	575.67	SI	557.47	453.54	SI
500_11	674.65	530.10	SI	654.77	567.81	SI	656.89	556.63	SI
500_12	371.15	208.75	SI	344.47	254.54	SI	250.32	237.07	SI
500_13	467.75	632.15	NO	443.53	667.50	NO	446.08	664.83	NO
500_14	376.38	421.77	NO	350.69	464.11	NO	358.02	237.55	SI
500_15	371.09	110.60	SI	342.78	146.62	SI	352.58	140.40	SI
500_16	472.02	426.02	SI	450.36	465.21	NO	458.29	451.16	SI
500_17	673.81	423.17	SI	654.70	454.84	SI	551.17	449.00	SI
500_18	892.98	642.29	SI	875.27	678.63	SI	873.19	668.56	SI
500_19	467.26	212.92	SI	441.01	254.98	SI	452.18	230.74	SI

Tabla 23. Comparativa de soluciones con los heurísticos frente al GRASP para 500 productos

Instancias de 1000 productos:

	fifo			greedy			Distancias		
	sin grasp	grasp	mejora	sin grasp	grasp	mejora	sin grasp	grasp	mejora
1000_00	1262.28	869.08	SI	1213.43	1254.54	NO	1123.97	792.07	SI
1000_01	1268.40	1168.03	SI	1217.99	846.60	SI	1124.37	1108.11	SI
1000_02	1166.12	965.71	SI	1112.38	1248.75	NO	1036.23	990.98	SI
1000_03	858.57	635.26	SI	796.97	720.41	SI	819.89	679.45	SI
1000_04	945.11	764.07	SI	894.19	1060.04	NO	899.18	692.35	SI
1000_05	944.50	671.65	SI	891.81	951.15	NO	905.46	607.56	SI
1000_06	1055.24	1093.00	NO	1002.74	1139.62	NO	802.45	818.23	NO
1000_07	1051.62	760.68	SI	995.79	1367.24	NO	915.25	793.76	SI
1000_08	956.44	942.68	SI	908.37	960.64	NO	818.54	877.28	NO
1000_09	740.12	1288.58	NO	684.07	621.87	SI	603.46	1112.33	NO
1000_10	1051.91	875.61	SI	1003.27	762.34	SI	919.71	799.07	SI
1000_11	1053.58	969.16	SI	1002.96	1164.98	NO	909.94	801.19	SI
1000_12	1250.29	1061.16	SI	1197.85	839.76	SI	797.69	890.67	NO
1000_13	939.41	767.76	SI	880.74	833.40	SI	901.22	586.38	SI
1000_14	931.75	655.12	SI	881.68	1018.89	NO	795.76	379.27	SI
1000_15	1052.29	856.60	SI	999.42	940.54	SI	915.76	674.46	SI
1000_16	946.19	751.79	SI	892.28	940.06	NO	907.88	790.43	SI
1000_17	955.86	866.80	SI	898.24	846.04	SI	819.27	787.29	SI
1000_18	1863.33	1180.38	SI	1811.75	750.41	SI	1312.46	1007.19	SI
1000_19	956.43	545.24	SI	905.72	1053.87	NO	818.87	470.45	SI

Tabla 24. Comparativa de soluciones con los heurísticos frente al GRASP para 1000 productos

De las tablas anteriores se puede observar que el algoritmo GRASP mejora a los heurísticos en 187 de las 240 soluciones, es decir, mejora en el 78% de los casos, para el mismo tiempo computacional. Además, si se comparan las mejores soluciones obtenidas para cada instancia con el heurístico FIFO, Greedy o de Distancias frente a las mejores soluciones obtenidas con el algoritmo GRASP se puede observar que el algoritmo GRASP mejora en 67 de las 80 instancias, es decir, en un 84% de los casos. De aquí se puede concluir claramente que el algoritmo GRASP desarrollado es más adecuado para encontrar mejores soluciones que los heurísticos individuales. En la Tabla 25 se pueden observar las mejores soluciones con cualquier heurístico frente a las mejores soluciones con el GRASP.

instancia	sin grasp	grasp	mejora	instancia	sin grasp	grasp	mejora
120_00.csv	62.58	22.85	SI	500_00.csv	345.09	320.62	SI
120_01.csv	62.50	24.94	SI	500_01.csv	657.38	562.77	SI
120_02.csv	59.45	23.72	SI	500_02.csv	457.64	103.81	SI
120_03.csv	61.93	19.34	SI	500_03.csv	453.92	530.92	NO
120_04.csv	65.97	21.91	SI	500_04.csv	567.04	206.17	SI
120_05.csv	67.37	23.67	SI	500_05.csv	354.74	225.85	SI
120_06.csv	164.35	233.10	NO	500_06.csv	554.49	343.91	SI
120_07.csv	64.58	24.80	SI	500_07.csv	466.53	336.96	SI
120_08.csv	62.60	26.49	SI	500_08.csv	346.97	342.83	SI
120_09.csv	59.08	19.81	SI	500_09.csv	446.49	538.76	NO
120_10.csv	68.52	29.96	SI	500_10.csv	557.47	453.54	SI
120_11.csv	62.02	127.59	NO	500_11.csv	654.77	530.10	SI
120_12.csv	60.33	24.33	SI	500_12.csv	250.32	208.75	SI
120_13.csv	59.77	21.87	SI	500_13.csv	443.53	632.15	NO
120_14.csv	63.13	126.24	NO	500_14.csv	350.69	237.55	SI
120_15.csv	65.85	25.08	SI	500_15.csv	342.78	110.60	SI
120_16.csv	67.59	127.52	NO	500_16.csv	450.36	426.02	SI
120_17.csv	64.74	26.56	SI	500_17.csv	551.17	423.17	SI
120_18.csv	65.18	27.13	SI	500_18.csv	873.19	642.29	SI
120_19.csv	168.27	127.59	SI	500_19.csv	441.01	212.92	SI
250_00.csv	222.34	174.96	SI	1000_00.csv	1123.97	792.07	SI
250_01.csv	132.54	52.16	SI	1000_01.csv	1124.37	846.60	SI
250_02.csv	328.59	259.37	SI	1000_02.csv	1036.23	965.71	SI
250_03.csv	228.71	161.43	SI	1000_03.csv	796.97	635.26	SI
250_04.csv	130.78	53.03	SI	1000_04.csv	894.19	692.35	SI
250_05.csv	224.77	153.68	SI	1000_05.csv	891.81	607.56	SI
250_06.csv	123.00	155.46	NO	1000_06.csv	802.45	818.23	NO
250_07.csv	121.29	51.98	SI	1000_07.csv	915.25	760.68	SI
250_08.csv	223.66	159.80	SI	1000_08.csv	818.54	877.28	NO
250_09.csv	120.35	49.99	SI	1000_09.csv	603.46	621.87	NO
250_10.csv	123.45	53.78	SI	1000_10.csv	919.71	762.34	SI
250_11.csv	119.85	50.01	SI	1000_11.csv	909.94	801.19	SI
250_12.csv	129.95	56.63	SI	1000_12.csv	797.69	839.76	NO
250_13.csv	221.12	165.71	SI	1000_13.csv	880.74	586.38	SI
250_14.csv	119.89	166.40	NO	1000_14.csv	795.76	379.27	SI
250_15.csv	126.50	51.33	SI	1000_15.csv	915.76	674.46	SI
250_16.csv	116.62	56.53	SI	1000_16.csv	892.28	751.79	SI
250_17.csv	119.69	55.78	SI	1000_17.csv	819.27	787.29	SI
250_18.csv	119.00	52.47	SI	1000_18.csv	1312.46	750.41	SI
250_19.csv	124.92	58.37	SI	1000_19.csv	818.87	470.45	SI

Tabla 25. Mejor solución obtenida con el GRASP frente a cualquier heurístico para cada instancia

8.2.2.2 Función de coste

La característica que define la bondad de la solución es el valor de la función de coste. En las siguientes tablas se observa una comparativa del valor obtenido con el algoritmo GRASP, utilizando los diferentes heurísticos para generar la solución inicial, para cada una de las instancias de prueba.

120 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
120_00.csv	22.85	29.00	30.15	FIFO
120_01.csv	24.94	33.09	32.38	FIFO
120_02.csv	23.72	30.54	32.31	FIFO
120_03.csv	19.34	27.01	27.82	FIFO
120_04.csv	21.91	26.70	29.38	FIFO
120_05.csv	23.67	28.90	30.46	FIFO
120_06.csv	233.10	240.17	241.30	FIFO
120_07.csv	24.80	31.63	33.30	FIFO
120_08.csv	26.49	34.40	33.97	FIFO
120_09.csv	19.81	28.17	28.13	FIFO
120_10.csv	29.96	37.39	33.43	FIFO
120_11.csv	127.59	134.10	134.57	FIFO
120_12.csv	24.33	33.29	32.18	FIFO
120_13.csv	21.87	30.08	29.24	FIFO
120_14.csv	126.24	133.66	138.16	FIFO
120_15.csv	25.08	32.79	31.55	FIFO
120_16.csv	127.52	134.29	134.70	FIFO
120_17.csv	26.56	34.53	30.22	FIFO
120_18.csv	27.13	34.77	36.25	FIFO
120_19.csv	127.59	135.38	135.00	FIFO

Tabla 26. Valores de la función de coste del GRASP con cada heurístico con las instancias de 120 productos

250 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
250_00.csv	265.12	284.25	174.96	DISTANCIAS
250_01.csv	52.16	69.74	64.72	FIFO
250_02.csv	259.37	278.07	266.87	FIFO
250_03.csv	161.43	176.58	171.18	FIFO
250_04.csv	53.03	70.74	71.82	FIFO
250_05.csv	153.68	175.55	170.61	FIFO
250_06.csv	155.46	176.47	174.58	FIFO
250_07.csv	51.98	73.93	65.29	FIFO
250_08.csv	159.80	182.89	175.25	FIFO
250_09.csv	49.99	74.09	59.92	FIFO
250_10.csv	53.78	76.11	65.12	FIFO
250_11.csv	50.01	70.66	61.05	FIFO
250_12.csv	56.63	76.51	68.06	FIFO
250_13.csv	165.71	188.38	171.31	FIFO
250_14.csv	260.17	282.21	166.40	DISTANCIAS
250_15.csv	51.33	73.09	65.43	FIFO
250_16.csv	56.53	75.24	64.99	FIFO
250_17.csv	55.78	75.08	70.26	FIFO
250_18.csv	52.47	76.89	68.44	FIFO
250_19.csv	58.37	74.32	69.69	FIFO

Tabla 27. Valores de la función de coste del GRASP con cada heurístico con las instancias de 250 productos

500 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
500_00.csv	320.62	359.48	341.23	FIFO
500_01.csv	640.91	676.01	562.77	DISTANCIAS
500_02.csv	103.81	147.20	128.33	FIFO
500_03.csv	530.92	572.46	562.83	FIFO
500_04.csv	206.17	248.85	236.84	FIFO
500_05.csv	225.85	264.66	238.87	FIFO
500_06.csv	428.08	473.78	343.91	DISTANCIAS
500_07.csv	619.79	654.52	336.96	DISTANCIAS
500_08.csv	342.83	369.01	347.01	FIFO
500_09.csv	538.76	577.27	564.19	FIFO
500_10.csv	536.94	575.67	453.54	DISTANCIAS
500_11.csv	530.10	567.81	556.63	FIFO
500_12.csv	208.75	254.54	237.07	FIFO
500_13.csv	632.15	667.50	664.83	FIFO
500_14.csv	421.77	464.11	237.55	DISTANCIAS
500_15.csv	110.60	146.62	140.40	FIFO
500_16.csv	426.02	465.21	451.16	FIFO
500_17.csv	423.17	454.84	449.00	FIFO
500_18.csv	642.29	678.63	668.56	FIFO
500_19.csv	212.92	254.98	230.74	FIFO

Tabla 28. Valores de la función de coste del GRASP con cada heurístico con las instancias de 500 productos

1000 productos:

	FIFO	GREEDY	DISTANCIAS	Mejor solución
1000_00.csv	869.08	1254.54	792.07	DISTANCIAS
1000_01.csv	1168.03	846.60	1108.11	GREEDY
1000_02.csv	965.71	1248.75	990.98	FIFO
1000_03.csv	635.26	720.41	679.45	FIFO
1000_04.csv	764.07	1060.04	692.35	DISTANCIAS
1000_05.csv	671.65	951.15	607.56	DISTANCIAS
1000_06.csv	1093.00	1139.62	818.23	DISTANCIAS
1000_07.csv	760.68	1367.24	793.76	FIFO
1000_08.csv	942.68	960.64	877.28	DISTANCIAS
1000_09.csv	1288.58	621.87	1112.33	GREEDY
1000_10.csv	875.61	762.34	799.07	GREEDY
1000_11.csv	969.16	1164.98	801.19	DISTANCIAS
1000_12.csv	1061.16	839.76	890.67	GREEDY
1000_13.csv	767.76	833.40	586.38	DISTANCIAS
1000_14.csv	655.12	1018.89	379.27	DISTANCIAS
1000_15.csv	856.60	940.54	674.46	DISTANCIAS
1000_16.csv	751.79	940.06	790.43	FIFO
1000_17.csv	866.80	846.04	787.29	DISTANCIAS
1000_18.csv	1180.38	750.41	1007.19	GREEDY
1000_19.csv	545.24	1053.87	470.45	DISTANCIAS

Tabla 29. Valores de la función de coste del GRASP con cada heurístico con las instancias de 1000 productos

De las 80 instancias de prueba, en 67 la mejor solución se consigue utilizando el heurístico FIFO para generar la solución inicial del GRASP. En 5 ocasiones se consigue con el Greedy, y en las 18 restantes con el algoritmo basado en Distancias. En la Figura 61 se puede ver la distribución de mejores soluciones del GRASP con cada heurístico para las 80 instancias.

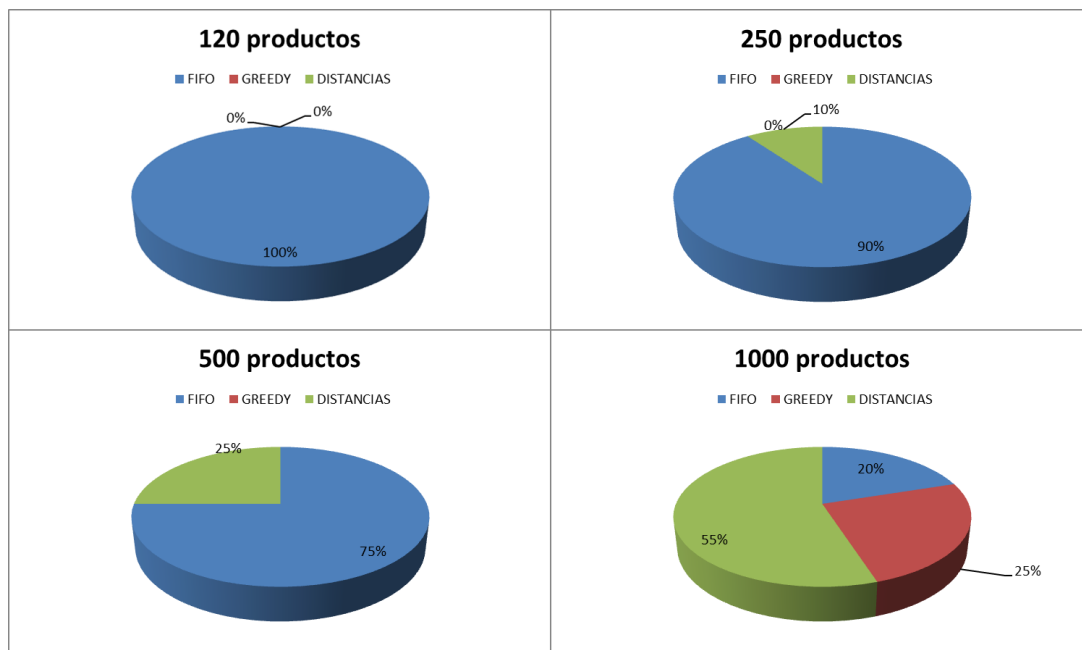


Figura 61. Distribución de mejores soluciones con el GRASP y cada heurístico

Es interesante destacar el hecho de que utilizando solamente los heurísticos, las mejores soluciones se consiguen con el heurístico Greedy y el de Distancias. Sin embargo, cuando estos heurísticos se utilizan como generadores de soluciones iniciales en el GRASP este comportamiento no es así, ya que el heurístico que mejor funciona como generador de soluciones al GRASP es el FIFO. El autor de esta tesis supone que esto se debe a que las soluciones obtenidas con los heurísticos Greedy y de Distancias alcanzaban unos valores mínimos locales, pero llegan a un punto en el que son incapaces de ser mejoradas debido a las restricciones del problema. Sin embargo, al utilizar el algoritmo GRASP, el partir de una peor solución a priori como el fifo permite tener más flexibilidad a la búsqueda local para mejorar la solución, explorando otras zonas del espacio de soluciones que no se podían encontrar con los heurísticos simples. El algoritmo Greedy y el basado en Distancias generan lotes muy homogéneos rápidamente, que son menos susceptibles de poder intercambiar elementos con otros lotes. Este hecho, unido a la bondad de mejora de la búsqueda local explican que las mejores soluciones se consigan con el GRASP utilizando el FIFO como base para obtener la solución inicial.

Para corroborar este hecho, se han repetido las pruebas aumentando mucho el tiempo computacional; para ello, se han utilizado un número de 500 iteraciones en el heurístico, y 75 veces la parte de búsqueda local (55 en el caso del FIFO como solución inicial), representando un tiempo computacional aproximado de unas 20 veces más. Tras la realización del test se ha podido comprobar que se repite la tendencia explicada en el párrafo anterior, siendo el GRASP con FIFO como solución inicial el mejor en 57 de las 80 instancias, y el GRASP con Distancias en las 23 instancias restantes. Se comprueba por lo tanto la calidad del algoritmo GRASP implementado, y su mejor eficiencia partiendo como base del heurístico FIFO, seguido del heurístico basado en Distancias.

8.2.2.3 Tiempo computacional

El tiempo medio empleado por el algoritmo GRASP, partiendo de cada heurístico para generar la solución inicial para cada uno de los tamaños de productos se muestra en la Tabla 30.

	120 productos	250 productos	500 productos	1000 productos
GRASP (Fifo)	1.22	1.5	4.5	13.45
GRASP (Greedy)	1.54	2.51	6.18	17.45
GRASP (Distancias)	2.03	4.4	10.06	30.5

Tabla 30. Tiempos medios de ejecución de las instancias de prueba

Representando gráficamente estos valores, se observa que igualmente que en el caso de los heurísticos, el tiempo de ejecución aumenta de manera exponencial. Se puede concluir que el comportamiento del GRASP presenta una evolución similar al de los heurísticos individuales.

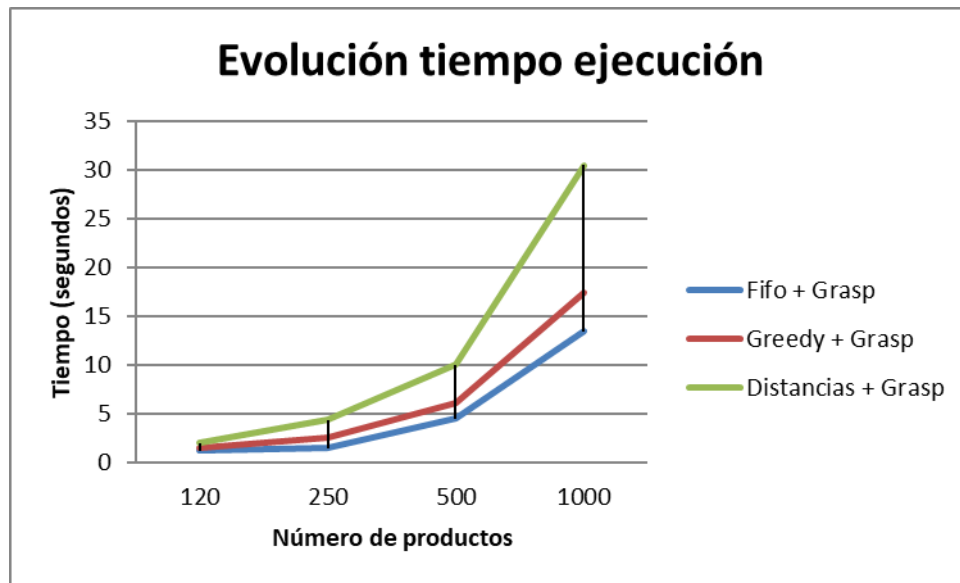


Figura 62. Evolución del tiempo de ejecución

8.2.2.4 Resumen de las pruebas con el GRASP

El algoritmo GRASP parte de una solución inicial y lo intenta mejorar mediante una búsqueda local, en este caso desarrollada en 2 fases: mejora de los lotes generados y reasignación final de los elementos sin asignar. Para generar la solución inicial se han realizado pruebas utilizando cada uno de los heurísticos desarrollados.

La mejora del GRASP desarrollado con respecto a los heurísticos iniciales es significativa, encontrando mejores soluciones en un 84% de las instancias. El motivo de esta mejora es que la búsqueda local desarrollada permite explorar otras zonas del espacio de soluciones, que mediante los heurísticos individuales no se podían analizar.

Las pruebas realizadas sobre todas las instancias generadas demuestran que el heurístico FIFO es el mejor para la generación de soluciones iniciales, precisamente debido a que estas soluciones iniciales no se encuentran tan limitadas por la capacidad de los lotes o la distancia entre los productos, con lo cual la búsqueda local del GRASP los mejora enormemente.

8.3 Experimento realizado con un caso de prueba real

Una vez realizada la comparación de las soluciones obtenidas con las diferentes algorítmicas propuestas, en este apartado se mostrará la solución propuesta por el algoritmo sobre un caso de uso real. Para ello, se han recogido los datos de 1 semana de 2017 que los expertos consideran como representativa de la realidad para el análisis.

8.3.1 Datos de entrada

El fichero de entrada del algoritmo tiene las siguientes características:

- Número de productos: 2771
- Plantas de origen de los productos: 26
- Familias: 12
- Subfamilias: 16
- Grados de acero distintos: 76

8.3.2 Configuración del algoritmo

Para la parte inicial de cálculo de distancias entre cada par de productos, el valor de los diferentes pesos se ha configurado a “1”. De esta manera se está dando la misma importancia a todas las características de las bobinas. Sin embargo, en el uso diario, los diferentes pesos deben de ser configurados por los expertos, asignando diferentes pesos a cada característica, en función de la familia.

Para la parte del clustering se han utilizado los siguientes parámetros:

- Threshold = 1.5
- MinSize = 20
- MaxSize = 100
- Step_refine = 0.8

Para la parte final de asignación, tras las conclusiones obtenidas con las instancias de prueba generadas, se ha decidido utilizar el meta-heurístico GRASP, y lanzarlo 2 veces: la primera vez utilizando como base para generar la solución el heurístico FIFO, y la segunda el

heurístico basado en Distancias. La combinación de GRASP tomando como base el heurístico Greedy no consiguió nunca la mejor solución sobre las instancias de prueba, y su inclusión aumentaría el tiempo computacional. Lanzando el GRASP tanto con el FIFO con el heurístico basado en Distancias se garantiza que la solución obtenida será de alta calidad, y el tiempo computacional encaja dentro de lo estipulado, como se mostrará más adelante. El número de iteraciones de los heurísticos es de 100, mientras que el número de veces que se realiza la búsqueda local es de 75. Además, se han utilizado los siguientes parámetros:

- Tamaño mínimo de lote: 15 toneladas
- Tamaño máximo de lote: 25 toneladas

Para los resultados que se muestran a continuación, el experimento se ha realizado 10 veces, para que los resultados sean estadísticamente significativos. Los resultados son la media de los 10 experimentos realizados.

8.3.3 Solución

En primer lugar, el algoritmo aplica la separación inicial por parámetros globales (planta, localización, forma, categoría, familia y número de lados de recubrimiento).

A continuación se lanza el algoritmo de clustering, resultando a la salida del mismo 208 clústeres de elementos que ya son cercanos entre sí, conteniendo el mayor de los clústeres 48 elementos. La distribución de los elementos en clústeres se puede ver en la Tabla 31. Se puede observar que existen 336 elementos que son ya lo suficientemente distintos al resto como para no poder agruparse; estos elementos serán bobinas sin asignar a ningún lote al final del algoritmo.

Número elementos	Número clústeres
1	336
2	41
3	15
4	13
5	12
6	7
8	11
9	8
10	7
11	8
12	5
13	3
14	4
15	2
16	4
17	1
19	2
20	6
21	8
22	9
23	2
24	6
25	4
26	3
27	3
28	4
30	1
31	2
33	2
35	1
38	1
39	1
40	2
42	1
45	1
48	1

Tabla 31. Distribución de elementos en clústeres

Cada uno de esos 208 clústeres pasará a la entrada de los algoritmos finales de asignación de productos a lotes, cuyo resultado final es:

- Número de lotes generados: 291
- Número de elementos individuales: 601

De los 2435 elementos que estaban agrupados en clústeres con tamaño mayor que 1, al final del algoritmo solamente 265 elementos han quedado sin asignar a un lote, que junto con los 336 que ya habían sido dejados como elementos individuales al final de la fase de clustering hacen los 601 elementos finales que quedan sin asignar a lotes.

Tras discutir los resultados con los expertos, confirman que son muy buenos tanto desde el punto de vista de lotes generados y de elementos sin asignar, como desde el punto de vista de la homogeneidad, garantizada con la distancia máxima asignada en cada lote.

Otro punto a destacar es la generalidad del algoritmo, ya que al hacer posible asignar distintos pesos a cada característica para el cálculo de distancia por cada familia de acero, es posible afinar la manera en la que los lotes son generados, en función de unas características u otras. Por poner un ejemplo, para algunos tipos de familia interesa más que los productos que se agrupan se parezcan más en espesor, lo cual es tan sencillo como dar más peso a ese parámetro para esa familia concreta. No se ha incluido dentro del ámbito de esta tesis el parametrizar de manera óptima cada familia para obtener los resultados deseados por los expertos, pero se considera un ejercicio interesante de cara al futuro.

El tiempo total de ejecución ha sido de 244 segundos, es decir, unos 4 minutos, lo cual concuerda totalmente con los requisitos definidos para el algoritmo.

Para exponer un ejemplo de la influencia de la distancia máxima permitida y del comportamiento del algoritmo, se ha realizado otra ejecución con los mismos datos de entrada, y los mismos parámetros, salvo por el valor de distancia máxima permitida que pasa a ser 1 en lugar de 1'5. Se desea obtener por lo tanto lotes más similares, lo cual probablemente habrá de repercutir en dejar más elementos sueltos y un mayor número de lotes. La comparativa de ambas ejecuciones se puede ver en la Tabla 32.

	Lotes	Elementos sin asignar	Suma max. dist. lotes	Media max. dist. lotes
Distancia 1'5	291	601	272	0.93
Distancia 1	318	947	255	0.80

Tabla 32. Comparativa de algoritmo ante un caso real con threshold = 1'5 y threshold = 1

Como se puede observar, el comportamiento del algoritmo es el esperado. El realizar lotes más homogéneos conlleva dejar más elementos sin asignar a los mismos. Si se comparan las medias de las distancias máximas de todos los lotes generados, ésta es menor en el caso en que la distancia máxima permitida es 1 (0.8), frente al caso con distancia máxima permitida 1'5 (0.93). En el día a día es necesario por lo tanto llegar a un compromiso entre el porcentaje de elementos que se desean dejar sin asignar, frente a cómo de buenos han de ser los lotes en términos de homogeneidad.

En el siguiente capítulo se muestran las conclusiones extraídas durante el desarrollo de esta tesis, junto con las principales aportaciones de la misma.

9 Conclusiones y líneas de futuro

En este último capítulo de la tesis se resumen las principales aportaciones de la misma, junto a las conclusiones derivadas. Finalmente se presentan futuras líneas de investigación surgidas a lo largo de la tarea de estudio realizada.

9.1 Resumen de las principales aportaciones

Las principales aportaciones de este trabajo al estado del arte de la literatura científica cubren todos los objetivos planteados en la tesis y además incluyen algunos adicionales surgidos durante el desarrollo del trabajo. Se enumeran todas a continuación:

- Análisis de los principales estudios existentes sobre las variantes de subastas en el sector del acero para analizar el rendimiento de las mismas, los principales factores de influencia y la idoneidad de agrupar en lotes.
- Propuesta de alternativas a utilizar durante el proceso de venta de productos “non-prime” para mejorar el proceso de manera global.
- Análisis de las variantes de problemas de agrupación de productos en lotes abordados en la literatura, y del estado del arte de las diferentes técnicas y estrategias utilizadas para resolverlos.
- Elaboración de una nueva clasificación de los problemas de bin packing existentes en la literatura, distinguiendo los siguiente tipos: versiones duales del problema, variaciones por tamaño de recipiente, variación en la forma de asignación de los artículos, variaciones en la restricción de los artículos y generalizaciones del problema.
- Definición del problema de empaquetado de lotes homogéneos con mínimo peso, junto con su formulación matemática.

- Diseño e implementación de una solución que permita abordar en un tiempo razonable el problema de agrupación de productos siderúrgicos en lotes lo más homogéneos posible, permitiendo dar más prioridad a unas características de los productos que a otras a la hora de formar lotes, considerando tanto la heterogeneidad de los productos como las restricciones de tamaño mínimo y máximo de lote.
- Desarrollo de un método que permita comparar cómo de parecidos son dos o varios productos en función de sus múltiples características.
- Diseño e implementación de una solución basada en 3 fases: cálculo de la distancia entre productos, clusterización por homogeneidad y asignación final de productos a lotes.
- Diseño e implementación de 3 heurísticos que combinan homogeneidad y peso (FIFO, Greedy y basado en Distancias), junto con un meta-heurístico GRASP para la parte final de asignación de productos a lotes.
- Comprobación de la influencia de los parámetros del algoritmo sobre los objetivos de la solución: número de lotes frente a homogeneidad de los mismos.
- Generación de instancias para validar los resultados del modelo basándose en instancias de la literatura y en análisis de datos reales.
- Comparativa de los heurísticos y el meta-heurístico GRASP desarrollados, considerando tanto la calidad de las soluciones como el tiempo de ejecución.
- Comprobación de la aplicabilidad del algoritmo desarrollado en un caso real.

Fruto de las anteriores aportaciones, se ha llegado a las conclusiones que se enumeran a continuación:

- Del análisis teórico de subastas se concluye que es coherente el que la subasta utilizada en la empresa del análisis de la tesis sea cerrada y con precio reserva. El hecho de que sea de primer precio va más enfocado a aumentar la confianza de los pujadores que al rendimiento real que se puede esperar.

- Se propone como recomendación realizar una subasta descendente para aquellos artículos que no se han vendido en la subasta inicial, antes de que esos productos sean convertidos en chatarra para aumentar el beneficio global del sistema de venta.
- Del exhaustivo estudio de la literatura existente se comprueba que no existe ningún problema similar al planteado. La naturaleza del problema hace imposible (como se comprobó empíricamente) la resolución del mismo de manera óptima.
- La comparativa de los heurísticos realizados basándose en los clásicos de la literatura (FIFO y Greedy), junto con el desarrollado basado en distancias para el problema a resolver muestra que las mejores soluciones se consiguen con el heurístico Greedy.
- De las pruebas realizadas con el algoritmo propuesto sobre las 80 instancias de prueba generadas se concluye que el algoritmo GRASP es la mejor opción para la asignación final de los productos a los lotes, ya que con él se mejoran las soluciones de los heurísticos individuales en un 84% de las instancias, para el mismo tiempo de ejecución. Como generador de soluciones iniciales, el heurístico Fifo es el que presenta mejores resultados, seguido del heurístico basado en distancias.
- Para su aplicación al caso real se propone lanzar la tercera fase del algoritmo 2 veces, una con soluciones iniciales generadas por el FIFO y otra con soluciones iniciales generadas con el heurístico de distancias. De esta forma, se consigue un tiempo de computación para 26 plantas y 2771 productos de unos 4 minutos y se garantiza una muy buena solución.
- La propia naturaleza del problema hace necesario llegar a un compromiso entre los elementos que se quieren dejar sin asignar, frente a cómo de buenos han de ser los lotes.

9.2 Líneas de futuro

Del desarrollo de esta tesis se pueden plantear las siguientes líneas de futuro:

- Realizar un estudio real utilizando la subasta cerrada de segundo precio, para poder comprobar si el rendimiento teórico coincide con la realidad.
- Realizar un análisis de la influencia de los coeficientes de la función de coste, para dar más peso al número de lotes o a la homogeneidad de los mismos, y poder plantear una parametrización en función de los objetivos deseados.
- Realizar un estudio junto con los expertos de proceso de la mejor parametrización de los pesos en el cálculo de la distancia en función de la familia del producto, para poder dar más importancia a unas características u otras a la hora de generar los lotes.

Anexo

Apéndice A. Instancias para las pruebas previas buscando la solución óptima

Para las pruebas previas se realizaron diferentes experimentos con 5, 15, 25, 35 y 40 elementos. Se presentan a continuación las características (distancias entre productos, pesos de los productos y límites mínimo y máximo de los lotes) para el fichero de 40 productos. El resto de ficheros han sido generados a partir de éste, considerando solamente los n primeros productos del mismo.

Pesos de los productos (toneladas):

Producto	Peso	Producto	Peso
0	5.06	20	2.87
1	6.35	21	6.69
2	3.51	22	5.32
3	4	23	6.93
4	4.48	24	8.07
5	2.31	25	8.6
6	3.34	26	2.67
7	17.11	27	7.67
8	17.71	28	1.71
9	25.27	29	1.87
10	19.76	30	3.15
11	20.2	31	1.47
12	19.66	32	8.62
13	27.68	33	10.09
14	16.27	34	12.27
15	16.14	35	5.04
16	27.62	36	9.19
17	22.2	37	13.42
18	6.46	38	1.32
19	10.87	39	2.19

Tabla 33. Pesos de los productos utilizados para las pruebas previas

Límite inferior del lote: 16 toneladas

Límite superior del lote: 24 toneladas

Distancias entre productos:

a	b	dist	a	b	dist	a	b	dist	a	b	dist	a	b	dist	a	b	dist
0	1	0.09	3	20	0.24	7	16	1.18	11	28	1.48	16	32	1.18	23	29	1.30
0	2	0.20	3	21	0.20	7	17	0.66	11	29	1.43	16	33	1.13	23	30	1.41
0	3	0.20	3	22	0.23	7	18	1.30	11	30	1.39	16	34	0.89	23	31	1.24
0	4	0.17	3	23	0.27	7	19	0.75	11	31	1.40	16	35	1.16	23	32	0.38
0	5	0.12	3	24	0.78	7	20	1.37	11	32	0.55	16	36	0.94	23	33	0.46
0	6	0.52	3	25	0.78	7	21	1.36	11	33	0.47	16	37	1.34	23	34	0.93
0	7	1.44	3	26	0.96	7	22	1.34	11	34	0.40	16	38	1.55	23	35	0.91
0	8	0.79	3	27	0.76	7	23	1.34	11	35	0.64	16	39	1.51	23	36	0.46
0	9	1.12	3	28	1.37	7	24	1.09	11	36	0.71	17	18	1.29	23	37	1.11
0	10	0.88	3	29	1.40	7	25	1.08	11	37	1.22	17	19	0.71	23	38	0.28
0	11	0.94	3	30	1.50	7	26	1.14	11	38	1.03	17	20	1.48	23	39	0.24
0	12	0.94	3	31	1.33	7	27	1.09	11	39	1.01	17	21	1.40	24	25	0.11
0	13	1.30	3	32	0.34	7	28	1.16	12	13	0.52	17	22	1.40	24	26	0.29
0	14	1.15	3	33	0.43	7	29	1.16	12	14	0.58	17	23	1.44	24	27	0.16
0	15	0.96	3	34	0.94	7	30	1.20	12	15	0.21	17	24	1.32	24	28	1.04
0	16	1.44	3	35	0.87	7	31	1.17	12	16	0.75	17	25	1.29	24	29	1.08
0	17	1.54	3	36	0.65	7	32	1.37	12	17	1.06	17	26	1.41	24	30	1.17
0	18	0.45	3	37	1.23	7	33	1.38	12	18	0.60	17	27	1.27	24	31	1.06
0	19	1.11	3	38	0.20	7	34	1.38	12	19	1.11	17	28	1.43	24	32	0.70
0	20	0.20	3	39	0.21	7	35	1.51	12	20	0.89	17	29	1.36	24	33	0.74
0	21	0.22	4	5	0.18	7	36	1.11	12	21	0.74	17	30	1.29	24	34	0.93
0	22	0.23	4	6	0.54	7	37	1.08	12	22	0.77	17	31	1.36	24	35	1.00
0	23	0.18	4	7	1.47	7	38	1.50	12	23	0.81	17	32	1.30	24	36	0.20
0	24	0.68	4	8	0.69	7	39	1.48	12	24	0.86	17	33	1.26	24	37	1.03
0	25	0.69	4	9	1.10	8	9	0.68	12	25	0.81	17	34	1.12	24	38	0.74
0	26	0.88	4	10	0.81	8	10	0.26	12	26	1.05	17	35	1.27	24	39	0.71
0	27	0.70	4	11	0.85	8	11	0.23	12	27	0.78	17	36	1.24	25	26	0.32
0	28	1.32	4	12	0.86	8	12	0.29	12	28	1.45	17	37	1.24	25	27	0.08
0	29	1.37	4	13	1.27	8	13	0.76	12	29	1.40	17	38	1.61	25	28	1.05
0	30	1.50	4	14	1.11	8	14	0.81	12	30	1.36	17	39	1.59	25	29	1.07
0	31	1.31	4	15	0.85	8	15	0.31	12	31	1.37	18	19	1.01	25	30	1.16
0	32	0.45	4	16	1.43	8	16	1.00	12	32	0.56	18	20	0.32	25	31	1.06
0	33	0.54	4	17	1.50	8	17	1.18	12	33	0.48	18	21	0.23	25	32	0.68
0	34	1.04	4	18	0.36	8	18	0.52	12	34	0.38	18	22	0.22	25	33	0.71

0	35	1.01	4	19	1.08	8	19	1.08	12	35	0.62	18	23	0.35	25	34	0.88
0	36	0.59	4	20	0.21	8	20	0.77	12	36	0.69	18	24	0.55	25	35	0.95
0	37	1.18	4	21	0.16	8	21	0.60	12	37	1.21	18	25	0.53	25	36	0.19
0	38	0.14	4	22	0.19	8	22	0.65	12	38	1.03	18	26	0.72	25	37	1.03
0	39	0.11	4	23	0.23	8	23	0.70	12	39	1.01	18	27	0.48	25	38	0.75
1	2	0.19	4	24	0.73	8	24	0.91	13	14	0.47	18	28	1.22	25	39	0.72
1	3	0.18	4	25	0.73	8	25	0.88	13	15	0.65	18	29	1.21	26	27	0.31
1	4	0.14	4	26	0.92	8	26	1.13	13	16	0.27	18	30	1.26	26	28	1.00
1	5	0.16	4	27	0.72	8	27	0.85	13	17	1.03	18	31	1.17	26	29	1.02
1	6	0.49	4	28	1.34	8	28	1.50	13	18	0.98	18	32	0.18	26	30	1.10
1	7	1.40	4	29	1.38	8	29	1.47	13	19	1.32	18	33	0.24	26	31	1.02
1	8	0.71	4	30	1.48	8	30	1.46	13	20	1.24	18	34	0.61	26	32	0.89
1	9	1.04	4	31	1.31	8	31	1.43	13	21	1.12	18	35	0.60	26	33	0.93
1	10	0.79	4	32	0.33	8	32	0.40	13	22	1.13	18	36	0.37	26	34	1.03
1	11	0.85	4	33	0.42	8	33	0.32	13	23	1.17	18	37	1.11	26	35	1.06
1	12	0.86	4	34	0.92	8	34	0.49	13	24	0.97	18	38	0.49	26	36	0.43
1	13	1.22	4	35	0.87	8	35	0.63	13	25	0.92	18	39	0.47	26	37	1.13
1	14	1.08	4	36	0.61	8	36	0.72	13	26	1.12	19	20	1.08	26	38	0.90
1	15	0.87	4	37	1.20	8	37	1.24	13	27	0.91	19	21	1.04	26	39	0.87
1	16	1.37	4	38	0.20	8	38	0.88	13	28	1.52	19	22	1.04	27	28	1.04
1	17	1.48	4	39	0.19	8	39	0.86	13	29	1.46	19	23	1.06	27	29	1.06
1	18	0.37	5	6	0.46	9	10	0.46	13	30	1.39	19	24	1.14	27	30	1.12
1	19	1.07	5	7	1.44	9	11	0.49	13	31	1.47	19	25	1.12	27	31	1.04
1	20	0.18	5	8	0.83	9	12	0.47	13	32	1.00	19	26	1.25	27	32	0.64
1	21	0.14	5	9	1.15	9	13	0.22	13	33	0.94	19	27	1.11	27	33	0.68
1	22	0.17	5	10	0.91	9	14	0.43	13	34	0.72	19	28	1.25	27	34	0.82
1	23	0.13	5	11	0.97	9	15	0.62	13	35	1.01	19	29	1.25	27	35	0.88
1	24	0.65	5	12	0.97	9	16	0.37	13	36	0.87	19	30	1.29	27	36	0.17
1	25	0.65	5	13	1.33	9	17	1.04	13	37	1.29	19	31	1.20	27	37	1.04
1	26	0.86	5	14	1.14	9	18	0.84	13	38	1.41	19	32	1.02	27	38	0.75
1	27	0.65	5	15	0.97	9	19	1.22	13	39	1.38	19	33	1.02	27	39	0.72
1	28	1.31	5	16	1.46	9	20	1.06	14	15	0.61	19	34	1.16	28	29	0.19
1	29	1.35	5	17	1.55	9	21	0.95	14	16	0.46	19	35	1.19	28	30	0.47
1	30	1.47	5	18	0.42	9	22	0.96	14	17	1.05	19	36	1.06	28	31	1.02
1	31	1.29	5	19	1.11	9	23	0.98	14	18	0.78	19	37	1.08	28	32	1.33
1	32	0.37	5	20	0.12	9	24	0.78	14	19	1.23	19	38	1.15	28	33	1.36
1	33	0.46	5	21	0.24	9	25	0.74	14	20	1.02	19	39	1.13	28	34	1.43
1	34	0.95	5	22	0.21	9	26	0.96	14	21	0.96	20	21	0.18	28	35	1.44
1	35	0.94	5	23	0.21	9	27	0.74	14	22	0.94	20	22	0.12	28	36	1.09
1	36	0.53	5	24	0.66	9	28	1.41	14	23	1.00	20	23	0.19	28	37	0.54
1	37	1.15	5	25	0.67	9	29	1.37	14	24	0.69	20	24	0.56	28	38	1.33
1	38	0.21	5	26	0.82	9	30	1.34	14	25	0.63	20	25	0.57	28	39	1.31

1	39	0.17	5	27	0.67	9	31	1.37	14	26	0.75	20	26	0.73	29	30	0.28
2	3	0.02	5	28	1.28	9	32	0.86	14	27	0.59	20	27	0.56	29	31	1.01
2	4	0.06	5	29	1.33	9	33	0.82	14	28	1.26	20	28	1.22	29	32	1.32
2	5	0.20	5	30	1.46	9	34	0.72	14	29	1.19	20	29	1.26	29	33	1.34
2	6	0.57	5	31	1.26	9	35	0.99	14	30	1.13	20	30	1.37	29	34	1.35
2	7	1.51	5	32	0.46	9	36	0.69	14	31	1.20	20	31	1.20	29	35	1.36
2	8	0.72	5	33	0.55	9	37	1.17	14	32	0.86	20	32	0.39	29	36	1.10
2	9	1.15	5	34	1.02	9	38	1.23	14	33	0.83	20	33	0.48	29	37	0.60
2	10	0.84	5	35	0.96	9	39	1.19	14	34	0.59	20	34	0.92	29	38	1.38
2	11	0.89	5	36	0.57	10	11	0.13	14	35	0.80	20	35	0.87	29	39	1.37
2	12	0.90	5	37	1.19	10	12	0.09	14	36	0.61	20	36	0.46	30	31	1.06
2	13	1.31	5	38	0.09	10	13	0.54	14	37	1.19	20	37	1.14	30	32	1.37
2	14	1.16	5	39	0.06	10	14	0.60	14	38	1.22	20	38	0.20	30	33	1.37
2	15	0.88	6	7	1.21	10	15	0.26	14	39	1.19	20	39	0.18	30	34	1.27
2	16	1.48	6	8	0.84	10	16	0.77	15	16	0.85	21	22	0.08	30	35	1.29
2	17	1.53	6	9	0.91	10	17	1.08	15	17	1.10	21	23	0.17	30	36	1.17
2	18	0.39	6	10	0.80	10	18	0.56	15	18	0.58	21	24	0.61	30	37	0.75
2	19	1.10	6	11	0.87	10	19	1.09	15	19	1.12	21	25	0.60	30	38	1.51
2	20	0.24	6	12	0.85	10	20	0.83	15	20	0.88	21	26	0.81	30	39	1.50
2	21	0.21	6	13	1.08	10	21	0.68	15	21	0.74	21	27	0.59	31	32	1.27
2	22	0.24	6	14	0.78	10	22	0.71	15	22	0.77	21	28	1.27	31	33	1.29
2	23	0.28	6	15	0.85	10	23	0.75	15	23	0.84	21	29	1.30	31	34	1.33
2	24	0.78	6	16	1.15	10	24	0.81	15	24	0.91	21	30	1.39	31	35	1.33
2	25	0.78	6	17	1.37	10	25	0.77	15	25	0.85	21	31	1.24	31	36	1.08
2	26	0.96	6	18	0.36	10	26	1.02	15	26	1.07	21	32	0.24	31	37	1.14
2	27	0.77	6	19	1.09	10	27	0.74	15	27	0.80	21	33	0.33	31	38	1.31
2	28	1.37	6	20	0.35	10	28	1.43	15	28	1.46	21	34	0.81	31	39	1.30
2	29	1.40	6	21	0.43	10	29	1.39	15	29	1.39	21	35	0.80	32	33	0.09
2	30	1.50	6	22	0.36	10	30	1.37	15	30	1.34	21	36	0.46	32	34	0.60
2	31	1.33	6	23	0.42	10	31	1.36	15	31	1.37	21	37	1.13	32	35	0.60
2	32	0.35	6	24	0.27	10	32	0.51	15	32	0.53	21	38	0.30	32	36	0.51
2	33	0.44	6	25	0.29	10	33	0.45	15	33	0.45	21	39	0.27	32	37	1.16
2	34	0.94	6	26	0.39	10	34	0.45	15	34	0.24	22	23	0.17	32	38	0.51
2	35	0.87	6	27	0.26	10	35	0.66	15	35	0.46	22	24	0.55	32	39	0.49
2	36	0.65	6	28	1.06	10	36	0.64	15	36	0.73	22	25	0.55	33	34	0.53
2	37	1.23	6	29	1.09	10	37	1.18	15	37	1.26	22	26	0.74	33	35	0.55
2	38	0.20	6	30	1.18	10	38	0.98	15	38	1.03	22	27	0.53	33	36	0.55
2	39	0.20	6	31	1.05	10	39	0.95	15	39	1.01	22	28	1.23	33	37	1.18
3	4	0.04	6	32	0.53	10	12	0.11	15	54	0.16	22	29	1.26	33	38	0.60
3	5	0.20	6	33	0.59	10	13	0.54	16	17	1.08	22	30	1.35	33	39	0.59
3	6	0.57	6	34	0.85	10	14	0.62	16	18	1.13	22	31	1.20	34	35	0.31
3	7	1.50	6	35	0.84	10	15	0.22	16	19	1.43	22	32	0.28	34	36	0.76

3	8	0.70	6	36	0.23	10	16	0.78	16	20	1.36	22	33	0.37	34	37	1.31
3	9	1.14	6	37	1.07	10	17	1.08	16	21	1.27	22	34	0.82	34	38	1.08
3	10	0.83	6	38	0.54	10	18	0.61	16	22	1.27	22	35	0.80	34	39	1.07
3	11	0.88	6	39	0.51	10	19	1.11	16	23	1.30	22	36	0.42	35	36	0.83
3	12	0.89	7	8	1.40	10	20	0.89	16	24	0.99	22	37	1.12	35	37	1.38
3	13	1.30	7	9	1.13	10	21	0.73	16	25	0.95	22	38	0.28	35	38	1.02
3	14	1.15	7	10	1.27	10	22	0.77	16	26	1.09	22	39	0.26	35	39	1.01
3	15	0.87	7	11	1.32	10	23	0.82	16	27	0.94	23	24	0.57	36	37	1.02
3	16	1.46	7	12	1.29	10	24	0.89	16	28	1.50	23	25	0.57	36	38	0.66
3	17	1.52	7	13	1.21	10	25	0.84	16	29	1.44	23	26	0.78	36	39	0.62
3	18	0.39	7	14	1.11	10	26	1.08	16	30	1.37	23	27	0.57	37	38	1.23
3	19	1.10	7	15	1.36	10	27	0.81	16	31	1.47	23	28	1.26	37	39	1.21
															38	39	0.04

Tabla 34. Distancias entre cada par de productos utilizadas en las pruebas previas

Apéndice B. Ejemplos de instancias de Benchmarking

Instancia de prueba 120_00:

id	planta	locación	forma	familia	subfamilia	categoría	grado	espesor	anchura	peso	aceitado	#lad os rec	espesor rec1	espesor rec2
120_0_1	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.04	161 5	42	1	2	75.72	32.98
120_0_2	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.58	162 9	69	1	2	40.48	68.22
120_0_3	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	2.00	137 6	67	1	2	31.10	91.08
120_0_4	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S45 OGD	0.57	159 3	57	0	2	34.58	59.10
120_0_5	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HX7 00LAD	1.45	156 8	93	1	2	94.88	43.91
120_0_6	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 6	0.72	124 2	90	1	2	35.06	76.29
120_0_7	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S39 OGD	5.43	183 3	38	1	2	53.61	57.82
120_0_8	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.81	844	36	1	2	44.89	63.79
120_0_9	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.57	189 3	45	1	2	40.99	124.1 1
120_0_10	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.92	888	42	1	2	31.82	59.54
120_0_11	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 4	0.56	134 7	33	0	2	80.35	60.87
120_0_12	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	4.69	144 1	79	1	2	145.8 8	61.39
120_0_13	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.56	195 5	27	0	2	95.98	62.23
120_0_14	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.46	848	57	0	2	54.07	68.62
120_0_15	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.56	182 8	44	1	2	62.71	60.48
120_0_16	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	3.12	104 7	84	1	2	43.79	32.16
120_0_17	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	5.02	147 6	86	1	2	95.71	63.34
120_0_18	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S35 OGD	3.41	194 7	92	1	2	53.41	95.17
120_0_19	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.66	193 8	46	0	2	22.34	48.10
120_0_20	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.82	188 2	38	1	2	63.50	44.16
120_0_21	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.71	121 3	85	1	2	74.90	69.73
120_0_22	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	1.61	177 0	33	1	2	44.47	89.94
120_0_23	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.56	112 4	82	1	2	92.56	43.50
120_0_24	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.56	145 5	73	1	2	53.14	75.54
120_0_25	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.96	200 5	49	1	2	165.1 2	53.60
120_0_26	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 6	0.64	188 9	70	1	2	101.2 6	65.94
120_0_27	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.58	155 9	59	1	2	28.45	66.00
120_0_28	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.69	136 7	23	1	2	58.71	67.95
120_0_29	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 6	1.03	102 7	57	0	2	90.69	49.15
120_0_30	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S35 OGD	0.81	168 6	72	1	2	123.8 7	73.03

120_0_31	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	0.56	163 9	74	1	2	55.10	51.02
120_0_32	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.59	154 4	69	1	2	41.47	91.80
120_0_33	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	1.70	128 3	33	1	2	72.23	67.69
120_0_34	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	1.38	187 1	42	1	2	70.52	45.66
120_0_35	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	3.74	173 3	28	1	2	50.08	30.54
120_0_36	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	0.63	141 1	46	1	2	48.75	41.41
120_0_37	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.67	154 4	30	1	2	50.91	27.04
120_0_38	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	3.00	114 8	64	1	2	40.44	70.26
120_0_39	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.78	111 4	29	1	2	50.55	173.2 4
120_0_40	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.76	144 5	74	1	2	101.0 0	66.90
120_0_41	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.56	125 4	41	0	2	31.04	128.1 7
120_0_42	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.56	146 1	49	1	2	61.83	63.27
120_0_43	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.79	169 7	55	1	2	86.37	107.0 4
120_0_44	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	0.64	183 0	98	1	2	39.91	36.08
120_0_45	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.58	200 4	80	1	2	90.27	55.43
120_0_46	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.87	197 5	32	1	2	89.22	57.83
120_0_47	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.56	199 1	25	1	2	53.12	65.37
120_0_48	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 6	0.63	132 6	38	1	2	111.0 1	109.5 7
120_0_49	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S55 OGD	1.14	200 5	82	1	2	54.47	81.43
120_0_50	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 3	2.04	141 5	30	1	2	27.05	119.2 2
120_0_51	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.73	149 3	35	1	2	65.82	32.84
120_0_52	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 4	0.62	189 6	39	1	2	114.1 6	75.32
120_0_53	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	2.64	120 7	57	1	2	284.8 5	71.72
120_0_54	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 2	4.19	198 0	84	1	2	98.03	77.84
120_0_55	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	0.60	179 2	62	0	2	68.41	95.78
120_0_56	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	4.41	139 6	50	1	2	66.19	43.51
120_0_57	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 1	3.01	120 2	55	1	2	36.93	66.24
120_0_58	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S35 OGD	2.84	178 8	27	0	2	51.45	118.9 9
120_0_59	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	1.21	194 1	30	1	2	76.17	77.83
120_0_60	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	0.70	150 1	36	1	2	88.03	58.29
120_0_61	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 4	2.15	192 6	20	1	2	51.96	32.33
120_0_62	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.58	118 4	78	1	2	75.02	75.16
120_0_63	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.59	107 2	47	1	2	92.30	62.21
120_0_64	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	0.56	198 6	26	0	2	39.14	90.65
120_0_65	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.15	170 9	45	1	2	100.6 8	51.74
120_0_66	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	1.07	106 4	41	0	2	64.04	51.27

120_0_67	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.59	105 2	58	0	2	81.15	65.85
120_0_68	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.72	169 1	98	1	2	51.18	41.03
120_0_69	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	3.55	956	91	1	2	75.63	95.80
120_0_70	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.63	155 0	96	1	2	65.95	68.41
120_0_71	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	3.57	185 4	73	1	2	135.6 5	64.63
120_0_72	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.56	120 2	84	0	2	55.33	52.32
120_0_73	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	3.42	178 0	37	0	2	112.9 4	109.7 6
120_0_74	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	3.92	131 4	93	1	2	35.55	37.82
120_0_75	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	1.80	172 8	91	1	2	78.47	79.30
120_0_76	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.40	143 5	43	1	2	61.94	66.38
120_0_77	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HX7 00LAD	2.34	127 1	73	1	2	79.87	47.82
120_0_78	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HX7 00LAD	0.57	115 5	85	0	2	73.81	28.81
120_0_79	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	1.47	196 9	81	1	2	76.59	65.62
120_0_80	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 6	1.52	119 2	79	1	2	70.05	26.48
120_0_81	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HX7 00LAD	0.98	122 3	71	1	2	47.58	66.26
120_0_82	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.56	187 1	80	1	2	53.92	53.33
120_0_83	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.57	125 4	76	0	2	41.94	53.96
120_0_84	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.07	105 1	83	1	2	47.07	69.96
120_0_85	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.57	176 7	41	1	2	68.73	62.06
120_0_86	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.81	182 2	78	1	2	92.84	83.76
120_0_87	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	2.85	186 9	70	1	2	76.79	67.33
120_0_88	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_S45 OGD	0.57	198 6	23	1	2	56.84	41.66
120_0_89	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	2.60	875	42	1	2	35.49	68.03
120_0_90	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HX7 00LAD	0.71	960	87	1	2	64.94	113.6 6
120_0_91	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.56	886	43	1	2	89.38	31.95
120_0_92	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	1.03	177 1	84	0	2	86.03	72.47
120_0_93	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	1.23	190 5	60	1	2	55.28	81.72
120_0_94	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.91	161 3	55	1	2	84.92	67.24
120_0_95	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.86	177 7	49	1	2	47.01	19.56
120_0_96	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	0.86	117 6	78	1	2	53.89	23.31
120_0_97	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.54	164 6	73	1	2	88.59	84.68
120_0_98	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.57	186 8	62	1	2	65.09	94.02
120_0_99	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	2.34	874	36	1	2	78.07	88.06
120_0_100	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.96	173 1	44	1	2	51.96	63.84
120_0_101	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S22 OGD	1.62	150 7	94	1	2	116.4 9	64.22
120_0_102	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.97	111 9	69	1	2	31.50	62.46

120_0_103	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.06	147 9	32	0	2	79.85	29.31
120_0_104	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.53	121 6	96	1	2	49.72	37.70
120_0_105	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	0.59	192 6	70	0	2	134.1 7	46.61
120_0_106	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.57	101 1	84	1	2	76.59	64.66
120_0_107	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_HCT 590X	0.59	198 7	58	1	2	47.54	79.13
120_0_108	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S35 OGD	4.53	126 8	78	1	2	62.25	94.56
120_0_109	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	1.28	174 4	25	1	2	116.9 0	76.59
120_0_110	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.75	169 6	80	1	2	28.61	59.26
120_0_111	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	1.09	174 0	58	1	2	45.31	48.88
120_0_112	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_S35 OGD	0.56	137 3	66	1	2	97.38	62.77
120_0_113	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 3	0.58	101 5	83	0	2	57.05	72.05
120_0_114	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_S22 OGD	2.30	176 2	24	1	2	45.67	44.81
120_0_115	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	2.25	144 4	98	1	2	44.80	42.67
120_0_116	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_ZM	1	HD_TYPE_DX5 1	1.07	154 8	60	1	2	37.29	75.00
120_0_117	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	3.83	144 1	42	1	2	48.09	60.39
120_0_118	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 4	1.39	200 3	43	0	2	44.46	69.28
120_0_119	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 2	0.64	943	43	1	2	69.80	57.20
120_0_120	1	LOCATI ON_1	COIL_ SLIT	HD_ MISC	HD_Z	1	HD_TYPE_DX5 1	4.26	130 1	39	1	2	85.62	64.01

Tabla 35. Instancia de prueba 120_00

Instancia de prueba 250_00:

id	planta	location	form	familia	subfamilia	cat	grado	espesor	ancho	peso	aceitado	#lados rec	espesor rec1	espesor rec2
250_0_1	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	3.51	1952	42	1	2	16.91	51.06
250_0_2	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.76	1371	69	1	2	37.86	19.52
250_0_3	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	3.22	1071	67	1	2	54.91	67.03
250_0_4	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	2.63	1202	57	0	2	19.07	51.88
250_0_5	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	0.60	1034	93	1	2	55.86	36.54
250_0_6	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX56	1.21	1916	90	1	2	70.60	77.42
250_0_7	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.59	1417	38	1	2	90.77	50.00
250_0_8	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	4.96	1880	36	1	2	60.96	55.89
250_0_9	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.68	1422	45	1	2	92.42	50.14
250_0_10	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S220G D	0.79	991	42	0	2	61.07	57.00
250_0_11	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	5.02	1217	33	1	2	80.50	83.67
250_0_12	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.30	1139	79	1	2	11.21	44.83
250_0_13	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.32	1594	27	1	2	77.12	100.84
250_0_14	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	2.94	1978	57	1	2	93.10	68.85
250_0_15	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	2.17	1669	44	1	2	78.53	86.35
250_0_16	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.50	1623	84	1	2	42.61	57.60
250_0_17	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.97	1908	86	1	2	56.36	67.97
250_0_18	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	3.29	1542	92	0	2	111.85	148.99
250_0_19	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	0.60	1955	46	1	2	42.02	64.15
250_0_20	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	4.10	1651	38	1	2	26.50	87.21
250_0_21	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.52	1050	85	1	2	69.53	74.53
250_0_22	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	0.57	1881	33	1	2	67.50	40.15
250_0_23	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.82	860	82	0	2	103.11	56.72
250_0_24	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	1.60	1822	73	0	2	121.10	86.81
250_0_25	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	3.24	964	49	1	2	64.13	98.71
250_0_26	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.56	1652	70	1	2	43.05	35.25
250_0_27	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.60	1978	59	0	2	57.71	58.32
250_0_28	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.50	2000	23	1	2	110.18	70.32
250_0_29	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.56	1789	57	0	2	61.72	20.38
250_0_30	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	0.83	2009	72	1	2	39.03	38.33
250_0_31	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	0.57	1995	74	1	2	76.41	49.42
250_0_32	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.99	1489	69	1	2	46.74	81.63
250_0_33	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.57	1898	33	1	2	70.70	65.90
250_0_34	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1313	42	1	2	43.19	53.66
250_0_35	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.63	1712	28	1	2	82.97	68.72
250_0_36	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.06	1960	46	0	2	70.17	84.61
250_0_37	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.56	1887	30	1	2	68.36	76.85
250_0_38	1	LOCATION_1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	2.39	1603	64	1	2	39.45	32.01

250_0_39	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.90	1620	29	1	2	79.46	76.33
250_0_40	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	1.05	1414	74	1	2	75.56	111.3 7
250_0_41	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.49	1529	41	0	2	33.76	104.8 4
250_0_42	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	5.18	1860	49	1	2	109.1 3	48.72
250_0_43	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	2.56	1464	55	1	2	63.90	111.3 2
250_0_44	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	1.62	1353	98	0	2	74.72	94.42
250_0_45	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.70	1038	80	1	2	56.22	51.55
250_0_46	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	2.70	904	32	1	2	50.20	45.18
250_0_47	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	3.81	1523	25	1	2	83.66	37.13
250_0_48	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.09	1278	38	1	2	44.45	53.55
250_0_49	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.98	1058	82	0	2	72.28	56.52
250_0_50	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	2.54	2007	30	1	2	92.62	56.67
250_0_51	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.16	1250	35	1	2	59.20	55.01
250_0_52	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	979	39	1	2	58.14	60.56
250_0_53	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.86	1994	57	1	2	35.25	91.24
250_0_54	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.22	1690	84	0	2	78.98	43.91
250_0_55	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	0.59	908	62	1	2	95.20	66.09
250_0_56	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.63	1460	50	1	2	78.58	41.88
250_0_57	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	0.60	1578	55	0	2	47.59	73.86
250_0_58	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	0.57	1853	27	0	2	85.11	39.11
250_0_59	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	2.17	1367	30	0	2	78.04	73.19
250_0_60	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.48	844	36	1	2	56.80	92.01
250_0_61	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.81	1866	20	1	2	63.49	143.4 4
250_0_62	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.60	1739	78	1	2	71.56	50.27
250_0_63	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	0.57	1323	47	1	2	38.37	64.65
250_0_64	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.73	991	26	1	2	75.16	67.66
250_0_65	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.66	1977	45	1	2	77.94	66.09
250_0_66	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.58	1615	41	1	2	39.93	60.84
250_0_67	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.46	1685	58	1	2	69.08	53.52
250_0_68	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.65	1212	98	1	2	127.9 3	49.30
250_0_69	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	2.10	2005	91	1	2	56.25	52.82
250_0_70	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.61	1906	96	1	2	69.53	72.89
250_0_71	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCTS9 OX	2.39	883	73	1	2	64.43	72.26
250_0_72	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	0.72	1951	84	0	2	25.19	127.3 4
250_0_73	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	2.88	1441	37	1	2	60.88	66.23
250_0_74	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.57	1102	93	1	2	73.68	45.08
250_0_75	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX52	0.59	1973	91	1	2	63.05	62.27
250_0_76	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.56	1739	43	1	2	64.93	69.42
250_0_77	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.56	1894	73	1	2	65.53	56.20
250_0_78	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.59	1528	85	1	2	35.43	108.3 7

250_0_79	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	0.87	1962	81	0	2	70.38	75.12
250_0_80	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	3.27	1688	79	0	2	81.37	67.86
250_0_81	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.28	1821	71	1	2	74.33	90.52
250_0_82	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	2.32	970	80	1	2	65.66	50.83
250_0_83	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	0.84	916	76	1	2	128.6 6	71.34
250_0_84	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	5.64	1550	83	1	2	47.35	51.31
250_0_85	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.10	1355	41	1	2	61.22	64.78
250_0_86	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	0.83	1946	78	1	2	71.74	69.69
250_0_87	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.66	1920	70	0	2	111.7 6	18.90
250_0_88	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.27	1917	23	1	2	24.95	58.59
250_0_89	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.10	1854	42	1	2	47.30	46.67
250_0_90	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.26	933	87	1	2	68.63	68.06
250_0_91	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.57	1773	43	1	2	84.03	76.25
250_0_92	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	0.59	1949	84	1	2	59.78	85.61
250_0_93	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.90	1989	60	1	2	66.87	41.76
250_0_94	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.58	1563	55	1	2	98.93	69.51
250_0_95	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	1.83	1234	49	0	2	49.44	50.98
250_0_96	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	1.40	1911	78	1	2	74.09	64.95
250_0_97	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.49	1961	73	1	2	111.4 7	93.73
250_0_98	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	1.39	1812	62	1	2	75.91	28.71
250_0_99	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.57	1316	36	1	2	54.75	85.98
250_0_10 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.61	1882	44	1	2	59.88	46.63
250_0_10 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.27	1059	94	1	2	130.5 4	37.90
250_0_10 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	4.10	1848	69	1	2	86.19	24.54
250_0_10 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	3.55	840	32	1	2	31.95	59.16
250_0_10 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.75	1901	96	1	2	61.09	81.60
250_0_10 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.69	1885	70	1	2	69.67	41.75
250_0_10 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.56	1794	84	0	2	39.24	65.82
250_0_10 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	2.01	1215	58	1	2	1.00	61.30
250_0_10 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.96	1525	78	0	2	46.96	39.69
250_0_10 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.58	908	25	1	2	62.34	84.16
250_0_11 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	4.53	1939	80	1	2	83.37	51.67
250_0_11 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	1.23	1228	58	1	2	94.76	75.47
250_0_11 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	5.41	1728	66	1	2	114.0 3	66.85
250_0_11 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	0.95	904	83	1	2	53.81	54.69
250_0_11 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S550G D	0.60	1146	24	1	2	60.37	69.31
250_0_11 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	1.28	1928	98	1	2	43.99	60.33
250_0_11 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.24	913	60	1	2	145.8 4	51.65
250_0_11 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	5.48	1182	42	1	2	69.74	91.04
250_0_11 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	0.57	899	43	0	2	32.50	40.61

250_0_11 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.58	1554	43	1	2	85.14	36.34
250_0_12 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.82	1779	39	0	2	47.10	81.05
250_0_12 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.98	1890	97	1	2	29.77	40.67
250_0_12 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.25	1898	57	1	2	60.82	44.68
250_0_12 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	3.37	1155	81	1	2	84.64	57.71
250_0_12 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1675	62	0	2	203.4 2	62.10
250_0_12 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	3.92	1852	75	1	2	49.83	65.19
250_0_12 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	0.67	1488	81	1	2	31.26	58.48
250_0_12 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1824	23	1	2	69.36	69.32
250_0_12 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	0.56	1396	43	1	2	53.99	105.8 8
250_0_12 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	2.39	1103	50	1	2	55.10	48.34
250_0_13 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.56	1906	38	1	2	67.53	96.73
250_0_13 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	1.50	1517	60	1	2	44.81	47.57
250_0_13 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	2.41	1587	58	0	2	28.01	68.10
250_0_13 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	1.82	1250	70	1	2	75.22	106.7 3
250_0_13 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.59	1498	88	1	2	13.29	32.92
250_0_13 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.56	1591	36	1	2	70.87	31.35
250_0_13 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.58	1593	90	1	2	54.59	43.49
250_0_13 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.57	942	37	1	2	81.24	59.67
250_0_13 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	2.35	1118	45	0	2	60.49	35.66
250_0_13 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.74	1724	45	0	2	76.64	85.30
250_0_14 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.71	1873	39	1	2	57.37	46.57
250_0_14 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.86	1509	44	1	2	114.7 6	48.25
250_0_14 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	4.98	1883	53	1	2	72.74	65.78
250_0_14 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX56	0.71	880	70	1	2	78.84	32.08
250_0_14 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	3.91	1287	24	1	2	73.02	51.79
250_0_14 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.86	2007	82	1	2	81.93	68.41
250_0_14 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.64	950	81	1	2	25.29	48.79
250_0_14 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.07	1999	47	1	2	66.79	72.56
250_0_14 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	3.52	1441	97	1	2	59.06	38.60
250_0_14 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	0.82	1059	35	1	2	48.08	49.79
250_0_15 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S550G D	0.56	1214	65	1	2	85.26	72.55
250_0_15 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.71	1075	74	1	2	90.61	77.66
250_0_15 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX54	1.17	1054	68	1	2	108.9 6	133.1 3
250_0_15 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.68	1414	49	1	2	58.11	49.72
250_0_15 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.80	1963	55	1	2	73.82	41.77
250_0_15 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	4.94	1892	52	0	2	68.04	46.79
250_0_15 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.10	1827	94	0	2	52.54	83.82
250_0_15 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.67	867	95	0	2	68.90	104.7 9
250_0_15 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	0.57	1317	29	1	2	87.60	33.71

250_0_15 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX56	1.68	945	99	0	2	68.17	62.94
250_0_16 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.88	1973	20	1	2	89.96	54.93
250_0_16 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	2.20	1768	22	0	2	126.3 5	74.31
250_0_16 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.05	1705	25	1	2	82.32	63.13
250_0_16 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	2.47	1462	49	1	2	61.12	57.56
250_0_16 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	0.62	1302	46	1	2	50.11	67.95
250_0_16 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.60	1171	98	1	2	60.16	63.73
250_0_16 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	0.89	1082	59	0	2	66.75	54.39
250_0_16 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.72	962	98	0	2	67.89	59.56
250_0_16 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.59	1742	60	1	2	49.02	66.97
250_0_16 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.84	978	23	1	2	64.49	57.87
250_0_17 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	1.58	1705	72	0	2	53.66	67.34
250_0_17 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.99	1617	33	1	2	66.54	90.25
250_0_17 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.65	903	98	0	2	54.10	52.66
250_0_17 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.03	842	80	1	2	83.90	105.8 8
250_0_17 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	1.41	1010	95	1	2	85.44	38.30
250_0_17 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.59	1292	78	1	2	75.99	70.29
250_0_17 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.56	915	57	1	2	83.82	66.09
250_0_17 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.66	1095	67	0	2	86.23	58.04
250_0_17 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.46	1834	53	1	2	50.25	66.09
250_0_17 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	4.19	1774	47	1	2	32.73	73.93
250_0_18 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.72	1589	53	0	2	38.60	41.71
250_0_18 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.64	1590	36	1	2	78.94	18.05
250_0_18 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.02	1966	38	1	2	63.74	62.19
250_0_18 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.16	1587	92	1	2	55.40	25.92
250_0_18 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.88	1426	30	1	2	55.96	49.00
250_0_18 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	1.38	1163	80	1	2	51.06	57.01
250_0_18 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	0.95	1642	32	1	2	72.47	32.79
250_0_18 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.09	1138	97	1	2	68.38	37.62
250_0_18 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	4.00	1857	39	1	2	67.62	44.57
250_0_18 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	1.09	1318	80	1	2	75.56	84.49
250_0_19 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	4.36	1310	72	1	2	14.65	78.29
250_0_19 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.17	872	55	1	2	65.32	35.30
250_0_19 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.05	1364	41	0	2	36.78	52.25
250_0_19 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.62	1915	60	0	2	70.13	44.89
250_0_19 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	0.56	1431	67	1	2	43.88	84.08
250_0_19 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.57	1543	53	1	2	71.51	71.70
250_0_19 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S550G D	0.57	1946	65	1	2	96.11	66.42
250_0_19 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	4.06	1165	95	1	2	75.92	77.63
250_0_19 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S450G D	0.75	1999	20	0	2	57.83	45.30

250_0_19 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	4.72	1470	66	1	2	90.29	86.46
250_0_20 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	0.56	913	78	1	2	81.87	56.58
250_0_20 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.70	1586	98	1	2	46.55	79.83
250_0_20 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX56	0.98	1915	47	1	2	77.51	55.63
250_0_20 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.09	1342	100	1	2	45.27	58.17
250_0_20 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	3.79	1654	85	0	2	37.90	103.8 7
250_0_20 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.56	1520	53	1	2	32.49	72.26
250_0_20 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.61	1935	53	0	2	61.68	35.14
250_0_20 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.90	1387	67	0	2	54.80	44.18
250_0_20 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S450G D	0.56	1829	27	1	2	76.15	59.96
250_0_20 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.57	1840	22	1	2	48.86	64.32
250_0_21 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.80	1701	61	1	2	32.88	52.82
250_0_21 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	0.82	1664	43	1	2	73.62	63.94
250_0_21 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	3.59	1876	52	1	2	101.0 4	78.65
250_0_21 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S550G D	0.56	1239	76	1	2	128.4 4	53.30
250_0_21 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.71	1800	64	1	2	82.20	59.86
250_0_21 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	2.24	1911	61	1	2	51.50	26.12
250_0_21 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.64	926	29	1	2	48.66	94.54
250_0_21 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1800	30	1	2	58.58	63.25
250_0_21 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S350G D	1.02	1992	46	1	2	87.63	51.11
250_0_21 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	4.94	1255	79	0	2	52.73	105.4 8
250_0_22 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S450G D	3.08	1727	66	1	2	50.03	71.04
250_0_22 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX56	0.78	1256	27	1	2	35.46	47.38
250_0_22 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	2.87	1580	79	0	2	95.39	77.05
250_0_22 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	2.04	1560	98	1	2	107.1 4	53.36
250_0_22 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.57	910	90	1	2	49.76	55.29
250_0_22 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.85	1032	22	1	2	43.68	113.7 1
250_0_22 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.58	1968	75	1	2	53.07	75.72
250_0_22 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 DX	0.56	1688	57	1	2	66.01	55.68
250_0_22 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	1.94	1028	67	1	2	101.9 4	70.61
250_0_22 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	2.47	1264	36	1	2	24.89	89.99
250_0_23 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.05	1822	70	1	2	91.37	65.03
250_0_23 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	3.21	1703	99	1	2	73.48	88.23
250_0_23 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1301	48	1	2	54.63	51.99
250_0_23 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	0.56	1395	43	1	2	48.00	82.38
250_0_23 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	3.30	1200	45	1	2	28.98	35.45
250_0_23 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_S220G D	1.01	1521	71	1	2	56.04	80.64
250_0_23 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HX700 LAD	4.42	1937	100	1	2	42.87	67.90
250_0_23 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	1.68	1952	88	1	2	130.0 4	40.66
250_0_23 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	1.32	1999	48	1	2	54.44	56.11

250_0_23 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	4.75	1068	27	0	2	40.55	76.54
250_0_24 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_HCT59 OX	2.56	1256	39	1	2	51.84	76.20
250_0_24 1	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	3.37	1987	38	1	2	61.03	76.73
250_0_24 2	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX51	2.22	1739	100	1	2	58.83	89.57
250_0_24 3	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX53	2.19	1507	60	0	2	102.8 4	52.65
250_0_24 4	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	2.22	1948	42	1	2	56.03	22.69
250_0_24 5	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	5.30	1980	20	1	2	61.15	44.37
250_0_24 6	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX51	0.61	1986	69	0	2	95.74	73.67
250_0_24 7	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.57	1779	24	0	2	77.53	47.40
250_0_24 8	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_Z	1	HD_TYPE_DX54	0.78	1928	23	1	2	31.20	50.02
250_0_24 9	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_S220G D	0.78	1328	92	0	2	48.34	111.7 9
250_0_25 0	1	LOCATION _1	COIL_SL IT	HD_MI SC	HD_ZM	1	HD_TYPE_DX53	2.27	1892	32	1	2	106.6 6	72.76

Tabla 36. Instancia de prueba 250_00

Apéndice C. Resultados de los experimentos realizados con las instancias generadas

En este apartado se detallan los resultados de todos los experimentos realizados. Para cada uno de ellos se presentan los siguientes campos (hay que tener en cuenta que cada experimento se ha realizado 10 veces, de ahí que los valores que se detallan son los medios de las 10 ejecuciones):

- **num_singles:** número de productos sin asignar a ningún lote
- **num_bundles:** número de lotes generados por el algoritmo
- **mean_max_distances:** media de las distancias máximas de los lotes generados
- **sum_bundling_cost:** valor total de la función de coste
- **total_time:** tiempo total de CPU de de ejecución del algoritmo, expresado en segundos
- **std_num_singles:** desviación estándar del número de productos sin asignar a lotes en las 10 ejecuciones
- **std_num_bundles:** desviación estándar del número de lotes generados por el algoritmo en las 10 ejecuciones
- **std_mean_max_distances:** desviación estándar de la media de las distancias máximas de los lotes generados en las 10 ejecuciones
- **std_sum_bundling_cost:** desviación estándar del valor de la función de coste en las 10 ejecuciones
- **std_total_time:** desviación estándar del tiempo de ejecución en las 10 ejecuciones

Los valores obtenidos con cada prueba son:

FIFO 120 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	18.9	43.7	0.75	66.75	1.07	1.58	1.35	0.03	0.66	0.52
120_01.csv	18.8	43.3	0.73	66.36	1.09	1.78	1.49	0.02	0.52	0.41
120_02.csv	16.8	42.5	0.86	64.08	1.14	1.83	1.02	0.01	0.97	0.40
120_03.csv	20	43.4	0.78	67.10	1.22	1.95	1.28	0.02	0.79	0.39
120_04.csv	22.6	42.2	0.79	69.76	1.20	1.20	1.17	0.03	0.49	0.39
120_05.csv	28.9	40	0.76	74.18	1.09	1.64	1.41	0.02	0.44	0.50
120_06.csv	20	42.5	0.82	167.63	1.16	1.90	1.75	0.02	0.49	0.40
120_07.csv	18.7	44.4	0.73	67.99	1.24	1.49	0.92	0.02	0.66	0.44
120_08.csv	19.9	44.6	0.85	67.07	0.90	1.45	1.20	0.01	0.47	0.35
120_09.csv	17.6	41.6	0.82	63.22	1.36	1.56	1.11	0.03	0.64	0.38
120_10.csv	22.6	44.1	0.75	72.56	1.28	1.91	1.22	0.02	0.66	0.45
120_11.csv	22.6	42.4	0.79	65.44	1.07	0.80	1.02	0.02	0.42	0.45
120_12.csv	13.5	47.2	0.80	65.35	1.36	1.75	1.40	0.03	0.43	0.39
120_13.csv	12.6	47.3	0.83	64.33	0.95	1.43	1.00	0.02	0.66	0.49
120_14.csv	17.4	45.6	0.86	67.54	1.23	2.06	1.62	0.04	0.80	0.41
120_15.csv	25.2	39.5	0.80	70.56	1.16	1.47	1.12	0.02	0.71	0.41
120_16.csv	22.3	43.8	0.77	71.13	1.16	1.90	1.54	0.02	0.83	0.41
120_17.csv	18.7	46.8	0.81	69.66	1.29	2.19	1.72	0.02	0.65	0.40
120_18.csv	17.9	45	0.77	68.47	1.08	1.81	1.48	0.02	0.56	0.44
120_19.csv	27.4	39.9	0.78	172.09	1.10	1.28	1.14	0.03	0.48	0.49

Tabla 37. Resultados FIFO 120 productos

FIFO 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_00.csv	33.9	90.9	0.75	232.57	1.11	2.21	1.14	0.02	1.26	0.02
250_01.csv	41	91.3	0.73	141.07	1.13	2.45	2.28	0.01	0.56	0.01
250_02.csv	46.6	86.9	0.76	339.47	1.10	1.36	1.51	0.03	0.80	0.01
250_03.csv	40.2	91	0.73	237.60	1.11	2.48	1.79	0.01	0.82	0.02
250_04.csv	33.2	95.7	0.69	137.75	1.04	3.46	2.49	0.02	1.00	0.02
250_05.csv	39.3	91.3	0.73	235.98	1.13	2.15	1.73	0.01	0.85	0.01
250_06.csv	40.1	90.8	0.72	134.52	1.17	3.62	2.23	0.01	1.43	0.02
250_07.csv	33.5	96.8	0.74	134.44	1.18	1.69	1.08	0.01	0.74	0.02
250_08.csv	43.4	92.3	0.70	238.07	1.20	1.80	1.27	0.01	1.07	0.01
250_09.csv	38.6	90.7	0.75	135.14	1.18	2.97	2.65	0.02	1.14	0.01
250_10.csv	37.3	95.7	0.72	137.25	1.18	3.29	2.19	0.02	1.43	0.01
250_11.csv	33.4	94.6	0.74	134.20	1.22	3.07	2.29	0.01	0.90	0.08
250_12.csv	48.5	89.9	0.69	143.94	1.22	3.50	2.21	0.02	1.27	0.06
250_13.csv	38.7	93.6	0.67	234.24	1.20	1.35	1.20	0.01	0.67	0.01
250_14.csv	38.6	91.8	0.71	232.80	1.16	3.44	2.32	0.01	1.54	0.01
250_15.csv	50.9	87.7	0.69	140.42	1.18	1.70	1.73	0.01	0.97	0.01
250_16.csv	36.6	89.7	0.73	128.76	1.14	2.54	1.79	0.01	1.00	0.01
250_17.csv	30.4	94.4	0.69	130.72	1.17	2.11	1.43	0.02	0.86	0.02
250_18.csv	39.3	91.1	0.75	131.72	1.16	2.33	1.30	0.02	1.39	0.01
250_19.csv	37.1	93.5	0.67	135.86	1.16	3.11	2.38	0.01	1.25	0.01

Tabla 38. Resultados FIFO 250 productos

FIFO 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	82.7	177.1	0.71	371.21	3.62	4.34	2.91	0.01	1.55	0.02
500_01.csv	96	171.9	0.66	678.03	3.62	2.45	2.02	0.01	1.14	0.06
500_02.csv	90.7	177.7	0.69	578.49	3.61	2.53	2.49	0.01	1.60	0.04
500_03.csv	100.3	170.4	0.70	480.80	3.58	4.52	3.23	0.01	1.06	0.04
500_04.csv	112.3	167.1	0.64	587.55	3.68	3.16	2.55	0.00	1.00	0.13
500_05.csv	101.1	174.5	0.66	382.26	3.77	3.48	2.62	0.01	1.28	0.05
500_06.csv	86.9	182.8	0.65	580.10	3.66	5.20	3.46	0.01	1.71	0.03
500_07.csv	110.1	170.3	0.70	790.63	3.51	3.05	2.28	0.01	1.41	0.05
500_08.csv	91.4	168.9	0.63	368.80	3.71	4.29	3.39	0.01	1.09	0.07
500_09.csv	80.7	182.8	0.68	472.58	3.78	2.97	2.14	0.01	1.34	0.03
500_10.csv	94.6	172	0.66	676.39	3.60	2.20	1.48	0.01	1.07	0.02
500_11.csv	95.8	168.2	0.70	674.65	3.66	6.16	3.68	0.01	2.27	0.05
500_12.csv	89.3	176	0.68	371.15	3.65	4.31	2.24	0.01	1.72	0.03
500_13.csv	77.5	179.6	0.71	467.75	3.63	4.65	4.08	0.01	1.62	0.03
500_14.csv	100.4	170.3	0.69	376.38	3.71	2.42	2.15	0.01	1.30	0.20
500_15.csv	80.9	181.2	0.68	371.09	3.72	4.30	3.66	0.01	1.24	0.02
500_16.csv	85.8	179	0.64	472.02	3.71	3.97	2.05	0.01	2.11	0.01
500_17.csv	99.2	167.4	0.64	673.81	3.54	3.52	3.14	0.01	1.38	0.02
500_18.csv	116.7	164.8	0.63	892.98	3.65	3.29	2.82	0.01	0.90	0.09
500_19.csv	78.3	177.9	0.73	467.26	3.70	3.61	2.66	0.01	0.93	0.02

Tabla 39. Resultados FIFO 500 productos

FIFO 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	211	336.5	0.62	1262.28	13.63	5.06	3.58	0.01	1.61	0.21
1000_01.csv	218.6	334	0.61	1268.40	12.96	5.24	3.07	0.01	2.77	0.26
1000_02.csv	200.6	348.9	0.62	1166.12	13.33	4.90	3.45	0.01	1.84	0.23
1000_03.csv	188.7	355.4	0.63	858.57	13.54	3.98	3.53	0.01	1.45	0.11
1000_04.csv	195.7	336.3	0.63	945.11	13.63	7.52	4.69	0.01	3.05	0.59
1000_05.csv	185.2	344	0.65	944.50	13.24	5.27	3.58	0.01	1.91	0.23
1000_06.csv	215.8	327.4	0.62	1055.24	13.65	4.94	3.23	0.01	2.15	0.23
1000_07.csv	184.2	351.9	0.64	1051.62	13.53	4.28	3.70	0.01	1.64	0.15
1000_08.csv	223	320.5	0.62	956.44	13.26	5.12	3.67	0.01	2.09	0.07
1000_09.csv	193.2	337	0.64	740.12	13.36	3.74	2.41	0.01	1.99	0.08
1000_10.csv	203.8	334.4	0.61	1051.91	13.25	6.97	4.39	0.01	2.25	0.30
1000_11.csv	199.8	342.6	0.61	1053.58	13.04	4.38	2.73	0.01	1.68	0.14
1000_12.csv	193.8	340.9	0.65	1250.29	13.15	3.99	3.18	0.01	1.27	0.12
1000_13.csv	173.7	351.5	0.65	939.41	12.75	4.36	3.75	0.01	0.86	0.20
1000_14.csv	164.5	352.5	0.63	931.75	13.28	5.82	3.41	0.01	2.37	0.30
1000_15.csv	195.7	341.1	0.63	1052.29	13.42	5.92	3.56	0.01	2.37	0.16
1000_16.csv	178.5	354.8	0.64	946.19	13.22	3.61	2.44	0.01	1.47	0.10
1000_17.csv	210.6	334.2	0.66	955.86	13.76	5.54	3.82	0.01	2.47	0.30
1000_18.csv	213.2	335.4	0.65	1863.33	13.57	5.42	4.18	0.00	2.15	1.10
1000_19.csv	206.3	333.9	0.64	956.43	13.33	5.02	3.62	0.01	2.14	0.16

Tabla 40. Resultados FIFO 1000 productos

GREEDY 120 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	14.2	45.1	0.72	62.58	1.37	1.25	0.94	0.02	0.43	0.40
120_01.csv	17.3	42.3	0.72	63.35	1.23	1.42	1.19	0.03	0.48	0.44
120_02.csv	13.2	42.3	0.84	59.99	1.20	0.98	0.78	0.02	0.30	0.46
120_03.csv	12.5	46.5	0.77	61.93	1.21	0.81	0.81	0.02	0.20	0.38
120_04.csv	18.7	43.3	0.79	65.97	1.42	1.10	1.10	0.02	0.20	0.43
120_05.csv	24.3	41.6	0.76	70.92	1.18	0.78	0.92	0.02	0.16	0.39
120_06.csv	18.3	42.4	0.81	165.12	1.50	1.27	1.11	0.02	0.54	0.31
120_07.csv	16.7	44.4	0.73	64.77	1.20	0.64	0.66	0.02	0.45	0.35
120_08.csv	15	46	0.82	62.60	1.24	0.63	0.63	0.02	0.15	0.35
120_09.csv	13	42.7	0.81	59.08	1.48	0.77	0.78	0.02	0.44	0.42
120_10.csv	17.3	45.7	0.75	68.52	1.35	0.46	0.46	0.02	0.18	0.18
120_11.csv	20.8	41.6	0.77	62.02	1.36	0.75	0.80	0.02	0.41	0.38
120_12.csv	10.1	46.8	0.80	60.77	1.27	0.83	0.75	0.02	0.29	0.47
120_13.csv	10.1	46.6	0.82	59.77	1.28	1.76	1.56	0.01	0.33	0.40
120_14.csv	13.7	45.9	0.84	63.13	1.68	1.10	0.94	0.01	0.37	0.33
120_15.csv	18.7	42.3	0.80	66.21	1.34	0.46	0.46	0.02	0.22	0.44
120_16.csv	17.4	45.6	0.78	67.59	1.28	0.49	0.49	0.02	0.31	0.46
120_17.csv	12.9	48.1	0.81	64.74	1.29	0.70	0.70	0.02	0.24	0.48
120_18.csv	14.3	45.6	0.78	65.18	1.13	0.46	0.49	0.02	0.32	0.44
120_19.csv	24	41	0.76	168.27	1.41	1.18	1.18	0.01	0.17	0.42

Tabla 41. Resultados Greedy 120 productos

GREEDY 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_00.csv	24.2	91.6	0.76	222.34	2.01	1.54	1.28	0.01	0.26	0.06
250_01.csv	33.6	91.1	0.74	132.54	1.78	1.11	0.94	0.01	0.61	0.01
250_02.csv	36.6	87.2	0.76	328.59	1.89	1.43	0.87	0.02	0.53	0.01
250_03.csv	32.2	91.4	0.74	228.71	1.92	0.98	1.11	0.01	0.33	0.01
250_04.csv	26	97.3	0.69	130.78	1.42	0.77	0.90	0.01	0.52	0.02
250_05.csv	28.2	93.6	0.72	224.77	1.99	1.54	1.28	0.01	0.39	0.02
250_06.csv	27	93.7	0.71	123.00	2.29	1.73	1.49	0.02	0.29	0.01
250_07.csv	18.7	100.2	0.73	121.29	2.33	0.90	0.98	0.01	0.36	0.02
250_08.csv	23.3	99.7	0.70	223.66	2.53	0.90	0.90	0.01	0.22	0.01
250_09.csv	17.5	97.6	0.76	120.35	2.50	0.92	1.02	0.01	0.39	0.05
250_10.csv	19.1	101.5	0.72	123.45	2.42	1.45	1.36	0.01	0.54	0.02
250_11.csv	16.5	98.1	0.75	119.85	2.48	1.28	1.22	0.01	0.67	0.01
250_12.csv	29	96.7	0.69	129.95	2.40	0.77	0.90	0.01	0.35	0.01
250_13.csv	24.6	96.3	0.67	221.12	2.57	0.80	0.64	0.01	0.35	0.04
250_14.csv	25.2	94	0.73	220.71	2.33	1.25	1.26	0.01	0.41	0.02
250_15.csv	29.3	95.9	0.68	126.50	2.50	1.95	1.64	0.02	0.55	0.04
250_16.csv	25.5	90.5	0.73	116.62	2.18	0.92	0.92	0.01	0.18	0.01
250_17.csv	21.5	94	0.70	119.69	2.28	1.57	1.34	0.01	0.28	0.02
250_18.csv	24.9	93.8	0.76	119.00	2.34	1.14	0.75	0.02	0.46	0.01
250_19.csv	23.6	97.1	0.68	124.92	2.26	1.80	1.51	0.01	0.56	0.03

Tabla 42. Resultados Greedy 250 productos

GREEDY 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	49.3	187.5	0.71	345.09	6.36	2.69	2.16	0.00	0.77	0.09
500_01.csv	75	174.3	0.66	657.38	5.73	1.79	1.55	0.01	0.65	0.13
500_02.csv	59.8	185.9	0.70	553.95	6.18	1.54	1.30	0.00	0.50	0.03
500_03.csv	64.5	181.7	0.70	453.92	6.43	1.63	1.49	0.01	0.57	0.03
500_04.csv	84.2	177.2	0.65	567.04	5.26	2.04	1.40	0.01	0.96	0.23
500_05.csv	60.1	190	0.66	354.74	7.02	2.12	1.79	0.01	0.56	0.22
500_06.csv	52.4	194	0.65	554.49	6.31	1.56	1.18	0.01	0.74	0.13
500_07.csv	85	176	0.69	768.52	5.98	2.19	1.55	0.01	0.80	0.11
500_08.csv	73.9	169.2	0.64	349.04	6.12	1.76	1.66	0.01	0.57	0.04
500_09.csv	51.7	188.3	0.68	446.49	7.08	1.62	1.62	0.01	0.61	0.31
500_10.csv	66.1	179.7	0.66	652.97	5.61	2.66	2.10	0.01	0.82	0.29
500_11.csv	76.6	169.9	0.69	654.77	5.32	1.62	1.22	0.01	0.97	0.04
500_12.csv	53.2	187.8	0.68	344.47	6.77	1.66	1.66	0.01	0.31	0.03
500_13.csv	52.1	183.8	0.70	443.53	6.16	1.14	1.54	0.01	0.84	0.13
500_14.csv	68.6	178.7	0.69	350.69	6.15	1.28	0.90	0.01	0.74	0.03
500_15.csv	43.8	191.1	0.68	342.78	6.80	1.33	1.30	0.01	0.77	0.04
500_16.csv	59.6	186.1	0.64	450.36	5.42	1.62	1.51	0.01	0.55	0.16
500_17.csv	78.8	170.6	0.63	654.70	5.02	1.47	1.62	0.01	0.56	0.11
500_18.csv	97.4	168.9	0.63	875.27	5.11	1.28	1.14	0.01	0.51	0.08
500_19.csv	50	182.7	0.73	441.01	6.68	0.89	1.10	0.01	0.56	0.04

Tabla 43. Resultados Greedy 500 productos

GREEDY 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	150	352.3	0.64	1213.43	20.00	1.55	1.10	0.00	0.71	0.46
1000_01.csv	151	355.1	0.62	1217.99	18.89	1.18	1.58	0.01	0.73	0.19
1000_02.csv	126.7	372.3	0.63	1112.38	19.25	2.45	1.90	0.00	1.24	0.09
1000_03.csv	101.8	383.7	0.64	796.97	21.11	2.04	2.15	0.01	0.72	0.38
1000_04.csv	134.5	349.6	0.64	894.19	19.91	1.69	1.74	0.00	0.53	0.11
1000_05.csv	119.1	361.5	0.65	891.81	19.58	2.95	1.80	0.01	1.20	0.07
1000_06.csv	147.4	346.4	0.63	1002.74	20.98	1.74	1.50	0.01	0.97	0.18
1000_07.csv	109	376	0.64	995.79	20.24	1.18	1.10	0.01	0.71	0.14
1000_08.csv	159.8	339.8	0.63	908.37	19.20	2.82	1.83	0.01	1.56	0.30
1000_09.csv	115.3	361.8	0.65	684.07	21.17	2.72	1.83	0.01	1.20	0.42
1000_10.csv	142.8	349.6	0.62	1003.27	19.01	1.54	1.69	0.01	0.89	0.09
1000_11.csv	135.1	360.7	0.61	1002.96	19.26	1.92	1.79	0.00	0.40	0.35
1000_12.csv	131.2	355.6	0.65	1197.85	19.62	2.18	1.96	0.00	0.74	0.12
1000_13.csv	96.7	373.1	0.66	880.74	20.20	1.79	1.22	0.01	1.19	0.18
1000_14.csv	103.9	366.5	0.64	881.68	19.93	1.45	1.12	0.01	0.92	0.15
1000_15.csv	123.5	363.6	0.63	999.42	20.47	1.86	1.02	0.00	0.70	0.06
1000_16.csv	112.5	369.8	0.64	892.28	20.43	2.50	2.44	0.01	0.65	0.66
1000_17.csv	132	359.4	0.66	898.24	21.33	2.14	2.06	0.01	0.55	0.18
1000_18.csv	151	349.7	0.66	1811.75	20.27	1.61	1.62	0.00	0.61	0.17
1000_19.csv	143.6	349.3	0.65	905.72	20.44	2.15	1.95	0.00	0.56	0.51

Tabla 44. Resultados Greedy 1000 productos

DISTANCIAS 120 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	16.1	42.9	0.62	64.59	1.71	0.30	0.30	0.00	0.05	0.48
120_01.csv	15.9	41.2	0.60	62.50	1.85	0.83	0.60	0.01	0.13	0.40
120_02.csv	12	42	0.72	59.45	1.86	0.00	0.00	0.01	0.00	0.38
120_03.csv	15.6	43.4	0.66	64.36	1.88	0.49	0.49	0.00	0.04	0.43
120_04.csv	21	41	0.66	66.95	1.69	0.00	0.00	0.00	0.00	0.44
120_05.csv	15	44	0.71	67.37	1.41	0.00	0.00	0.00	0.00	0.43
120_06.csv	11.2	44.9	0.72	164.35	1.88	0.60	0.30	0.00	0.18	0.41
120_07.csv	13	45	0.63	64.58	1.75	0.00	0.00	0.00	0.00	0.43
120_08.csv	19	42	0.76	64.40	1.96	0.00	0.00	0.00	0.00	0.35
120_09.csv	11	43	0.72	59.53	2.10	0.00	0.00	0.00	0.00	0.37
120_10.csv	18.5	44.5	0.65	70.40	1.77	0.50	0.50	0.01	0.00	0.40
120_11.csv	18	42	0.69	62.45	1.86	0.00	0.00	0.00	0.00	0.34
120_12.csv	9.5	46.5	0.65	60.33	2.19	0.50	0.50	0.01	0.00	0.46
120_13.csv	16.6	41.6	0.68	62.36	2.02	1.80	1.20	0.01	0.38	0.36
120_14.csv	14.9	44.1	0.69	63.85	2.08	0.30	0.30	0.01	0.00	0.40
120_15.csv	19	39	0.73	65.85	2.06	0.00	0.00	0.00	0.00	0.45
120_16.csv	21	43	0.66	69.77	2.11	0.00	0.00	0.00	0.00	0.33
120_17.csv	15	46	0.70	66.31	1.95	0.00	0.00	0.00	0.00	0.44
120_18.csv	16	43	0.61	65.83	1.88	0.00	0.00	0.00	0.00	0.46
120_19.csv	23	40	0.67	168.49	2.33	0.00	0.00	0.00	0.00	0.15

Tabla 45. Resultados Distancias 120 productos

DISTANCIAS 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_00.csv	27	89	0.61	225.79	3.71	0.00	0.00	0.00	0.12	0.09
250_01.csv	36.3	85.8	0.59	134.24	3.14	0.64	0.40	0.00	0.28	0.10
250_02.csv	40.3	83.7	0.65	331.51	3.37	0.46	0.46	0.00	0.02	0.04
250_03.csv	37.2	84.2	0.61	232.06	3.50	1.47	0.98	0.00	0.39	0.03
250_04.csv	26	93	0.60	131.09	2.35	0.00	0.00	0.00	0.00	0.02
250_05.csv	32	89	0.60	227.19	3.65	0.00	0.00	0.00	0.07	0.16
250_06.csv	29.1	89.9	0.57	124.33	4.27	0.30	0.30	0.00	0.15	0.13
250_07.csv	31	91	0.55	126.54	4.37	0.00	0.00	0.00	0.01	0.14
250_08.csv	41	87	0.53	232.09	4.62	0.00	0.00	0.00	0.00	0.05
250_09.csv	41	82	0.57	130.80	4.65	0.00	0.00	0.00	0.03	0.01
250_10.csv	43.4	83.6	0.53	132.05	4.47	0.49	0.49	0.01	0.06	0.02
250_11.csv	35.2	84.8	0.54	127.55	4.64	0.40	0.40	0.00	0.15	0.11
250_12.csv	46.4	83.2	0.49	136.23	4.37	2.50	1.72	0.01	0.53	0.03
250_13.csv	41.4	83.6	0.48	227.46	4.79	0.49	0.49	0.01	0.11	0.04
250_14.csv	26.5	88.7	0.54	119.89	4.33	1.28	0.90	0.00	0.37	0.05
250_15.csv	47.2	81.9	0.53	133.23	4.50	0.60	0.30	0.01	0.19	0.01
250_16.csv	37.8	80.2	0.57	121.78	4.11	0.40	0.40	0.00	0.06	0.01
250_17.csv	27	89	0.54	122.36	4.28	0.00	0.00	0.00	0.03	0.05
250_18.csv	35.3	85.8	0.59	124.00	4.33	0.90	0.60	0.01	0.12	0.01
250_19.csv	29.7	91.3	0.52	128.42	4.12	0.46	0.46	0.01	0.03	0.02

Tabla 46. Resultados Distancias 250 productos

DISTANCIAS 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	79.3	162	0.53	352.74	11.22	1.00	0.63	0.00	0.22	0.41
500_01.csv	85.1	162	0.51	659.63	9.61	0.70	0.45	0.00	0.31	0.23
500_02.csv	80.2	166.1	0.50	457.64	10.51	0.87	0.54	0.01	0.27	0.07
500_03.csv	72.9	170.5	0.51	456.43	11.15	3.36	2.58	0.01	0.54	0.19
500_04.csv	102.4	159.7	0.49	569.80	7.99	1.36	1.10	0.01	0.16	0.05
500_05.csv	88.8	165	0.47	363.98	11.68	1.99	1.61	0.01	0.38	0.09
500_06.csv	84.6	167.6	0.49	564.00	10.38	1.28	1.02	0.00	0.22	0.05
500_07.csv	92.2	161.2	0.52	466.53	10.08	1.17	0.75	0.00	0.40	0.18
500_08.csv	71.3	164.3	0.48	346.97	10.25	2.24	1.35	0.01	0.75	0.03
500_09.csv	68.1	172.2	0.51	450.88	12.03	0.54	0.60	0.00	0.37	0.27
500_10.csv	80.8	165.2	0.50	557.47	8.84	0.40	0.40	0.00	0.00	0.06
500_11.csv	83.7	161.3	0.53	656.89	8.65	0.90	0.90	0.00	0.01	0.10
500_12.csv	77.3	163.8	0.48	250.32	11.69	0.46	0.60	0.00	0.28	0.11
500_13.csv	64.5	169.7	0.50	446.08	10.32	1.50	0.90	0.00	0.50	0.06
500_14.csv	91.8	159.3	0.48	358.02	10.49	0.60	0.46	0.00	0.21	0.13
500_15.csv	74.8	167.3	0.49	352.58	11.80	0.98	0.78	0.00	0.35	0.16
500_16.csv	80	168.3	0.48	458.29	8.62	1.48	0.90	0.00	0.36	0.12
500_17.csv	77.2	164.8	0.48	551.17	8.05	0.40	0.40	0.00	0.11	0.06
500_18.csv	109	151	0.48	873.19	7.99	0.00	0.00	0.00	0.16	0.26
500_19.csv	84.1	155.4	0.53	452.18	11.58	1.45	0.80	0.01	0.81	0.10

Tabla 47. Resultados Distancias 500 productos

DISTANCIAS 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	202.9	304.5	0.43	1123.97	29.58	1.64	1.02	0.00	0.53	0.50
1000_01.csv	191.3	315.8	0.44	1124.37	28.12	1.27	1.08	0.00	0.23	0.11
1000_02.csv	198.2	319.2	0.43	1036.23	28.74	1.40	0.98	0.00	0.29	0.60
1000_03.csv	179.2	322.6	0.42	819.89	32.05	2.14	1.43	0.00	0.51	0.38
1000_04.csv	159.5	324.4	0.43	899.18	30.24	1.50	0.80	0.00	0.65	0.23
1000_05.csv	164.5	324.8	0.44	905.46	29.82	3.56	2.82	0.00	0.88	0.32
1000_06.csv	176.5	312.2	0.41	802.45	32.48	3.20	2.04	0.01	0.75	0.67
1000_07.csv	180.6	317.5	0.44	915.25	30.35	0.66	0.50	0.00	0.29	0.11
1000_08.csv	206.8	299.1	0.43	818.54	28.29	1.40	0.83	0.00	0.59	0.75
1000_09.csv	182.4	309.7	0.44	603.46	33.65	1.20	0.90	0.00	0.22	2.90
1000_10.csv	192.8	309	0.41	919.71	28.51	2.68	1.79	0.00	0.71	0.17
1000_11.csv	165.4	327.2	0.43	909.94	28.53	1.69	1.17	0.00	0.65	0.17
1000_12.csv	159.8	319.7	0.46	797.69	30.10	1.94	1.35	0.00	0.68	0.92
1000_13.csv	151.8	330.6	0.45	901.22	31.60	2.04	1.56	0.00	0.18	0.19
1000_14.csv	151.3	326.4	0.43	795.76	30.48	2.19	1.56	0.00	0.54	0.20
1000_15.csv	181.8	315	0.45	915.76	31.70	2.44	1.55	0.00	0.79	0.73
1000_16.csv	163.9	328.6	0.43	907.88	30.81	1.92	1.28	0.00	0.29	0.08
1000_17.csv	203.7	303	0.44	819.27	32.35	1.95	1.34	0.00	0.58	0.12
1000_18.csv	189	305.4	0.44	1312.46	30.98	2.32	1.96	0.00	0.43	0.19
1000_19.csv	187.6	312.8	0.44	818.87	30.65	2.54	1.72	0.00	0.82	0.20

Tabla 48. Resultados Distancias 1000 productos

FIFO + GRASP 120 productos:

Instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	8	16.8	0.73	22.85	1.38	0.00	0.98	0.03	0.79	0.20
120_01.csv	8	20	0.76	24.94	1.36	0.00	1.10	0.02	1.03	0.26
120_02.csv	9	20.1	0.78	23.72	1.25	0.00	0.70	0.02	0.78	0.18
120_03.csv	8	16.1	0.75	19.34	1.38	0.00	0.94	0.02	1.04	0.26
120_04.csv	11	17.2	0.75	21.91	1.24	0.00	0.75	0.03	0.58	0.22
120_05.csv	8	15.7	0.79	23.67	1.25	0.00	0.64	0.02	0.51	0.17
120_06.csv	17	18.3	0.73	233.10	1.16	0.00	0.64	0.02	0.65	0.17
120_07.csv	13	21.3	0.68	24.80	1.18	0.00	0.78	0.03	0.71	0.14
120_08.csv	2	18	0.93	26.49	1.05	0.00	0.89	0.04	0.75	0.20
120_09.csv	12	16.4	0.81	19.81	1.09	0.00	1.43	0.02	1.14	0.16
120_10.csv	8	21.5	0.82	29.96	1.32	0.00	0.81	0.01	0.82	0.13
120_11.csv	9	18.6	0.78	127.59	1.35	0.00	0.92	0.04	0.54	0.18
120_12.csv	13	19.5	0.79	24.33	1.24	0.00	1.69	0.02	1.47	0.14
120_13.csv	15	17.3	0.78	21.87	1.32	0.00	0.64	0.03	0.63	0.16
120_14.csv	15	17.1	0.79	126.24	1.25	0.00	0.94	0.04	0.90	0.13
120_15.csv	9	18.5	0.83	25.08	1.21	0.00	0.67	0.03	0.74	0.19
120_16.csv	19	14.1	0.78	127.52	1.22	0.00	0.70	0.03	0.70	0.23
120_17.csv	7	20.8	0.85	26.56	1.08	0.00	0.87	0.03	0.54	0.18
120_18.csv	8	18.7	0.87	27.13	1.06	0.00	0.90	0.02	0.96	0.16
120_19.csv	13	16.7	0.84	127.59	1.14	0.00	0.64	0.01	0.59	0.17

Tabla 49. Resultados FIFO + GRASP 120 productos

FIFO + GRASP 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_0 0.csv	20	45.8	0.75	265.12	1.64	0.00	1.47	0.01	1.23	0.19
250_0 1.csv	10	43.5	0.75	52.16	1.53	0.00	1.63	0.02	1.42	0.16
250_0 2.csv	28	43.6	0.70	259.37	1.35	0.00	2.29	0.02	1.82	0.17
250_0 3.csv	13	46.5	0.73	161.43	1.56	0.00	1.28	0.01	1.27	0.21
250_0 4.csv	10	41.1	0.77	53.03	1.76	0.00	1.70	0.02	1.56	0.26
250_0 5.csv	11	45.3	0.72	153.68	1.46	0.00	1.35	0.02	1.08	0.27
250_0 6.csv	7	45.8	0.68	155.46	1.37	0.00	1.40	0.01	1.23	0.22
250_0 7.csv	6	46.8	0.70	51.98	1.29	0.00	1.17	0.02	1.28	0.20
250_0 8.csv	6	50.5	0.69	159.80	1.60	0.00	1.57	0.02	1.43	0.25
250_0 9.csv	5	43.6	0.76	49.99	1.58	0.00	1.36	0.02	1.14	0.17
250_1 0.csv	1	48.1	0.69	53.78	1.69	0.00	2.21	0.02	2.14	0.20
250_1 1.csv	7	45.2	0.73	50.01	1.63	0.00	1.47	0.03	1.57	0.18
250_1 2.csv	0	51.6	0.66	56.63	1.75	0.00	1.62	0.02	1.59	0.25
250_1 3.csv	9	53.7	0.68	165.71	1.63	0.00	1.55	0.02	1.43	0.16
250_1 4.csv	13	45.2	0.72	260.17	1.66	0.00	1.78	0.02	1.82	0.18
250_1 5.csv	1	45.5	0.67	51.33	1.57	0.00	1.12	0.02	1.04	0.14
250_1 6.csv	3	49.5	0.76	56.53	1.56	0.00	2.42	0.02	2.50	0.16
250_1 7.csv	4	51.1	0.68	55.78	1.58	0.00	1.70	0.01	1.67	0.27
250_1 8.csv	5	46.3	0.74	52.47	1.60	0.00	1.19	0.02	1.03	0.17
250_1 9.csv	0	50	0.69	58.37	1.86	0.00	1.18	0.02	1.21	0.17

Tabla 50. Resultados FIFO + GRASP 250 productos

FIFO + GRASP 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	18	97.2	0.67	320.62	5.38	0.00	1.99	0.01	2.12	1.02
500_01.csv	40	95.3	0.65	640.91	4.45	0.00	1.42	0.01	1.24	0.91
500_02.csv	6	95.2	0.62	103.81	4.72	0.00	2.23	0.01	2.16	0.83
500_03.csv	31	90.2	0.72	530.92	5.03	0.00	1.94	0.01	1.81	0.66
500_04.csv	14	88.9	0.60	206.17	5.07	0.00	1.45	0.01	1.23	0.55
500_05.csv	14	101.6	0.63	225.85	4.12	0.00	1.56	0.01	1.32	0.73
500_06.csv	23	103.1	0.65	428.08	4.60	0.00	2.21	0.01	2.05	1.10
500_07.csv	41	78.7	0.65	619.79	4.22	0.00	1.62	0.01	1.54	1.09
500_08.csv	19.6	108.8	0.65	342.83	4.15	1.80	2.09	0.01	31.75	0.74
500_09.csv	26	103.6	0.63	538.76	4.23	0.00	2.33	0.02	2.05	0.72
500_10.csv	25	99.8	0.68	536.94	4.11	0.00	2.36	0.01	2.53	0.60
500_11.csv	41	90.4	0.64	530.10	4.53	0.00	1.96	0.01	1.92	0.85
500_12.csv	5	94.4	0.65	208.75	4.93	0.00	2.15	0.01	1.86	0.87
500_13.csv	42	89.3	0.63	632.15	5.09	0.00	1.79	0.02	1.95	0.83
500_14.csv	25	89.5	0.66	421.77	3.83	0.00	2.20	0.02	2.22	0.70
500_15.csv	0	99.9	0.65	110.60	4.46	0.00	2.26	0.01	2.27	0.89
500_16.csv	27	91.5	0.62	426.02	4.15	0.00	2.58	0.01	2.48	0.83
500_17.csv	23	91.8	0.62	423.17	4.68	0.00	2.32	0.01	2.46	0.71
500_18.csv	47	87.8	0.59	642.29	4.96	0.00	2.40	0.01	2.15	0.96
500_19.csv	11	94.2	0.69	212.92	3.85	0.00	2.71	0.01	2.63	0.60

Tabla 51. Resultados FIFO + GRASP 500 productos

FIFO + GRASP 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	56	204.1	0.60	869.08	14.34	0.00	3.70	0.01	3.51	1.92
1000_01.csv	62	198.6	0.58	1168.03	12.59	0.00	3.01	0.00	3.10	1.52
1000_02.csv	47	214.8	0.60	965.71	12.54	0.00	3.82	0.01	3.51	1.77
1000_03.csv	33	193.8	0.60	635.26	14.14	0.00	2.82	0.01	2.70	1.53
1000_04.csv	38	217.6	0.62	764.07	12.84	0.00	2.15	0.01	2.17	1.86
1000_05.csv	54	207.8	0.63	671.65	14.35	0.00	2.68	0.01	2.58	2.75
1000_06.csv	69	206.5	0.59	1093.00	13.72	0.00	3.23	0.01	3.24	2.32
1000_07.csv	47	212.5	0.61	760.68	14.73	0.00	2.84	0.01	2.93	1.61
1000_08.csv	39	187.2	0.62	942.68	13.89	0.00	4.28	0.01	4.14	1.45
1000_09.csv	79	201.4	0.60	1288.58	13.09	0.00	3.88	0.01	3.91	1.96
1000_10.csv	55	211.7	0.62	875.61	12.93	0.00	3.66	0.01	3.48	1.67
1000_11.csv	53	209.2	0.59	969.16	13.41	0.00	4.53	0.01	4.43	1.84
1000_12.csv	46	205	0.62	1061.16	13.33	0.00	3.97	0.01	3.99	2.00
1000_13.csv	36	218.1	0.62	767.76	12.21	0.00	2.98	0.01	2.60	0.91
1000_14.csv	28	222.9	0.60	655.12	12.66	0.00	2.95	0.01	2.88	1.95
1000_15.csv	41	205.2	0.61	856.60	13.75	0.00	1.66	0.01	1.65	2.25
1000_16.csv	30	205.3	0.62	751.79	13.62	0.00	3.10	0.01	3.23	2.20
1000_17.csv	54	205	0.62	866.80	13.80	0.00	3.35	0.01	3.06	2.44
1000_18.csv	69	201.3	0.60	1180.38	13.28	0.00	1.62	0.01	1.44	1.99
1000_19.csv	24	210.7	0.61	545.24	13.86	0.00	3.80	0.01	3.84	1.59

Tabla 52. Resultados FIFO + GRASP 1000 productos

GREEDY + GRASP 120 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	8	23.3	0.70	29.00	1.56	0.00	0.78	0.02	0.74	0.34
120_01.csv	8	28.1	0.74	33.09	1.61	0.00	0.83	0.02	0.78	0.19
120_02.csv	9	26.9	0.78	30.54	1.62	0.00	0.83	0.02	0.73	0.21
120_03.csv	8	23.9	0.74	27.01	1.59	0.00	0.83	0.03	0.87	0.24
120_04.csv	11	22.1	0.74	26.70	1.64	0.00	0.83	0.02	0.63	0.37
120_05.csv	8	21.3	0.78	28.90	1.56	0.00	1.10	0.02	1.01	0.23
120_06.csv	17	25.3	0.75	240.17	1.42	0.00	1.00	0.03	0.85	0.22
120_07.csv	13	28.2	0.68	31.63	1.45	0.00	0.60	0.02	0.59	0.18
120_08.csv	2	26.9	0.85	34.40	1.73	0.00	1.04	0.02	0.82	0.23
120_09.csv	12	25.2	0.74	28.17	1.50	0.00	0.87	0.02	0.84	0.32
120_10.csv	8	29	0.79	37.39	1.65	0.00	0.63	0.02	0.56	0.18
120_11.csv	9	25.5	0.76	134.10	1.65	0.00	1.20	0.01	1.04	0.21
120_12.csv	13	29	0.75	33.29	1.40	0.00	1.34	0.03	1.07	0.25
120_13.csv	15	26	0.75	30.08	1.40	0.00	0.77	0.03	0.61	0.19
120_14.csv	15	24.5	0.81	133.66	1.28	0.00	0.81	0.03	0.67	0.24
120_15.csv	9	26.2	0.81	32.79	1.37	0.00	0.75	0.01	0.68	0.20
120_16.csv	19	21.7	0.72	134.29	1.56	0.00	1.19	0.04	0.85	0.30
120_17.csv	7	29.2	0.81	34.53	1.68	0.00	1.08	0.02	0.99	0.22
120_18.csv	8	27.3	0.81	34.77	1.61	0.00	0.64	0.02	0.56	0.15
120_19.csv	13	24.5	0.79	135.38	1.59	0.00	0.50	0.02	0.31	0.22

Tabla 53. Resultados Greedy + GRASP 120 productos

GREEDY + GRASP 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_00.csv	20	64.8	0.72	284.25	2.53	0.00	1.54	0.02	1.44	0.53
250_01.csv	10	61.9	0.74	69.74	2.39	0.00	1.14	0.02	1.00	0.21
250_02.csv	28	62.9	0.71	278.07	2.16	0.00	1.37	0.02	1.16	0.36
250_03.csv	13	63	0.73	176.58	2.46	0.00	1.26	0.01	1.12	0.41
250_04.csv	10	60	0.73	70.74	2.54	0.00	0.45	0.02	0.55	0.53
250_05.csv	11	67.2	0.72	175.55	2.41	0.00	1.17	0.02	1.02	0.33
250_06.csv	7	68.1	0.68	176.47	2.49	0.00	1.14	0.01	1.13	0.47
250_07.csv	6	69.9	0.71	73.93	2.42	0.00	2.17	0.02	2.01	0.48
250_08.csv	6	73.8	0.70	182.89	2.62	0.00	1.33	0.01	1.05	0.64
250_09.csv	5	68.7	0.75	74.09	2.44	0.00	1.35	0.02	1.30	0.40
250_10.csv	1	70.8	0.71	76.11	2.58	0.00	1.83	0.01	1.56	0.34
250_11.csv	7	66.8	0.72	70.66	2.50	0.00	1.72	0.02	1.60	0.55
250_12.csv	0	71.4	0.69	76.51	2.53	0.00	1.50	0.01	1.44	0.31
250_13.csv	9	76.4	0.70	188.38	2.46	0.00	1.02	0.01	1.01	0.40
250_14.csv	13	67.5	0.72	282.21	2.32	0.00	1.12	0.01	0.96	0.53
250_15.csv	1	67.3	0.68	73.09	2.50	0.00	1.19	0.01	1.15	0.53
250_16.csv	3	68.3	0.76	75.24	2.72	0.00	1.27	0.01	1.07	0.57
250_17.csv	4	70.4	0.70	75.08	2.82	0.00	1.69	0.01	1.53	0.46
250_18.csv	5	71	0.75	76.89	2.50	0.00	1.67	0.01	1.49	0.52
250_19.csv	0	67.3	0.70	74.32	2.86	0.00	1.73	0.01	1.50	0.36

Tabla 54. Resultados Greedy + GRASP 250 productos

GREEDY + GRASP 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	18	136.1	0.69	359.48	6.21	0.00	2.17	0.01	2.14	0.91
500_01.csv	40	131.6	0.65	676.01	5.26	0.00	0.92	0.01	1.07	0.52
500_02.csv	6	138.9	0.65	147.20	6.24	0.00	2.43	0.01	2.30	0.65
500_03.csv	31	132.3	0.71	572.46	5.89	0.00	0.90	0.01	0.67	0.65
500_04.csv	14	131.6	0.62	248.85	6.03	0.00	1.91	0.01	1.88	1.01
500_05.csv	14	141.2	0.64	264.66	6.89	0.00	1.54	0.01	1.28	1.32
500_06.csv	23	149.9	0.66	473.78	6.26	0.00	2.07	0.01	1.56	0.88
500_07.csv	41	115.4	0.66	654.52	6.47	0.00	1.96	0.01	1.85	1.57
500_08.csv	19	146	0.66	369.01	6.53	0.00	0.45	0.01	0.59	0.86
500_09.csv	26	142.7	0.65	577.27	6.75	0.00	1.68	0.01	1.36	0.82
500_10.csv	25	139.3	0.69	575.67	6.76	0.00	2.19	0.01	1.91	1.00
500_11.csv	41	128.2	0.66	567.81	6.22	0.00	1.78	0.01	1.68	0.81
500_12.csv	5	141.3	0.65	254.54	6.81	0.00	1.49	0.01	1.50	0.69
500_13.csv	42	124.8	0.66	667.50	5.63	0.00	1.66	0.02	1.40	0.75
500_14.csv	25	132.1	0.68	464.11	6.41	0.00	1.14	0.01	0.97	1.21
500_15.csv	0	137.3	0.66	146.62	6.53	0.00	1.95	0.01	1.84	0.67
500_16.csv	27	131.8	0.63	465.21	5.84	0.00	1.54	0.01	1.36	0.68
500_17.csv	23	123.5	0.64	454.84	5.55	0.00	1.28	0.01	1.14	0.71
500_18.csv	47	124.6	0.62	678.63	5.86	0.00	1.36	0.01	1.41	0.81
500_19.csv	11	136.9	0.70	254.98	5.63	0.00	1.81	0.01	1.34	1.16

Tabla 55. Resultados Greedy + GRASP 500 productos

GREEDY + GRASP 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	69	275.8	0.63	1254.54	17.41	0.00	2.89	0.01	2.12	0.47
1000_01.csv	47	299.8	0.62	846.60	16.82	0.00	3.09	0.01	2.94	2.13
1000_02.csv	62	280.3	0.60	1248.75	16.83	0.00	3.20	0.01	3.07	2.00
1000_03.csv	33	279.4	0.62	720.41	17.56	0.00	2.69	0.01	2.50	2.34
1000_04.csv	47	309.7	0.62	1060.04	17.14	0.00	1.79	0.00	1.84	1.19
1000_05.csv	54	289.9	0.65	951.15	17.58	0.00	2.70	0.01	2.57	2.77
1000_06.csv	46	283.2	0.64	1139.62	17.14	0.00	2.60	0.00	2.53	1.85
1000_07.csv	79	280.2	0.62	1367.24	16.54	0.00	1.94	0.01	1.87	1.82
1000_08.csv	55	298	0.64	960.64	16.47	0.00	1.95	0.01	2.05	2.07
1000_09.csv	24	288.9	0.63	621.87	20.22	0.00	1.45	0.01	1.30	2.21
1000_10.csv	54	298.5	0.65	762.34	16.77	0.00	2.84	0.01	2.52	1.89
1000_11.csv	69	280.1	0.61	1164.98	18.72	0.00	2.98	0.01	2.67	1.78
1000_12.csv	38	294.6	0.64	839.76	16.73	0.00	2.06	0.01	2.12	1.29
1000_13.csv	30	287.4	0.64	833.40	17.55	0.00	2.80	0.01	2.64	1.37
1000_14.csv	39	266	0.63	1018.89	17.04	0.00	2.45	0.01	2.38	1.74
1000_15.csv	41	289.5	0.63	940.54	17.15	0.00	2.84	0.01	2.86	1.87
1000_16.csv	56	277.2	0.64	940.06	18.47	0.00	1.72	0.01	1.26	2.43
1000_17.csv	36	297.8	0.64	846.04	18.39	0.00	3.03	0.01	2.63	2.23
1000_18.csv	28	318.9	0.63	750.41	17.10	0.00	2.88	0.01	2.87	1.75
1000_19.csv	53	294.6	0.62	1053.87	17.38	0.00	1.69	0.01	1.56	1.93

Tabla 56. Resultados Greedy + GRASP 1000 productos

DISTANCIAS + GRASP 120 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
120_00.csv	8	25	0.60	30.15	2.39	0.00	0.00	0.02	0.04	0.34
120_01.csv	8	27	0.66	32.38	2.22	0.00	0.00	0.02	0.11	0.33
120_02.csv	9	28	0.69	32.31	2.35	0.00	0.00	0.02	0.09	0.27
120_03.csv	8	25	0.64	27.82	2.13	0.00	0.00	0.01	0.11	0.34
120_04.csv	11	26	0.63	29.38	2.08	0.00	0.00	0.01	0.00	0.31
120_05.csv	8	23	0.74	30.46	2.20	0.00	0.00	0.01	0.10	0.31
120_06.csv	17	28	0.62	241.30	1.94	0.00	0.00	0.02	0.08	0.35
120_07.csv	13	30	0.57	33.30	1.81	0.00	0.00	0.02	0.04	0.35
120_08.csv	2	27	0.73	33.97	2.20	0.00	0.00	0.01	0.14	0.28
120_09.csv	12	26	0.68	28.13	2.03	0.00	0.00	0.01	0.07	0.32
120_10.csv	8	26	0.68	33.43	2.14	0.00	0.00	0.01	0.08	0.31
120_11.csv	9	26.5	0.65	134.57	1.89	0.00	0.50	0.02	0.08	0.25
120_12.csv	13	28	0.64	32.18	1.85	0.00	0.00	0.02	0.11	0.31
120_13.csv	15	26	0.66	29.24	1.99	0.00	0.00	0.01	0.02	0.19
120_14.csv	15	30	0.67	138.16	1.97	0.00	0.00	0.01	0.06	0.29
120_15.csv	9	25	0.73	31.55	1.92	0.00	0.00	0.01	0.03	0.26
120_16.csv	19	24	0.63	134.70	1.69	0.00	0.00	0.01	0.06	0.19
120_17.csv	7	25	0.72	30.22	2.03	0.00	0.00	0.01	0.01	0.32
120_18.csv	8	31	0.66	36.25	2.04	0.00	0.00	0.01	0.04	0.41
120_19.csv	13	26.2	0.67	135.00	1.85	0.00	0.40	0.02	0.42	0.26

Tabla 57. Resultados Distancias + GRASP 120 productos

DISTANCIAS + GRASP 250 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
250_00.csv	15	61	0.61	174.96	4.82	0.00	0.00	0.01	0.17	0.71
250_01.csv	10	58	0.62	64.72	4.34	0.00	0.00	0.01	0.13	0.49
250_02.csv	28	53	0.58	266.87	4.10	0.00	0.00	0.01	0.12	0.62
250_03.csv	13	59	0.59	171.18	4.21	0.00	0.00	0.01	0.06	0.57
250_04.csv	10	62	0.60	71.82	4.88	0.00	0.00	0.01	0.12	0.65
250_05.csv	11	62.1	0.58	170.61	3.91	0.00	0.30	0.01	0.28	0.71
250_06.csv	7	66	0.54	174.58	4.37	0.00	0.00	0.01	0.14	0.59
250_07.csv	6	62	0.54	65.29	4.35	0.00	0.00	0.01	0.07	0.63
250_08.csv	6	67.1	0.54	175.25	4.42	0.00	0.30	0.01	0.26	0.70
250_09.csv	5	55	0.62	59.92	3.65	0.00	0.00	0.02	0.05	0.63
250_10.csv	1	62	0.55	65.12	4.17	0.00	0.00	0.01	0.18	0.47
250_11.csv	7	58.3	0.56	61.05	4.33	0.00	0.46	0.01	0.40	0.63
250_12.csv	0	63.3	0.56	68.06	4.91	0.00	0.46	0.01	0.39	0.64
250_13.csv	9	60	0.50	171.31	4.93	0.00	0.00	0.01	0.17	0.75
250_14.csv	8	58	0.55	166.40	4.14	0.00	0.00	0.01	0.12	0.32
250_15.csv	1	61	0.51	65.43	4.42	0.00	0.00	0.01	0.11	0.53
250_16.csv	3	60	0.59	64.99	4.53	0.00	0.00	0.01	0.07	0.76
250_17.csv	4	66.3	0.54	70.26	4.38	0.00	0.46	0.01	0.45	0.74
250_18.csv	5	64	0.56	68.44	4.43	0.00	0.00	0.01	0.12	0.55
250_19.csv	0	64	0.54	69.69	4.83	0.00	0.00	0.01	0.08	0.68

Tabla 58. Resultados Distancias + GRASP 250 productos

DISTANCIAS + GRASP 500 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
500_00.csv	18	119	0.52	341.23	10.35	0.00	0.00	0.01	0.22	0.21
500_01.csv	34	125	0.50	562.77	9.65	0.00	0.00	0.01	0.08	0.25
500_02.csv	6	121.5	0.51	128.33	10.63	0.00	0.67	0.01	0.61	0.24
500_03.csv	31	125	0.52	562.83	11.43	0.00	0.00	0.00	0.11	0.82
500_04.csv	14	123	0.49	236.84	10.17	0.00	0.00	0.01	0.08	0.35
500_05.csv	14	117	0.46	238.87	10.57	0.00	0.00	0.01	0.05	0.28
500_06.csv	18	126	0.47	343.91	10.62	0.00	0.00	0.01	0.17	0.23
500_07.csv	23	116.1	0.51	336.96	9.59	0.00	0.30	0.01	0.31	0.29
500_08.csv	19	124.5	0.47	347.01	10.50	0.00	0.50	0.01	0.55	0.19
500_09.csv	26	131.1	0.49	564.19	9.52	0.00	0.30	0.00	0.37	0.27
500_10.csv	19	123	0.53	453.54	9.83	0.00	0.00	0.01	0.15	0.25
500_11.csv	41	118.4	0.51	556.63	9.28	0.00	0.66	0.01	0.60	0.25
500_12.csv	5	124.7	0.48	237.07	10.45	0.00	0.46	0.01	0.49	0.16
500_13.csv	42	124	0.49	664.83	9.35	0.00	0.00	0.01	0.17	0.20
500_14.csv	15	117.1	0.49	237.55	9.84	0.00	0.30	0.00	0.35	0.16
500_15.csv	0	131.1	0.49	140.40	10.08	0.00	0.30	0.01	0.28	0.25
500_16.csv	27	119.1	0.49	451.16	9.53	0.00	0.30	0.01	0.33	0.83
500_17.csv	23	120.1	0.48	449.00	9.79	0.00	0.30	0.01	0.34	0.23
500_18.csv	47	115	0.48	668.56	9.93	0.00	0.00	0.00	0.19	0.24
500_19.csv	11	113	0.51	230.74	10.28	0.00	0.00	0.01	0.20	0.15

Tabla 59. Resultados Distancias + GRASP 500 productos

DISTANCIAS + GRASP 1000 productos:

instancia	num singles	num bundles	mean max distances	sum bundling cost	total time	std num singles	std num bundles	std mean max distances	std sum bundling cost	std total time
1000_00.csv	51	235.4	0.40	792.07	30.52	0.00	0.49	0.01	0.37	0.98
1000_01.csv	57	245	0.43	1108.11	28.50	0.00	0.00	0.01	0.24	0.95
1000_02.csv	47	243.7	0.42	990.98	31.53	0.00	0.46	0.00	0.39	2.67
1000_03.csv	33	243.4	0.41	679.45	30.42	0.00	0.66	0.01	0.61	0.80
1000_04.csv	33	254.1	0.42	692.35	31.15	0.00	0.30	0.01	0.31	0.85
1000_05.csv	49	250.9	0.45	607.56	30.26	0.00	0.70	0.00	0.64	0.98
1000_06.csv	54	250.5	0.40	818.23	30.56	0.00	1.57	0.01	1.57	0.87
1000_07.csv	47	247.7	0.43	793.76	29.79	0.00	0.64	0.00	0.72	0.79
1000_08.csv	33	234.4	0.45	877.28	29.23	0.00	0.49	0.00	0.47	0.78
1000_09.csv	69	237.7	0.42	1112.33	29.55	0.00	0.64	0.00	0.70	1.08
1000_10.csv	49	244.4	0.43	799.07	30.80	0.00	0.66	0.01	0.62	2.18
1000_11.csv	43	252.8	0.41	801.19	31.06	0.00	0.98	0.01	0.99	2.55
1000_12.csv	36	247.5	0.45	890.67	29.67	0.00	0.67	0.01	0.64	0.81
1000_13.csv	25	251.3	0.45	586.38	31.11	0.00	0.46	0.01	0.48	0.86
1000_14.csv	13	264	0.43	379.27	30.63	0.00	0.63	0.00	0.74	0.70
1000_15.csv	31	236.9	0.43	674.46	30.45	0.00	0.70	0.01	0.67	0.96
1000_16.csv	30	250	0.44	790.43	30.67	0.00	0.00	0.01	0.43	0.92
1000_17.csv	48	235.5	0.44	787.29	30.68	0.00	0.50	0.01	0.53	1.15
1000_18.csv	59	241	0.43	1007.19	31.16	0.00	0.45	0.00	0.47	2.30
1000_19.csv	18	245.2	0.44	470.45	32.33	0.00	1.08	0.01	1.16	2.97

Tabla 60. Resultados Distancias + GRASP 1000 productos

Bibliografía

Adler, M., Gibbons, P. B. & Matias, Y., 2002. Scheduling space-sharing for internet advertising. *Journal of Scheduling*, 5(2), pp. 103-119.

Alvarez-Valdés, R., Parreño, F. & Tamarit, J. M., 2013. A grasp/path relinking algorithm for two- and three-dimensional multiple bin-size bin packing problems. *Computers & Operations Research*, 40(12), pp. 3081-3090.

Alvisi, S. & Franchini, M., 2014. A heuristic procedure for the automatic creation of district metered areas in water distribution systems. *Urban Water Journal*, 11(2), pp. 137-159.

An, X., Xu, S., Chen, J. & Zhang, Y., 2013. Distributed risk aversion parameter estimation for first-price auction in sensor networks. *International Journal of Distributed Sensor Networks*, Volumen 2013.

Asker, J., 2010. A study of the internal organization of a bidding cartel. *The American Economic Review*, 100(3), pp. 724-762.

Ausubel, L. M., 2004. An efficient ascending-bid auction for multiple objects. *American Economic Review*, pp. 1452-1475.

Avery, C. & Hendershott, T., 2000. Bundling and optimal auctions of multiple products. *The Review of Economic Studies*, 67(3), pp. 483-497.

Azar, Y. y otros, 2000. *Fair versus unrestricted bin packing*. s.l., s.n., pp. 200-213.

Baewicz, J. & Ecker, K., 1983. A linear time algorithm for restricted bin packing and scheduling problems. *Operations Research Letters*, 2(2), pp. 80-83.

Bai, R. y otros, 2007. A simulated annealing hyper-heuristic methodology for flexible decision support. *School of CSIT, University of Nottingham, UK, Tech. Rep.*

Bajari, P. & Hortacsu, A., 2003. The winner's curse, reserve prices, and endogenous entry: Empirical insights from eBay auctions. *RAND Journal of Economics*, pp. 329-355.

Baker, B. S. & Coffman, J. E. G., 1981. A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM Journal on Algebraic Discrete Methods*, 2(2), pp. 147-152.

- Bakos, Y. & Brynjolfsson, E., 2000. Bundling and Competition on the Internet. *Marketing science*, 19(1), pp. 63-82.
- Baldi, M. M. & Bruglieri, M., 2016. On the generalized bin packing problem. *International Transactions in Operational Research*.
- Bansal, N., Liu, Z. & Sankar, A., 2009. Bin-packing with fragile objects and frequency allocation in cellular networks. *Wireless Networks*, 15(6), pp. 821-830.
- Bapna, R., Goes, P. & Gupta, A., 2000. A theoretical and empirical investigation of multi-item on-line auctions. *Information Technology and Management*, 1(1-2), pp. 1-23.
- Ba, S. & Pavlou, P. A., 2002. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS quarterly*, pp. 243-268.
- Beaumont, O., Bonichon, N., Larcheveque, H. & Sud-Ouest, I. N. R. I. A. B., 2010. *Bin packing under distance constraint*, s.l.: s.n.
- Bell, D. E., 1983. Risk premiums for decision regret. *Management Science*, 29(10), pp. 1156-1166.
- Bell, D. E., 1985. Disappointment in decision making under uncertainty. *Operations research*, 33(1), pp. 1-27.
- Belov, G. & Scheithauer, G., 2007. Setup and open-stacks minimization in one-dimensional stock cutting. *INFORMS Journal on Computing*, 19(1), pp. 27-35.
- Bezdek, J. C., 2013. *Pattern recognition with fuzzy objective function algorithms*. s.l.:Springer Science & Business Media.
- Bilo, V., 2006. *On the packing of selfish items*. s.l., s.n., pp. 9--pp.
- Boykov, Y. & Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9), pp. 1124-1137.
- Brélaz, D., 1979. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4), pp. 251-256.
- Brillout, R., 2009. *A Multi-Level Framework for Bisection Heuristics*, s.l.: s.n.

- Bulow, J. & Klemperer, P., 2002. Prices and the Winner's Curse. *RAND journal of Economics*, pp. 1-21.
- Busygin, S., 2006. A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154(15), pp. 2080-2096.
- Cason, T. N., 1995. An experimental investigation of the seller incentives in the EPA's emission trading auction. *The American Economic Review*, pp. 905-922.
- Chakraborty, I., 1999. Bundling decisions for selling multiple objects. *Economic Theory*, 13(3), pp. 723-733.
- Chandra, A. K., Hirschberg, D. S. & Wong, C. K., 1978. Bin packing with geometric constraints in computer network design. *Operations Research*, 26(5), pp. 760-772.
- Chan, W.-T. y otros, 2007. Online bin packing of fragile objects with application in cellular networks. *Journal of Combinatorial Optimization*, 14(4), pp. 427-435.
- Chen, Z. & Ji, H., 2010. *Graph-based clustering for computational linguistics: A survey*. s.l., s.n., pp. 1-9.
- Cherng, J.-S. & Lo, M.-J., 2001. *A hypergraph based clustering algorithm for spatial data sets*. s.l., s.n., pp. 83-90.
- Che, Y.-K. & Gale, I., 1996. Expected revenue of all-pay auctions and first-price sealed-bid auctions with budget constraints. *Economics Letters*, 50(3), pp. 373-379.
- Chow, Y. L. & Ooi, J. T. L., 2014. First-Price Sealed-Bid Tender versus English Open Auction: Evidence from Land Auctions. *Real Estate Economics*, 42(2), pp. 253-278.
- Coffman Jr, E. G. y otros, 2013. Bin packing approximation algorithms: survey and classification. En: *Handbook of Combinatorial Optimization*. s.l.:Springer, pp. 455-531.
- Coffman Jr, E. G., Garey, M. R. & Johnson, D. S., 1984. Approximation algorithms for bin-packing—an updated survey. En: *Algorithm design for computer system design*. s.l.:Springer, pp. 49-106.
- Coffman, E. G., Garey, M. R. & Johnson, D. S., 1987. Bin packing with divisible item sizes. *Journal of Complexity*, 3(4), pp. 406-428.

- Coffman, J. E. G., Garey, M. R. & Johnson, D. S., 1983. Dynamic bin packing. *SIAM Journal on Computing*, 12(2), pp. 227-258.
- Cramton, P., 1998. Ascending auctions. *European Economic Review*, 42(3), pp. 745-756.
- Csirik, J. & Woeginger, G. J., 1998. On-line packing and covering problems. En: *Online Algorithms*. s.l.:Springer, pp. 147-177.
- Cui, X. & Lai, V. S., 2013. Bidding strategies in online single-unit auctions: Their impact and satisfaction. *Information & Management*, 50(6), pp. 314-321.
- Dahmani, N., Krichen, S. & Ghazouani, D., 2015. A variable neighborhood descent approach for the two-dimensional bin packing problem. *Electronic Notes in Discrete Mathematics*, Volumen 47, pp. 117-124.
- De La Vega, W. F. & Lueker, G. S., 1981. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1(4), pp. 349-355.
- Dellarocas, C., 2000. The design of reliable trust management systems for electronic trading communities. URL:< citeseer.ist.psu.edu/380109.html.
- Dewan, S. & Hsu, V., 2001. *Trust in electronic markets: Price discovery in generalist versus specialty online auctions*, s.l.: s.n.
- Dósa, G. & He, Y., 2006. Bin packing problems with rejection penalties and their dual problems. *Information and Computation*, 204(5), pp. 795-815.
- Duran, B. S. & Odell, P. L., 2013. *Cluster analysis: a survey*. s.l.:Springer Science & Business Media.
- Elhedhli, S., Li, L., Gzara, M. & Naoum-Sawaya, J., 2011. A branch-and-price algorithm for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 23(3), pp. 404-415.
- Elmaghraby, W., 2007. Auctions within e-sourcing events. *Production and Operations Management*, 16(4), pp. 409-422.
- Engelbrecht-Wiggans, R., 1989. The effect of regret on optimal bidding in auctions. *Management Science*, 35(6), pp. 685-692.

- Engelbrecht-Wiggans, R. & Katok, E., 2008. Regret and feedback information in first-price sealed-bid auctions. *Management Science*, 54(4), pp. 808-819.
- Engelbrecht-Wiggans, R. & Katok, E., 2009. A direct test of risk aversion and regret in first price sealed-bid auctions. *Decision Analysis*, 6(2), pp. 75-86.
- Epstein, L., Favrholt, L. M. & Levin, A., 2011. Online variable-sized bin packing with conflicts. *Discrete Optimization*, 8(2), pp. 333-343.
- Epstein, L. & Levin, A., 2008. On bin packing with conflicts. *SIAM Journal on Optimization*, 19(3), pp. 1270-1298.
- Epstein, L. & Van Stee, R., 2007. *Approximation schemes for packing splittable items with cardinality constraints*. s.l., s.n., pp. 232-245.
- Falkenauer, E., 1996. A hybrid grouping genetic algorithm for bin packing. *Journal of heuristics*, 2(1), pp. 5-30.
- Falkenauer, E. & Delchambre, A., 1992. *A genetic algorithm for bin packing and line balancing*. s.l., s.n., pp. 1186-1192.
- Feng, J. & Chatterjee, K., 2008. Simultaneous vs. Sequential Sales, Intensity of Competition and Uncertainty. *Sequential Sales, Intensity of Competition and Uncertainty (July 13, 2008)*.
- Fiedler, M., 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4), pp. 619-633.
- Fisher, D. C., 1988. Next-fit packs a list and its reverse into the same number of bins. *Operations research letters*, 7(6), pp. 291-293.
- Fleszar, K. & Charalambous, C., 2011. Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research*, 210(2), pp. 176-184.
- Friesen, D. K. & Langston, M. A., 1991. Analysis of a compound bin packing algorithm. *SIAM Journal on Discrete Mathematics*, 4(1), pp. 61-79.
- Gabrielli, M. F., 2014. Econometrics of first price auctions: a survey of the theoretical and applied literature. *Económica*, Volumen 60.

- Galambos, G., 1986. Parametric lower bound for on-line bin-packing. *SIAM Journal on Algebraic Discrete Methods*, 7(3), pp. 362-367.
- Galambos, G. & Woeginger, G. J., 1993. Repacking helps in bounded space on-line bin-packing. *Computing*, 49(4), pp. 329-338.
- Garey, M. R., Graham, R. L., Johnson, D. S. & Yao, A. C.-C., 1976. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory, Series A*, 21(3), pp. 257-298.
- Garey, M. R. & Johnson, D. S., 1979. A Guide to the Theory of NP-Completeness. *WH Freeman, New York*.
- Geiger, M. J., 2008. Bin packing under multiple objectives-a heuristic approximation approach. *arXiv preprint arXiv:0809.0755*.
- Gendreau, M., Laporte, G. & Semet, F., 2004. Heuristics and lower bounds for the bin packing problem with conflicts. *Computers & Operations Research*, 31(3), pp. 347-358.
- Gentry, M., Li, T. & Lu, J., 2015. Existence of monotone equilibrium in first price auctions with private risk aversion and private initial wealth. *Games and Economic Behavior*, Volumen 94, pp. 214-221.
- Geva, A., 1999. Hierarchical unsupervised fuzzy clustering. *IEEE Trans. Fuzzy Syst.*
- Gorke, R., 2010. *An algorithmic walk from static to dynamic graph clustering*, s.l.: s.n.
- Graham, D. A. & Marshall, R. C., 1987. Collusive bidder behavior at single-object second-price and English auctions. *Journal of Political economy*, 95(6), pp. 1217-1239.
- Grove, E. F., 1995. *Online Bin Packing with Lookahead*. s.l., s.n., pp. 430-436.
- Guerre, E., Perrigne, I. & Vuong, Q., 2000. Optimal Nonparametric Estimation of First-Price Auctions. *Econometrica*, 68(3), pp. 525-574.
- Gupta, J. N. D. & Ho, J. C., 1999. A new heuristic algorithm for the one-dimensional bin-packing problem. *Production planning & control*, 10(6), pp. 598-603.
- Haile, P. A., Hong, H. & Shum, M., 2003. *Nonparametric tests for common values at first-price sealed-bid auctions*, s.l.: s.n.

- Hall, L. A., 1996. *Approximation algorithms for scheduling*. s.l., s.n., pp. 1-45.
- Hann, I.-H. & Terwiesch, C., 2003. Measuring the frictional costs of online transactions: The case of a name-your-own-price channel. *Management Science*, 49(11), pp. 1563-1579.
- Harrington, J. E., 2008. Detecting cartels. *Handbook of antitrust economics*, Volumen 213, p. 245.
- Harris, M. & Raviv, A., 1981. A theory of monopoly pricing schemes with demand uncertainty. *The American Economic Review*, 71(3), pp. 347-365.
- Hartuv, E. & Shamir, R., 2000. A clustering algorithm based on graph connectivity. *Information processing letters*, 76(4-6), pp. 175-181.
- Haruvy, E. y otros, 2008. Competition between auctions. *Marketing Letters*, 19(3-4), pp. 431-448.
- Hausch, D. B., 1986. Multi-object auctions: Sequential vs. simultaneous sales. *Management Science*, 32(12), pp. 1599-1610.
- Hendricks, K. & Porter, R. H., 1989. Collusion in auctions. *Annales d'Economie et de Statistique*, pp. 217-230.
- Hendrickson, B. & Leland, R., 1995. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing*, 16(2), pp. 452-469.
- Houser, D. & Wooders, J., 2006. Reputation in auctions: Theory, and evidence from eBay. *Journal of Economics & Management Strategy*, 15(2), pp. 353-369.
- Hu, A., Matthews, S. A. & Zou, L., 2010. Risk aversion and optimal reserve prices in first-and second-price auctions. *Journal of Economic Theory*, 145(3), pp. 1188-1202.
- Isaac, R. M. & James, D., 2000. Just who are you calling risk averse?. *Journal of Risk and Uncertainty*, 20(2), pp. 177-187.
- Jain, A. K. & Dubes, R. C., 1988. *Algorithms for clustering data*. s.l.:Prentice-Hall, Inc..
- Jansen, K., 1999. An approximation scheme for bin packing with conflicts. *Journal of combinatorial optimization*, 3(4), pp. 363-377.
- Johnson, D. S., 1973. *Near-optimal bin packing algorithms*, s.l.: s.n.

- Johnson, D. S., 1974. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences*, 9(3), pp. 256-278.
- Johnson, D. S., 1974. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3), pp. 272-314.
- Johnson, D. S. & Garey, M. R., 1985. A 7160 theorem for bin packing. *Journal of Complexity*, 1(1), pp. 65-106.
- Johnson, J., Reck, B. K., Wang, T. & Graedel, T. E., 2008. The energy benefit of stainless steel recycling. *Energy Policy*, 36(1), pp. 181-192.
- Kagel, J. H. & Levin, D., 1993. Independent private value auctions: Bidder behaviour in first-, second- and third-price auctions with varying numbers of bidders. *The Economic Journal*, 103(419), pp. 868-879.
- Kagel, J. H. & Richard, J.-F., 2001. Super-experienced bidders in first-price common-value auctions: Rules of thumb, Nash equilibrium bidding, and the winner's curse. *Review of Economics and Statistics*, 83(3), pp. 408-419.
- Kaiser, L. F. & Kaiser, M., 2010. *The official eBay guide to buying, selling, and collecting just about anything*. s.l.:Simon and Schuster.
- Kang, J. & Park, S., 2003. Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147(2), pp. 365-372.
- Kannan, P. K., Pope, B. K. & Jain, S., 2009. Practice prize winner-pricing digital content product lines: A model and application for the national academies press. *Marketing Science*, 28(4), pp. 620-636.
- Karmarkar, N. & Karp, R. M., 1982. *An efficient approximation scheme for the one-dimensional bin-packing problem*. s.l., s.n., pp. 312-320.
- Karypis, G., Han, E.-H. & Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8), pp. 68-75.
- Katona, G. Y., 1997. Edge disjoint polyp packing. *Discrete applied mathematics*, 78(1), pp. 133-152.

- Kauffman, R. J. & Wood, C. A., 2000. Running up the bid: Modeling seller opportunism in Internet auctions. *AMCIS 2000 Proceedings*, p. 376.
- Kaufman, L. & Rousseeuw, P. J., 2009. *Finding groups in data: an introduction to cluster analysis*. s.l.:John Wiley & Sons.
- Kawaji, H., Takenaka, Y. & Matsuda, H., 2004. Graph-based clustering for finding distant relationships in a large set of protein sequences. *Bioinformatics*, 20(2), pp. 243-252.
- Kellerer, H. & Pferschy, U., 1999. Cardinality constrained bin-packing problems. *Annals of Operations Research*, Volumen 92, pp. 335-348.
- Khanafer, A., Clautiaux, F. & Talbi, E.-G., 2012. Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts. *Computers & Operations Research*, 39(1), pp. 54-63.
- Klein, S. & O'Keefe, M., 1999. The impact of the web on auctions: some empirical evidence and theoretical considerations. *International Journal of Electronic Commerce*, 3(3), pp. 7-20.
- Krishna, V., 2009. *Auction theory*. s.l.:Academic press.
- Krishna, V. & Rosenthal, R. W., 1996. Simultaneous auctions with synergies. *Games and economic behavior*, 17(1), pp. 1-31.
- Laffont, J.-J., Ossard, H. & Vuong, Q., 1995. Econometrics of first-price auctions. *Econometrica: Journal of the Econometric Society*, pp. 953-980.
- Lenstra Jr, H. W., 1983. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4), pp. 538-548.
- Lin, S.-J., Jheng, Y.-Y. & Yu, C.-H., 2012. Combining ranking concept and social network analysis to detect collusive groups in online auctions. *Expert Systems with Applications*, 39(10), pp. 9079-9086.
- Liu, G., 1968. *Introduction to Combinatorial Mathematics*. s.l.:Mc Graw - Hill.
- Lizzeri, A. & Persico, N., 2000. Uniqueness and existence of equilibrium in auctions with a reserve price. *Games and Economic Behavior*, 30(1), pp. 83-114.

- Lucking-Reiley, D., 1999. Using field experiments to test equivalence between auction formats: Magic on the Internet. *American Economic Review*, pp. 1063-1080.
- Maiza, M. & Radjef, M. S., 2011. Heuristics for solving the bin-packing problem with conflicts. *Applied Mathematical Sciences*, 5(35), pp. 1739-1752.
- Maiza, M., Radjef, M. S. & Sais, L., 2016. Efficient Lower Bounds for Packing Problems in Heterogeneous Bins with Conflicts Constraint. En: *Intelligent Mathematics II: Applied Mathematics and Approximation Theory*. s.l.:Springer, pp. 263-270.
- Manyem, P., Salt, R. L. & Visser, M. S., 2003. Approximation lower bounds in online LIB bin packing and covering. *Journal of Automata, Languages and Combinatorics*, 8(4), pp. 663-674.
- Marechal, F. & Morand, P.-H., 2011. First-price sealed-bid auctions when bidders exhibit different attitudes toward risk. *Economics Letters*, 113(2), pp. 108-111.
- Mares, V. & Swinkels, J. M., 2014. On the analysis of asymmetric first price auctions. *Journal of Economic Theory*, Volumen 152, pp. 1-40.
- Marshall, R. C. & Meurer, M. J., 2004. Bidder collusion and antitrust law: refining the analysis of price fixing to account for the special features of auction markets. *Antitrust Law Journal*, 72(1), pp. 83-118.
- Martello, S. & Toth, P., 1990. *Knapsack problems: algorithms and computer implementations*. s.l.:John Wiley & Sons, Inc..
- Maskin, E. & Riley, J., 1984. Optimal auctions with risk averse buyers. *Econometrica: Journal of the Econometric Society*, pp. 1473-1518.
- Maskin, E. & Riley, J., 2000. Asymmetric auctions. *The Review of Economic Studies*, 67(3), pp. 413-438.
- Matthews, S., 1987. Comparing auctions for risk averse buyers: A buyer's point of view. *Econometrica: Journal of the Econometric Society*, pp. 633-646.
- McAfee, R. P. & McMillan, J., 1987. Auctions and bidding. *Journal of economic literature*, 25(2), pp. 699-738.
- McDonald, C. G. & Slawson, V. C., 2002. Reputation in an internet auction market. *Economic inquiry*, 40(4), pp. 633-650.

- McGill, J. I. & Van Ryzin, G. J., 1999. Revenue management: Research overview and prospects. *Transportation science*, 33(2), pp. 233-256.
- Menakerman, N. & Rom, R., 2001. *Bin packing with item fragmentation*. s.l., s.n., pp. 313-324.
- Milgrom, P., 1989. Auctions and bidding: A primer. *The Journal of Economic Perspectives*, 3(3), pp. 3-22.
- Milgrom, P. R., 2004. *Putting auction theory to work*. s.l.:Cambridge University Press.
- Miller, J. J., 2014. From English to First-Price Sealed Bid: An Empirical Assessment of the Change in Auction Type on Experienced Bidders. *Review of Economic Perspectives*, 14(2), pp. 105-127.
- Muritiba, A. E. F., Iori, M., Malaguti, E. & Toth, P., 2010. Algorithms for the bin packing problem with conflicts. *Infoms Journal on computing*, 22(3), pp. 401-415.
- Nash, J., 1951. Non-cooperative games. *Annals of mathematics*, pp. 286-295.
- Neugebauer, T. & Selten, R., 2006. Individual behavior of first-price auctions: The importance of information feedback in computerized experimental markets. *Games and Economic Behavior*, 54(1), pp. 183-204.
- Ockenfels, A., Reiley, D. & Sadrieh, A., 2006. *Online auctions*, s.l.: s.n.
- Palfrey, T. R., 1983. Bundling decisions by a multiproduct monopolist with incomplete information. *Econometrica: Journal of the Econometric Society*, pp. 463-483.
- Parsons, S., Rodriguez-Aguilar, J. A. & Klein, M., 2011. Auctions and bidding: A guide for computer scientists. *ACM Computing Surveys (CSUR)*, 43(2), p. 10.
- Pesendorfer, M., 2000. A study of collusion in first-price auctions. *The Review of Economic Studies*, 67(3), pp. 381-411.
- Pinker, E. J., 2001. *Using transaction data for the design of sequential, multi-unit, online auctions*, s.l.: s.n.
- Pinker, E. J., Seidmann, A. & Vakrat, Y., 2003. Managing online auctions: Current business and research issues. *Management science*, 49(11), pp. 1457-1484.

- Popkowski Leszczyc, P. T. L. & Hübner, G., 2010. To bundle or not to bundle: Determinants of the profitability of multi-item auctions. *Journal of Marketing*, 74(4), pp. 110-124.
- Porter, R. H. & Zona, J. D., 1992. *Detection of bid rigging in procurement auctions*, s.l.: s.n.
- Presnick, P., Zeckhauser, R., Friedman, E. & Kuwabara, K., 2000. Reputation systems: Facilitating trust in Internet interactions. *Communications of the ACM*, 43(12), pp. 45-48.
- Rajkumar, M., Asokan, P., Anilkumar, N. & Page, T., 2011. A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, 49(8), pp. 2409-2423.
- Robinson, M. S., 1985. Collusion and the Choice of Auction. *The RAND Journal of Economics*, pp. 141-145.
- Robinson, M. S. & others, 1983. *Oil lease auctions: Reconciling economic theory with practice*. s.l.:Centre of Policy Studies, Monash University.
- Rosenthal, R. W. & Wang, R., 1996. Simultaneous auctions with synergies and common values. *Games and Economic Behavior*, 17(1), pp. 32-55.
- Roth, A. E., Ockenfels, A. & others, 2002. Last minute bidding and the rules for ending second price auctions: evidence from ebay and amazon auctions on the internet. *American economic review*, 92(4), pp. 1093-1103.
- Sadykov, R. & Vanderbeck, F., 2013. Bin packing with conflicts: a generic branch-and-price algorithm. *INFORMS Journal on Computing*, 25(2), pp. 244-255.
- Sakawa, M., 2013. *Fuzzy sets and interactive multiobjective optimization*. s.l.:Springer Science & Business Media.
- Saraiva, R. D., Nepomuceno, N. & Pinheiro, P. R., 2015. A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company. *IFAC-PapersOnLine*, 48(3), pp. 490-495.
- Schoenherr, T. & Mabert, V. A., 2008. The use of bundling in B2B online reverse auctions. *Journal of Operations Management*, 26(1), pp. 81-95.
- Scholl, A., Klein, R. & Jurgens, C., 1997. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, 24(7), pp. 627-645.

- Schulz, C., 2015. *Graph Partitioning and Graph Clustering in Theory and Practice*. s.l., s.n.
- Schwerin, P. & Wascher, G., 1997. The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP. *International Transactions in Operational Research*, 4(5), pp. 377-389.
- Shachnai, H. & Tamir, T., 2001. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6), pp. 313-338.
- Sheikhzadeh, M. & Elahi, E., 2013. Product bundling: Impacts of product heterogeneity and risk considerations. *International Journal of Production Economics*, 144(1), pp. 209-222.
- Shubik, M., 1983. Auctions, bidding, and markets: an historical sketch. *Auctions, bidding, and contracting*, pp. 165-191.
- Simon, H. D., 1991. Partitioning of unstructured problems for parallel processing. *Computing systems in engineering*, 2(2-3), pp. 135-148.
- Smith, M. D., Bailey, J. & Brynjolfsson, E., 2001. Understanding digital markets: Review and assesment.
- Stremersch, S. & Tellis, G. J., 2002. Strategic bundling of products and prices: A new synthesis for marketing. *Journal of Marketing*, 66(1), pp. 55-72.
- Subramaniam, R. & Venkatesh, R., 2009. Optimal bundling strategies in multiobject auctions of complements or substitutes. *Marketing Science*, 28(2), pp. 264-273.
- Thelning, K.-E., 2013. *Steel and its heat treatment*. s.l.:Butterworth-Heinemann.
- Trivella, A. & Pisinger, D., 2016. The load-balanced multi-dimensional bin-packing problem. *Computers & Operations Research*, Volumen 74, pp. 152-164.
- Vakrat, Y. & Seidmann, A., 2000. *Implications of the bidders' arrival process on the design of online auctions*. s.l., s.n., pp. 7--pp.
- Varian, H. R., 1995. *Economic Mechanism Design for Computerized Agents*. s.l., s.n., pp. 13-21.
- Venkatesh, R. & Kamakura, W., 2003. Optimal bundling and pricing under a monopoly: Contrasting complements and substitutes from independently valued products*. *The Journal of business*, 76(2), pp. 211-231.

- Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1), pp. 8-37.
- Vigo, D., 2013. Bin Packing Approximation Algorithms: Combinatorial Analysis. *Handbook of Combinatorial Optimization: Supplement*, Volumen 1, p. 151.
- Waltman, L. & van Eck, N. J., 2013. A smart local moving algorithm for large-scale modularity-based community detection. *The European Physical Journal B*, 86(11), p. 471.
- Wang, W. & Zoltan Hidvegi, A. B. W., 2001. Designing mechanisms for e-commerce security: An example from sealed-bid auctions. *International Journal of Electronic Commerce*, 6(2), pp. 139-156.
- Wascher, G. & Gau, T., 1996. Heuristics for the integer one-dimensional cutting stock problem: A computational study. *Operations-Research-Spektrum*, 18(3), pp. 131-144.
- Wee, T. S. & Magazine, M. J., 1982. Assembly line balancing as generalized bin packing. *Operations Research Letters*, 1(2), pp. 56-58.
- Wilcox, R. T., 2000. Experts and amateurs: The role of experience in Internet auctions. *Marketing Letters*, 11(4), pp. 363-374.
- Wong, K.-C., 2015. *A Short Survey on Data Clustering Algorithms*. s.l., s.n., pp. 64-68.
- WorldSteelAssociation, 2016. *World Steel In Figures 2016*. s.l.:s.n.
- Xu, R. & Wunsch, D., 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3), pp. 645-678.
- Yao, A. C.-C., 1980. New algorithms for bin packing. *Journal of the ACM (JACM)*, 27(2), pp. 207-227.
- Yokoo, M., Sakurai, Y. & Matsubara, S., 2002. Robust combinatorial auction protocol against false-name bids. En: *Game Theory and Decision Theory in Agent-Based Systems*. s.l.:Springer, pp. 355-373.
- Yuan, Y., Li, Y.-j. & Wang, Y.-q., 2014. *An improved ACO algorithm for the bin packing problem with conflicts based on graph coloring model*. s.l., s.n., pp. 3-9.

Zhang, Y., Lee, C., Niyato, D. & Wang, P., 2013. Auction approaches for resource allocation in wireless systems: A survey. *Communications Surveys & Tutorials, IEEE*, 15(3), pp. 1020-1041.

Zhou, L. y otros, 2016. A dynamic graph-based scheduling and interference coordination approach in heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 65(5), pp. 3735-3748.

Zhou, Y., Cheng, H. & Yu, J. X., 2009. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, 2(1), pp. 718-729.

Zhu, D., 2016. Max--min Bin Packing Algorithm and its application in nano-particles filling. *Chaos, Solitons & Fractals*, Volumen 89, pp. 83-90.