

Trabajo Fin de Grado

Ingeniería de Tecnologías Industriales

Programación de Operaciones en Procesos de Fabricación Aditiva

Autor: Enrique Hernández Malfeitos

Tutor: Jose Manuel Framiñán Torres

Dpto. Organización Industrial y Gestión de
Empresas I

Escuela Técnica Superior de Ingeniería

Sevilla, 2022



Trabajo Fin de Grado
Ingeniería de Tecnologías Industriales

Programación de Operaciones en Procesos de Fabricación Aditiva

Autor:

Enrique Hernández Malfeitos

Tutor:

Jose Manuel Framiñán Torres

Catedrático de Universidad

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022

Trabajo Fin de Grado: Programación de Operaciones en Procesos de Fabricación Aditiva

Autor: Enrique Hernández Malfeitos

Tutor: Jose Manuel Framiñán Torres

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

*A mi madre, por todo el apoyo y
cariño que ha mostrado hacia mi
persona toda mi vida.*

En el este proyecto se analiza la tecnología de fabricación aditiva estudiando sus principales ventajas e inconvenientes. Una vez examinado dicho método de fabricación y las características de la maquinaria con la que se fabrican las piezas, se detecta que uno de los problemas que presenta es el alto tiempo que se necesita invertir en la fabricación de las piezas. Por lo tanto, se trata de solucionar dicho problema llevando a cabo una propuesta de un algoritmo el cual realice la programación de las operaciones para ofrecer así una solución que consiste en aportar una secuencia de trabajos en los cuales se definen que piezas van agrupadas en lotes y en que máquina deben procesarse dichos lotes, con el objetivo de minimizar el tiempo total de finalización (Makespan). En dicho problema intervienen distintos factores que son explicados convenientemente, como, por ejemplo, las características de las máquinas y de las piezas. Para su resolución se elabora una metaheurística del tipo de búsqueda de vecindario variable la cual incluye distintos algoritmos. Una vez confeccionada, se estudian los resultados obtenidos para una serie de instancias y se proponen distintas alternativas diferentes para así llevar a cabo una comparativa y finalmente decidir qué alternativa proporciona mejores resultados según los objetivos del usuario.

Abstract

In this project, additive manufacturing technology is analyzed, studying its main advantages and disadvantages. After examining said manufacturing method and the characteristics of the machinery with which the parts are manufactured, it is detected that one of the problems it presents is the high time that needs to be invested in the manufacture of the parts. Therefore, it is about solving this problem by carrying out a proposal of an algorithm which performs the programming of the operations to offer a solution that consists of providing a sequence of works in which the pieces are grouped in batches. and in which machine these batches should be processed, with the aim of minimizing the total completion time (Makespan). Different factors are involved in this problem that are conveniently explained, such as, for example, the characteristics of the machines and parts. For its resolution, a metaheuristic of the variable neighborhood search type is elaborated, which includes different algorithms. Once made, the results obtained for a series of instances are studied and different alternatives are proposed to carry out a comparison and finally decide which alternative provides better results according to the user's objectives.

Resumen	ix
Abstract	xi
Índice	xii
Índice de Tablas	xiv
Índice de Figuras	xvi
1 Objeto del proyecto	1
1.1 <i>Motivación de la elección del problema</i>	1
1.2 <i>Definición de los capítulos del trabajo</i>	2
2 Descripción del problema	5
2.1 <i>Introducción a la fabricación aditiva</i>	5
2.1.1 <i>Procedimiento de fabricación de piezas mediante impresión 3D</i>	5
2.1.2 <i>Técnicas de fabricación aditiva</i>	6
2.1.3 <i>Ventajas y desventajas de la fabricación aditiva</i>	7
2.2 <i>Problema abordado</i>	8
2.3 <i>Análisis del problema</i>	8
3 Descripción de la metodología	11
3.1 <i>Metaheurística aplicada</i>	11
3.2 <i>Problema de empaquetado</i>	12
3.3 <i>Búsqueda de vecindario variable</i>	18
3.4 <i>Algoritmo de autoaprendizaje</i>	19
3.5 <i>Perturbación de la solución</i>	22
3.6 <i>Criterio de parada del algoritmo</i>	22
3.7 <i>Resumen de la metaheurística</i>	23
4 Implementación de la metaheurística	25
4.1 <i>Equipo y herramientas utilizados</i>	25
4.2 <i>Datos de entrada del problema</i>	25
4.3 <i>Algoritmo e instancias</i>	26
4.4 <i>Ejemplo numérico</i>	26
4.4.1 <i>Ejemplo numérico. Problema de empaquetado</i>	27
4.4.2 <i>Ejemplo numérico. Vecindades y búsqueda de vecindario</i>	28
4.4.3 <i>Ejemplo numérico. Resultados</i>	32
5 Variantes propuestas	33
5.1 <i>Variante 1. Modificación 1 de la vecindad “Flip”</i>	33
5.2 <i>Variante 2. Modificación 2 de la vecindad “Flip”</i>	34
5.3 <i>Variante 3. Modificación de la vecindad “Swap”</i>	34
5.4 <i>Variante 4. Modificación de la vecindad “Job Alloc”</i>	35
5.5 <i>Variante 5. Modificación del problema de empaquetado</i>	36
5.6 <i>Variantes 6 y 7. Combinación de diferentes modificaciones</i>	37
6 Resultados y conclusiones	39
6.1 <i>Resultados de la metaheurística</i>	40

6.2	<i>Comparativa de las variantes con la metaheurística</i>	42
6.3	<i>Conclusión</i>	44
Referencias		47
7	Anexo A. Ejemplo numérico. Procedimiento de algoritmos de empaquetado	49
7.1	<i>Procedimiento del algoritmo NFDH</i>	49
7.2	<i>Procedimiento del algoritmo FFDH</i>	52
7.3	<i>Procedimiento del algoritmo BFDH</i>	55
8	Anexo B. Resultados de la metaheurística	59
9	Anexo C. Resultados de las variantes propuestas	63
9.1	<i>Resultados de variante 1</i>	64
9.2	<i>Resultados de variante 2</i>	67
9.3	<i>Resultados de variante 3</i>	70
9.4	<i>Resultados de variante 4</i>	73
9.5	<i>Resultados de variante 5</i>	76
9.6	<i>Resultados de variante 6</i>	79
9.7	<i>Resultados de variante 7</i>	82

ÍNDICE DE TABLAS

Tabla 1. Configuración de parámetros.	26
Tabla 2. Máquinas del ejemplo numérico.	27
Tabla 3. Partes del ejemplo numérico.	27
Tabla 4. Ejemplo numérico. Solución del problema de empaquetado NFDH.	27
Tabla 5. Ejemplo numérico. Solución del problema de empaquetado FFDH.	28
Tabla 6. Ejemplo numérico. Solución del problema de empaquetado BFDH.	28
Tabla 7. Ejemplo numérico. Tiempos de terminación de problema de empaquetado.	28
Tabla 8. Ejemplo numérico. Vecindad de tipo Flip.	29
Tabla 9. Ejemplo numérico. Vecindad de tipo Swap.	30
Tabla 10. Ejemplo numérico. Vecindad de tipo Job Alloc.	31
Tabla 11. Ejemplo numérico. Resultados.	32
Tabla 12. Comparativa en instancias pequeñas y medianas.	40
Tabla 13. Comparativa en instancias grandes.	41
Tabla 14. Comparativa de variantes. Instancias pequeñas y medianas.	42
Tabla 15. Comparativa de comportamiento de vecindades. Instancias pequeñas y medianas.	42
Tabla 16. Comparativa de variantes. Instancias grandes.	43
Tabla 17. Comparativa de comportamiento de vecindades. Instancias grandes.	43
Tabla 18. Ejemplo numérico. Partes ordenadas de menor a mayor altura por máquina.	49
Tabla 19. Resultados del algoritmo NFDH en la máquina 1.	50
Tabla 20. Resultados del algoritmo NFDH en la máquina 2.	52
Tabla 21. Resultados del algoritmo FFDH en la máquina 1.	53
Tabla 22. Resultados del algoritmo FFDH en la máquina 2.	55
Tabla 23. Resultados del algoritmo BFDH en la máquina 1.	56
Tabla 24. Resultados del algoritmo BFDH en la máquina 2.	58
Tabla 25. Metaheurística básica. Resultados de instancias de tipo B.	60
Tabla 26. Metaheurística básica. Resultados de instancias de tipo C.	61
Tabla 27. Metaheurística básica. Resultados de instancias de tipo R.	62
Tabla 28. Variante 1. Resultados de instancias de tipo B.	64
Tabla 29. Variante 1. Resultados de instancias de tipo C.	65
Tabla 30. Variante 1. Resultados de instancias de tipo R.	66
Tabla 31. Variante 2. Resultados de instancias de tipo B.	67
Tabla 32. Variante 2. Resultados de instancias de tipo C.	68
Tabla 33. Variante 2. Resultados de instancias de tipo R.	69
Tabla 34. Variante 3. Resultados de instancias tipo B.	70

Tabla 35. Variante 3. Resultados de instancias de tipo C.	71
Tabla 36. Variante 3. Resultados de instancias de tipo R.	72
Tabla 37. Variante 4. Resultados de instancias de tipo B.	73
Tabla 38. Variante 4. Resultados de instancias de tipo C.	74
Tabla 39. Variante 4. Resultados de instancias de tipo R.	75
Tabla 40. Variante 5. Resultados de instancias de tipo B.	76
Tabla 41. Variante 5. Resultados de instancias de tipo C.	77
Tabla 42. Variante 5. Resultados de instancias de tipo R.	78
Tabla 43. Variante 6. Resultados de instancias de tipo B.	79
Tabla 44. Variante 6. Resultados de instancias de tipo C.	80
Tabla 45. Variante 6. Resultados de instancias de tipo R.	81
Tabla 46. Variante 7. Resultados de instancias de tipo B.	82
Tabla 47. Variante 7. Resultados de instancias de tipo C.	83
Tabla 48. Variante 7. Resultados de instancias de tipo R.	84

ÍNDICE DE FIGURAS

Ilustración 1. Gráfico del aumento de interés por la fabricación mediante impresión 3D [6].	2
Ilustración 2. Concepto de fabricación sustractiva comparado con AM [7]	5
Ilustración 3. Procedimiento para la elaboración de una pieza mediante fabricación aditiva [8].	6
Ilustración 4. Esquema de problema de empaquetado. Elaboración propia.	13
Ilustración 5. Diagrama del algoritmo NFDH. Elaboración propia.	15
Ilustración 6. Diagrama del algoritmo FFDH. Elaboración propia.	16
Ilustración 7. Diagrama del algoritmo BFDH. Elaboración propia.	17
Ilustración 8. Ejemplo de vecino de tipo Flip. Elaboración propia.	18
Ilustración 9. Ejemplo vecino tipo Swap. Elaboración Propia.	18
Ilustración 10. Ejemplo vecindad tipo Job Alloc 1. Elaboración propia.	19
Ilustración 11. Diagrama de algoritmo Q-Learning. Elaboración propia.	21
Ilustración 12. Metaheurística completa. Elaboración propia.	23
Ilustración 13. Diagrama de Gantt del ejemplo numérico. Elaboración propia.	32
Ilustración 14. Modificación de la vecindad Flip. Elaboración propia.	34
Ilustración 15. Comparación de vecindades Swap. Elaboración propia.	35
Ilustración 16. Ejemplo de vecindad tipo Job Alloc 2. Elaboración propia.	36
Ilustración 17. Ejemplo de vecindad tipo Job Alloc modificada. Elaboración propia.	36

1 OBJETO DEL PROYECTO

El objeto de este proyecto es el desarrollo de una metaheurística con la cual se consiga programar una serie de trabajos a procesar en un conjunto de impresoras 3D de diferentes características con el objetivo de terminar todos los trabajos lo antes posible.

En este proyecto se estudia la mejor manera de agrupar las piezas en lotes de trabajo para ser procesadas por un conjunto de máquinas de fabricación aditiva (de diferentes especificaciones), así como las máquinas por las que serán fabricadas, mediante la elaboración de algoritmos. La agrupación de piezas en lotes es necesaria debido a que las máquinas tienen la capacidad de conformar varias piezas simultáneamente en una bandeja de fabricación. En dicho análisis intervienen una serie de parámetros, como por ejemplo las capacidades de las máquinas, a nivel de área de procesamiento, o el tiempo de formación por unidad de volumen, que definen los tiempos de fabricación de cada grupo de partes.

La metaheurística estudiada tiene base en estudios previos de la literatura [1] y se elaboran diversas variaciones, de modo que se pueda hacer una comparación final en la cual se seleccione el programa que obtiene mejores resultados dependiendo de los objetivos que se tengan. El método en cuestión consta de dos fases: una fase inicial en la cual se elabora una agrupación de piezas en lotes y, a continuación, se inicia una segunda fase en la cual se realiza una búsqueda local iterada en la cual se estudian soluciones similares con el fin de reducir el tiempo total de fabricación de los trabajos.

Para el diseño del programa se ha hecho uso del lenguaje de programación Python [2] en el entorno de desarrollo Anaconda® [3] y de conceptos de la librería Scheptk [4]. En cuanto a las metodologías usadas tanto para el problema de empaquetado, como para la búsqueda de vecinos, se ha tomado como referencia lo descrito en [1] así como para el algoritmo Q-Learning, que permite realizar el proceso iterativo de búsqueda local de manera más eficiente.

1.1 Motivación de la elección del problema

La fabricación aditiva (AM, Additive Manufacturing) es una tecnología en desarrollo capaz de revolucionar la industria manufacturera, ya que su principal característica es la polivalencia que posee este sistema a la hora de imprimir objetos con un nivel de detalle muy alto, aparte de simplificar todas las operaciones de elaboración del producto en una sola máquina. Además, mediante dicha tecnología, es posible explotar una amplia variedad de materiales para crear productos que van desde prótesis médicas hasta componentes de la industria aeroespacial. Sin embargo, la mencionada tecnología, como toda tecnología primeriza, tiene desventajas frente a otras tecnologías de fabricación, y la principal es el tiempo invertido en la fabricación de los productos, cuya solución es mejorar las prestaciones de las máquinas, refinando diferentes partes de su proceso, añadiendo técnicas nuevas de conformado, herramientas distintas, reglas de diseño innovadoras, etc. como se muestra en [5]. Pero también se puede llegar a optimizar la fabricación, en términos de tiempo de fabricación, con este tipo de maquinaria de forma significativa haciendo hincapié en la programación de las operaciones que se llevan a cabo. En la ilustración 1 [6] se representa el aumento del uso de la fabricación AM en los últimos años.

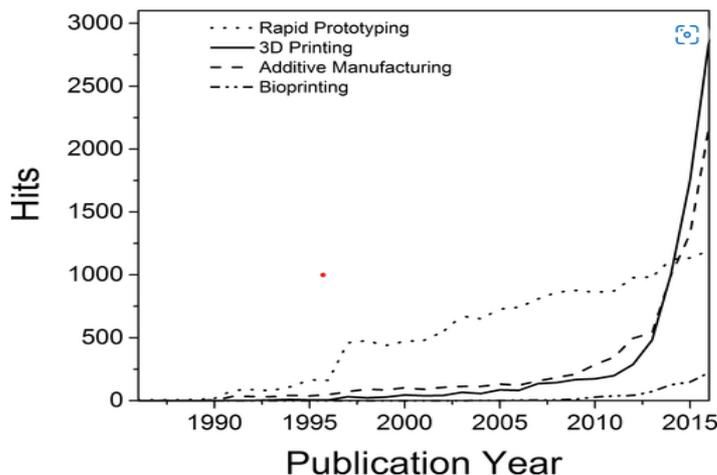


Ilustración 1. Gráfico del aumento de interés por la fabricación mediante impresión 3D [6].

Como se presenta en [1], en torno a dicho método de fabricación, normalmente las contribuciones científicas se suelen especializar en diferentes decisiones del proceso, como, por ejemplo, determinar la orientación de las piezas a la hora de ser procesadas, el orden de estas, o el problema de empaquetado previo para ahorrar tiempos de Set-Up, por lo que es interesante desarrollar un método que trate de aportar una buena solución desde una perspectiva de inicio a fin del procedimiento.

Otra característica que hace de interés dicho problema es que se trata de la formulación más genérica en cuanto a fabricación aditiva, esto quiere decir que, la solución a dicho problema abarca tanto el entorno en cuestión (disposición de máquinas 3D con diferentes especificaciones en paralelo) como el problema de máquinas 3D idénticas en paralelo o el problema de una única máquina 3D.

1.2 Definición de los capítulos del trabajo

El proyecto se desarrolla en seis capítulos: Objeto del Proyecto, Descripción del Problema, Descripción de la metodología, Implementación de la metodología, Variantes propuestas y Resultados y Conclusión.

En el primer capítulo, “Objeto del Proyecto”, se introduce el problema de programación de operaciones aplicado a la fabricación aditiva. También se explica en qué consiste, concretando los pasos que se han seguido para obtener la resolución y aquellas variables que intervienen. A continuación, se explica el motivo por el cual se pretende abarcar el trabajo en cuestión y la importancia que tiene en su entorno. Finalmente, se presenta un breve resumen del contenido de cada uno de los bloques de los que dispone el documento con el objetivo de facilitar al lector el temario de una manera sencilla.

El segundo capítulo, “Descripción del Problema”, está enfocado en el estado del arte de la fabricación aditiva, en el cual se exponen los distintos modos de fabricación con impresoras 3D, los pros y contras de dicha tecnología, y una revisión de la literatura para obtener referencias y resultados con los que comparar nuestra metaheurística y el porqué de su elección.

En el tercer capítulo, “Descripción de la metodología”, se explican las diferentes maneras de resolver el problema, sus ventajas e inconvenientes, y el método elegido, así como su correspondiente justificación y explicación detallada.

En el cuarto capítulo, “Implementación de la metodología”, se expone toda la información necesaria para la correcta comprensión de la metodología utilizada, como las decisiones que se han tomado a priori, o las hipótesis que influyen en el desarrollo de la técnica. Este bloque, para ser mejor comprendido, se basa en todo momento en un ejemplo de un problema a solucionar.

En el quinto capítulo, “Variantes propuestas”, se introducen diferentes modificaciones que afectan a la metaheurística con el objetivo de mejorarla en función de los objetivos que se tengan y de comparar finalmente todas las alternativas.

En el sexto capítulo, “Resultados y conclusión”, se presenta un resumen de los resultados obtenidos con la metaheurística elaborada, así como de las diferentes variaciones que se han llevado a cabo tomando como base la metaheurística principal, la comparación de las mismas con diferentes criterios, y una conclusión generalizada acerca del problema en cuestión, de la metaheurística y los resultados obtenidos, y de los posibles caminos que podrían tomar los problemas de estas características en un futuro.

2 DESCRIPCIÓN DEL PROBLEMA

2.1 Introducción a la fabricación aditiva

Para modificar la forma de un material existen distintas técnicas de fabricación: la fabricación sustractiva, la fabricación por deformación, y la fabricación por adición.

En la fabricación sustractiva las formas se obtienen mediante la exclusión de material de un volumen de material mayor. En la fabricación por deformación, la geometría se consigue al dar forma a un bloque de material mediante una matriz con una cavidad con la figura que se quiere obtener. Y finalmente, mediante la fabricación por incremento de material (o adición), la geometría se obtiene mediante la manipulación del material de forma minuciosa y depositarlo de forma muy precisa para construir el sólido. Las impresoras 3D implementan la última forma de fabricación descrita a escala micrométrica. En la ilustración 2 [7] se puede observar gráficamente la diferencia entre la fabricación aditiva y la fabricación sustractiva.

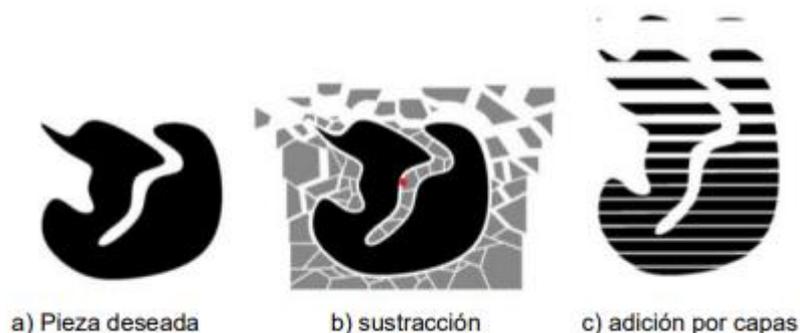


Ilustración 2. Concepto de fabricación sustractiva comparado con AM [7]

2.1.1 Procedimiento de fabricación de piezas mediante impresión 3D

Una vez se conoce el fundamento principal de la fabricación aditiva, se procede a definir, a rasgos generales, el proceso que siguen las impresoras 3D para conformar los productos [8].

Para comenzar, se necesita un modelo digital 3D que normalmente se obtiene con programas informáticos, como por ejemplo Autodesk® o SolidWorks®. Una de las características que hacen a dicha tecnología tan interesante es que, de manera digital, se pueden llegar a obtener geometrías muy complejas que mediante técnicas tradicionales jamás se conseguirían. Además, también cabe la posibilidad de llevar a cabo ingeniería inversa, es decir, obtener el modelo digital de una forma física mediante un escaneo 3D.

A continuación, se convierte el modelo anteriormente formado en un fichero estandarizado para la lectura en la máquina. Este formato, por regla general es el .STL, aunque los tipos de fichero soportados por las impresoras 3D evolucionan de manera muy rápida al igual que las propias máquinas.

En el siguiente paso, mediante un programa informático denominado “slicer” se prepara el fichero .STL para el procesamiento en la impresora 3D. En dicho procedimiento se establecen valores para los distintos parámetros de fabricación, como, por ejemplo, la temperatura de conformado, la adición de material de soporte si fuera necesario, las capas de material o la orientación de la pieza.

Finalmente, una vez preparado el archivo para su procesamiento con todos sus valores ajustados e información necesaria, se introduce en la impresora 3D para la fabricación de la pieza. Además, las máquinas también cuentan con la posibilidad de variar la configuración de los parámetros, en caso de que fuera necesario, durante el procedimiento de conformado. Dependiendo de la máquina usada o la finalidad de la pieza obtenida, el resultado puede ser final o no, y en caso de que no fuese definitivo, pasa por procesos de post-procesado para refinar su forma. A modo de resumen, se puede observar en la ilustración 3 [8] el proceso genérico que se lleva a cabo desde principio a fin a la hora de fabricar una pieza en una impresora 3D.



Ilustración 3. Procedimiento para la elaboración de una pieza mediante fabricación aditiva [8].

2.1.2 Técnicas de fabricación aditiva

Aunque el proceso de fabricación genérico para la fabricación aditiva es el descrito anteriormente, existen diferentes tecnologías de fabricación aditiva [7, 9, 10] que tienen ciertas peculiaridades. Según la norma UNE-EN ISO/ASTM 52900:2017 los procesos de AM vienen clasificados en siete grupos:

- **Inyección Aglutinante (BJ):** se trata de una técnica en la cual se conforma el objeto mediante la deposición de un líquido aglutinante sobre un lecho de polvo, de manera que el líquido se queda adherido al polvo dando la forma del objeto. Para el conformado total del objeto se repite dicho proceso en varias capas. Usada principalmente para objetos de gran tamaño y con formas complejas.
- **Deposición Directa de Energía (Directed Energy Deposition, DED):** se le da la forma deseada al producto en una atmósfera inerte mediante alimentación continua de material metálico en un punto focal de un láser formando una deposición de un material fundido.
- **Proyección de material (Material Jetting):** la fabricación de formas mediante esta técnica tiene base en la impresión con inyección de tinta, sin embargo, a diferencia de las impresoras convencionales de dos

dimensiones, en aquellas de fabricación aditiva se depositan y posteriormente se solidifican gotas de material, capa a capa, para formar un objeto en tres dimensiones.

- Fusión de Lecho en Polvo (Powder Bed Fusion, PBF): este método de fabricación es similar al de deposición directa de energía, con la diferencia de que el material depositado ha sido fundido previamente, mientras que en la técnica DED se hace a la vez que se realiza la alimentación. Para el conformado de las piezas, se deposita en un plano una capa de polvo del material, después se fija dicho polvo con la forma de la pieza y se calienta la superficie deseada para así formar una capa. Repitiendo el proceso capa a capa se consigue la forma final del objeto que se desea imprimir. Dentro de la clasificación PBF, las técnicas más conocidas son la Selective Laser Sintering (SLS) y la Selective Laser Melting (SLM), donde esta última se caracteriza por tener una mayor capacidad de producir objetos con mayor complejidad, de manera más compacta y con mayor rapidez debido al uso de láseres mucho más potentes. También se encuentran otros tipos de fabricación de tipo PBF como la técnica Electron Beam Melting (EBM), que se caracteriza por usar un haz de electrones en vez de un láser para la fusión.
- Laminado de Hojas (Sheet Lamination): en este proceso la forma general del objeto se consigue con la superposición de cierto material, y el exceso de material, para definir los detalles de la forma se eliminan cortándolo con un láser o mecánicamente.
- Fotopolimerización en cuba (VAT Photopolymerization, VP): aquí, una luz (ya sea ultravioleta, o digital de alta definición) es proyectada sobre una resina fotosensible para ser solidificada. Capa a capa, el componente, a medida que se va solidificando, va formando la estructura tridimensional. Las técnicas más usadas de las del grupo de VP son la Estereolitografía (Stereolithography, SLA), que es también una de las técnicas de fabricación aditiva más antiguas, y la de Procesado por Luz Digital (Digital Light Processing, DLP).
- Extrusión de material (Material Extrusion): es la técnica que más crecimiento ha logrado últimamente debido a que posee las grandes ventajas que presentan el resto de las técnicas. Dentro de dicho método de fabricación aditiva existen numerosas variantes, dentro de las cuales la más difundida es el modelado por deposición fundida (Fused Filament Modeling, FDM), este método consiste en ir fabricando el objeto depositando el material fundido en forma de finas líneas de material que forman capas y finalmente el objeto completo.

2.1.3 Ventajas y desventajas de la fabricación aditiva

La tecnología de fabricación aditiva está ganando mucho terreno en el ámbito industrial y está experimentando un gran desarrollo debido a sus notables ventajas frente a las otras técnicas de fabricación ya mencionadas en el inicio de la sección presente, por ello, se van a mostrar los principales pros y contras de la tecnología de fabricación aditiva [7,9].

En primer lugar, esta técnica tiene la gran ventaja de que existe una muy alta precisión al igual que personalización a un coste bajo.

En segundo lugar, también es importante destacar que todo el procedimiento de conformado del objeto viene atribuido a un solo proceso en una única máquina, mientras que en las técnicas de manufactura convencionales son necesarios distintos pasos en distintos equipos hasta llegar a la forma final. Esto significa un gran ahorro en maquinaria, mantenimiento y personal. A parte de la unificación de todo el proceso de fabricación, es una gran ventaja la flexibilidad de dicha técnica para modificar los diseños de los productos sin costes adicionales y sin generar retrasos en la producción.

En tercer lugar, como se lleva a cabo una producción bajo demanda, se reducen costes de almacenamiento, logística y transporte.

Finalmente, en cuanto al ámbito medioambiental, es una técnica muy limpia ya que se generan muy pocos residuos y cabe la posibilidad de reciclarlos in-situ.

Por el contrario, a pesar de ser un sistema innovador y con un gran potencial, la fabricación aditiva tiene dos principales problemas. El primero se relaciona con la alta inversión que se tiene que realizar para tener un sistema de fabricación de este tipo debido al alto coste de los equipos, dicha cuestión será solucionada en el futuro si se normaliza este tipo de técnica de fabricación y aumenta la demanda de las impresoras 3D, reduciéndose así tanto su coste como su mantenimiento. La segunda está asociada al elevado tiempo de procesamiento de los trabajos

en las máquinas, esto se debe a la complejidad de las técnicas usadas en el conformado de las piezas capa a capa. Este trabajo se centra por lo tanto en optimizar la fabricación llevando a cabo una programación de la producción que permita tener el trabajo finalizado lo antes posible intentando solucionar, en la medida de lo posible, la segunda problemática expuesta anteriormente.

2.2 Problema abordado

El objeto de este proyecto está enfocado en la problemática del tiempo de fabricación en las técnicas de fabricación aditiva, concretamente en la maquinaria con tecnología SLM (Selective Laser Melting), descrita en la primera sección del capítulo actual. El objetivo es minimizar el tiempo de procesamiento total (también conocido como Makespan) de un conjunto de piezas en un entorno en el cual se tiene un grupo de máquinas con distintas características.

Este tipo de máquinas cuentan con una serie de especificaciones que influyen en gran medida en el tiempo de procesamiento de las partes. Por una parte, se tiene, en primer lugar, el tiempo de fabricación por unidad de volumen de material, en segundo lugar, es importante el tiempo de aplicación de capas de polvo, y finalmente, el tiempo de set-up o de preparación de la máquina, debido a que las máquinas pueden procesar varias piezas simultáneamente. Por otra parte, en relación con las características relacionadas con las capacidades físicas que tiene el equipo, se encuentra el área máxima permitida, que limita la suma de las áreas de las piezas situadas en la bandeja de fabricación de la máquina y la altura máxima que soporta la máquina.

Las máquinas disponen de una bandeja de fabricación en la cual se pueden formar varias piezas simultáneamente, por lo que, cada vez que se quiera producir una pieza (también denominadas partes), a parte del tiempo de formación de las partes en la impresora, es necesario dedicar un tiempo a la preparación de la máquina para su procesamiento, como puede ser la configuración de esta, la limpieza o el tratamiento del archivo digital. Es por ello por lo que, para reducir tiempos de set-up y aprovechar al máximo el área disponible en la bandeja de las máquinas, en el tiempo total de producción, las distintas partes se agrupan en lotes (también denominados trabajos) intentando maximizar la ocupación del área permitida de la impresora 3D. A esta cuestión se le ha prestado una gran atención en la literatura [11] ya que para el caso en el cual el entorno de fabricación es de una única máquina, el problema de minimización del Makespan es simplemente resolver de la mejor manera posible el problema de agrupamiento de partes en trabajos.

Este proyecto tiene base en estudios previos acerca de la problemática descrita. Se trata el problema llevando a cabo simplificaciones [1]. Por una parte, se considera que las piezas entran en la máquina con una orientación predefinida, suposición que permite evitar el estudio de la geometría exacta de las partes, centrándose el problema únicamente en que no se sobrepase el área máxima permitida por la máquina en cuestión. Por otra parte, usar solamente un tipo de material para la fabricación de las piezas [12]. Además, se considera como base el modelo propuesto para el entorno de máquinas no relacionadas en [13].

Como se comenta en el capítulo 1, la solución del problema se aborda elaborando una metaheurística, puesto que el único enfoque ya planteado ha sido usar un programa informático para la resolución del modelo exacto, obteniéndose resultados en tiempos computacionales muy altos [1]. Además, como se dispone de los resultados obtenidos tanto con solucionadores exactos como con metaheurísticas ya confeccionadas, se puede hacer una comparativa de la eficacia de nuestra propuesta frente a las ya elaboradas.

2.3 Análisis del problema

En este punto se describe de manera formal el problema a tratar tal como se presenta en [13] y en [1]. Para expresar el problema se usa la notación descrita en [14], para así obtener un modelo del problema. En cuanto al entorno, el problema se sitúa en un entorno de máquinas paralelas no relacionadas, dicho entorno se representa con R_m . Las restricciones con las que se cuenta son aquellas relacionadas con la agrupación de partes en lotes (problema de empaquetado) (Batch), y las que son intrínsecas de la fabricación aditiva ({AM}), expresadas en el modelo matemático presentado en esta sección. Finalmente, el objetivo del problema es minimizar el tiempo total de fabricación (Makespan), por lo que, la intención es que el tiempo de finalización del último trabajo procesado sea mínimo, dicho objetivo se expresa como C_{max} . Por lo tanto, el modelo viene definido por:

$$R_m \mid \text{Batch}, \{AM\} \mid C_{max}$$

Donde, en primer lugar, se representa el entorno, en segundo lugar, las restricciones y finalmente el objetivo del problema [14].

Las constantes asociadas a los parámetros específicos para las máquinas son:

- VT (Volume time): tiempo de formación de la máquina por unidad de volumen, expresado en horas por centímetro cúbico.
- HT (Height time): tiempo dedicado a la aplicación de capas de polvo, que se repite para cada capa, variable relacionada con la parte más alta de cada trabajo, expresado en horas por centímetro.
- SET (Set-Up time): tiempo de configuración necesario para inicializar y limpiar la máquina, expresado en horas.
- MA (Machine area): el área de producción de la bandeja de la máquina, expresada en centímetros cuadrados.
- MH (Machine height): la altura máxima soportada por la máquina, expresada en centímetros.

Como se dispone de máquinas con diferentes valores para dichas características se usa el subíndice m que indica la máquina ante la cual se hace referencia.

Las constantes asociadas a los parámetros específicos para las partes son:

- h (Height): altura de la pieza, expresada en centímetros.
- a (Area): superficie de la pieza, expresada en centímetros cuadrados.
- v (Volume): volumen que ocupa cada pieza, expresado en centímetros cúbicos.

Para indicar a que parte se hace referencia se usa el subíndice i .

Tanto las características de las máquinas como de las partes son datos a la hora de llevar a cabo el algoritmo y proceden de una batería de instancias que se comentan en la sección 4.2.

El problema consiste en la programación de operaciones a realizar por un conjunto de impresoras 3D ($m \in M = \{1, 2, \dots, p\}$), cuyo objetivo es procesar por lotes de manera factible, minimizando el Makespan, un conjunto de partes ($i \in I = \{1, 2, \dots, q\}$) en trabajos ($j \in J = \{1, 2, \dots, k\}$) para ser secuenciados en las máquinas AM.

Obviamente $k \leq q$ ya que, como máximo, se puede disponer de una parte en cada trabajo. Además, cabe destacar que si hay alguna parte que exceda el área permitida en la bandeja de la máquina (MA_m) o su altura máxima (MH_m) ésta no podrá ser procesada por dicha máquina en ninguna circunstancia.

Es importante estudiar el modelo matemático debido a que en él se recogen todos los detalles y restricciones del problema y sirve de ayuda para la orientación del programa cuando se comience con el desarrollo del algoritmo.

Para la definición del modelo es necesario explicar una serie de variables de decisión de tipo booleano:

- X_{mij} : la variable vale 1 si la parte i está contenida en el trabajo j y está programada en la máquina m . En caso contrario valdrá 0.
- Z_{mj} : la variable vale 1 si el trabajo j en la máquina m contiene al menos una parte. En caso contrario valdrá 0.

A parte de las variables descritas, se definen aquellas asociadas al tiempo de procesamiento de los trabajos en las máquinas (PT_{mj} , processing time of job j in machine m) donde se van a programar y a su vez el tiempo de finalización (C_{mj} , completion time of job j on machine m) de los mismos.

Para la definición del tiempo de procesamiento de los trabajos se tiene en cuenta: el tiempo de preparación de la máquina, el tiempo de procesamiento por unidad de volumen de material y el tiempo asociado a la aplicación de las capas de polvo, que depende de la pieza de mayor altura del trabajo [13].

La fórmula para la obtención del tiempo de fabricación es la siguiente:

$$PT_j = SET_m \cdot Z_{mj} + VT_m \cdot \sum_{i \in I} v_i \cdot X_{mij} + HT_m \cdot \max\{h_i \cdot X_{mij}\} \quad (1)$$

Y la fórmula asociada a la obtención del tiempo de terminación del trabajo j en la máquina m es:

$$C_{mj} \geq C_{m(j-1)} + PT_{mj} \quad (2)$$

El modelo del problema en cuestión es el siguiente [13]:

$$\text{Minimize} \quad \text{Max } C_{mj} \quad (3)$$

$$m \in M, j \in J$$

$$s.a \quad \sum_{m \in M} \sum_{j \in J} X_{mji} = 1 \quad \forall i \in I \quad (4)$$

$$\sum_{i \in I} a_i \cdot X_{mij} \leq MA_m \quad \forall m \in M, \forall j \in J \quad (5)$$

$$h_i \cdot X_{mji} \leq MH_m \quad \forall m \in M, \forall j \in J, \forall i \in I \quad (6)$$

$$\sum_{i \in I} X_{m(j+1)i} \leq \psi \cdot \sum_{i \in I} X_{mji} \quad \forall m \in M, \forall j \in k-1 \quad (7)$$

$$C_{m(j-1)} + PT_{mj} \leq C_{mj} \quad \forall m \in M, \forall j \in J \quad (8)$$

$$C_{m0} = 0 \quad \forall m \in M \quad (9)$$

$$PT_j = SET_m \cdot Z_{mj} + VT_m \cdot \sum_{i \in I} v_i \cdot X_{mij} + HT_m \cdot \max\{h_i \cdot X_{mij}\} \quad (10)$$

$$X_{mji}, Z_{mj} \in \{0, 1\} \quad \forall m \in M, \forall j \in J, \forall i \in I \quad (11)$$

El modelo está compuesto por restricciones que impiden que todas las partes se procesen a la vez (ecuación 4), que se excedan los límites de área (ecuación 5) y de altura (ecuación 6). Además, la restricción asociada a la ecuación 7 fuerza que la asignación de los trabajos a las máquinas se haga de manera ascendente y así impedir que se le asignen trabajos vacíos (que no contienen ninguna parte) a las máquinas. Y finalmente se tienen las restricciones asociadas a las ecuaciones 8, 9 y 10, que corresponden a los tiempos de finalización y de procesamiento de los distintos trabajos en las máquinas.

Este problema es NP-Hard (Nondeterministic Polynomial Time) según [13] por lo que se cree que una buena opción aplicar algoritmos heurísticos para generar buenas soluciones (sin tener la certeza de si son las óptimas) en tiempos computacionales cortos. Para ello se desarrolla la metaheurística descrita en el siguiente capítulo.

3 DESCRIPCIÓN DE LA METODOLOGÍA

El problema propuesto puede resolverse de dos modos, mediante una metaheurística en la cual se exploren diferentes soluciones con un criterio de búsqueda determinado, o haciendo uso de una aplicación que resuelva el modelo matemático del problema de manera exacta, como puede ser el programa CPLEX [15], de IBM®, como se demuestra en [1]. Para abordar la cuestión, se opta por elaborar una metaheurística debido a que resulta una herramienta mucho más rápida, modificable y polivalente como se puede observar en los siguientes capítulos del documento.

Una vez se ha introducido la técnica de fabricación aditiva y se ha descrito el problema al que va enfocado el proyecto en el bloque anterior, a continuación, se lleva a cabo una exposición del método de resolución utilizado. Para dichas explicaciones, el capítulo se divide en varias secciones en las cuales se aborda, en primer lugar, el problema completo, ofreciendo un resumen a modo de visión general de la estructura de la metaheurística y, en segundo lugar, se explica de manera detallada cada algoritmo usado en el programa y su finalidad.

3.1 Metaheurística aplicada

La metaheurística es del tipo ILS (Iterated Local Search), lo que significa que a la solución se llega explorando distintos tipos de vecindades (soluciones similares) a partir de una solución inicial generada de manera aleatoria. Una vez comenzadas las iteraciones, la solución va sufriendo perturbaciones durante el procedimiento heurístico, y se ejecuta una búsqueda en diferentes tipos de vecindarios (conjunto de soluciones que surgen de una modificación determinada de una solución) de la solución en busca de un mejor resultado de Makespan. Este tipo de metaheurísticas tienen un gran número de variantes, y en el presente trabajo se elabora con una caracterización peculiar, la inclusión de un algoritmo de autoaprendizaje denominado Q-Learning [16]. A continuación, se describe de manera general cada uno de los pasos que se dan en el proceso de cada iteración.

En primer lugar, se efectúa la fase de construcción, en la cual se produce una solución inicial de manera aleatoria que indica las partes que van asociadas a cada máquina, incluyendo la restricción de que, cada máquina debe tener asociada al menos una parte, además se ordenan las partes de menor a mayor altura considerándose esto una especie de regla de despacho cuya justificación se detalla en el siguiente párrafo.

En segundo lugar, se resuelve el problema de agrupación de partes en lotes (Problema de Empaquetado o Strip Packing Problem), el objetivo de dicho problema es lograr una agrupación de partes en lotes que maximicen el área ocupada dentro de los límites de la bandeja de entrada de la máquina. El propósito de dichas agrupaciones es minimizar el tiempo de preparación de la máquina y agrupar las piezas en conjuntos que contengan partes de alturas similares para así ahorrar el tiempo de aplicación de capas de polvo, que depende de la pieza que mayor altura tenga, de ahí el interés de ordenar en un inicio las partes de mayor a menor altura. Dentro del problema de empaquetado hay diferentes métodos para llevarse a cabo, pero en la metaheurística se trabaja solo con tres de ellos [17] que son: Next-Fit Decreasing Height, First-Fit Decreasing Height y Best-Fit Decreasing Height que se explican más adelante, en la sección 3.2 del presente capítulo, en detalle.

En tercer lugar, una vez se tiene un conjunto de trabajos, con sus correspondientes partes, asignados a una serie de máquinas, se procede a realizar la búsqueda de vecindario variable (Variable Neighborhood Search, VNS) con el objetivo de encontrar un vecino (solución muy parecida a la ya existente) que proporcione un resultado mejor que el que se tiene en el momento. En concreto, se aplica una técnica que usa tres tipos de exploraciones distintas [18], vecinos Flip, Swap y Job-Alloc, que se detallan en la sección 3.3.

Para conocer qué tipo de vecindario explorar en cada iteración se hace uso del algoritmo de autoaprendizaje Q-Learning, que proporciona cual es el mejor vecindario en función de los datos obtenidos en búsquedas locales anteriores. Dicho algoritmo se explica minuciosamente en la sección 3.4.

En cuarto lugar, una vez se obtiene el mejor vecino en la búsqueda local con la aportación del algoritmo de autoaprendizaje, se compara con la mejor solución obtenida en las iteraciones pasadas y a medida que se repite el proceso se guarda la mejor solución que se ha obtenido en algún momento.

Finalmente, para concluir el proceso iterativo, se implementa un criterio de parada que hace que el programa finalice cuando sea muy poco probable que la calidad de la solución supere a la mejor que ya se ha obtenido anteriormente, sopesando así la rentabilidad de seguir haciendo trabajar al programa en lugar de terminarlo. Dicho criterio de parada probabilístico se describe en detalle en la sección 3.6 del presente capítulo.

Es importante destacar, que, a medida que se realizan las iteraciones, con una probabilidad definida por el usuario, se cambia por completo la solución que se está explorando con el objetivo de evitar encontrar una solución que fuese un óptimo local del problema. A este proceso se le denomina perturbación de la solución, y su procedimiento se explica en la sección 3.5.

3.2 Problema de empaquetado

El primer paso que se realiza en la metaheurística es la resolución del problema de empaquetado. Tras la obtención de una solución inicial aleatoria que indica a que máquina va a ir dirigida cada parte, se debe llevar a cabo el proceso de agrupamiento de dichas piezas en lotes con el propósito de optimizar el tiempo de procesamiento en las máquinas.

Explicado de manera técnica, sea $I_m = \{i_1, i_2, \dots, i_h\}$ el conjunto de partes ordenadas de mayor a menor altura a reagrupar y m el subíndice que indica la máquina en la que se van a procesar. El problema de empaquetado, para una máquina, trata de crear un conjunto $J_m = \{j_1, j_2, \dots, j_k\}$ en el cual cada trabajo contiene un grupo de partes.

A modo de resumen, primero, las partes se dividen en tantos conjuntos como máquinas haya, proceso que se realiza en la fase de construcción, segundo, cada conjunto de partes a su vez se vuelve a agrupar en grupos más reducidos denominados trabajos respetando las restricciones de capacidad superficial que tiene la máquina. El segundo paso puede realizarse siguiendo distintos criterios, en el algoritmo desarrollado se implementan aquellos expuestos en [17] los cuales son: NFDH (Next-Fit Decreasing Height), FFDH (First-Fit Decreasing Height) y BFDH (Best-Fit Decreasing Height). Para que la metaheurística abarque más soluciones iniciales posibles se lleva a cabo la agrupación mediante los tres métodos en las distintas máquinas, se calcula el tiempo de finalización de la producción de las soluciones obtenidas por los tres sistemas y posteriormente se comparan los resultados para elegir el que mejor Makespan aporta, es decir, el mínimo. Se puede observar en la ilustración 4 el procedimiento a seguir de una manera más esquemática, dicho esquema está basado en la representación en [13].

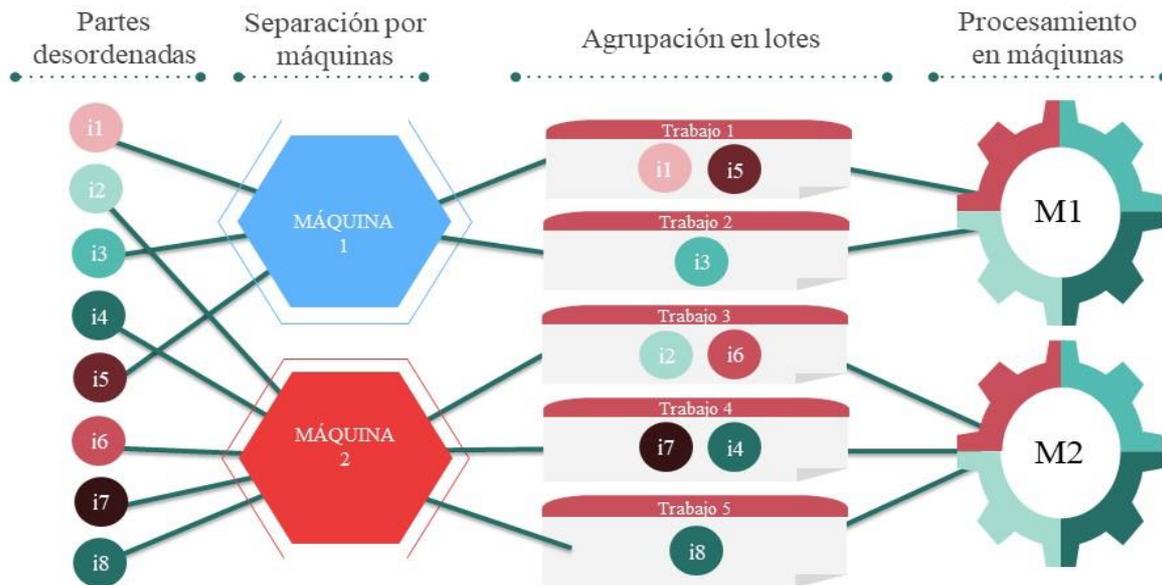


Ilustración 4. Esquema de problema de empaquetado. Elaboración propia.

Para comenzar a explicar en qué consisten los procedimientos que se llevan a cabo en los algoritmos correspondientes para lograr agrupar las piezas, es necesario comentar dos aspectos importantes. Por un lado, se comienza teniendo como entradas la máquina en la que se van a procesar las piezas y sus características, y un grupo de piezas I las cuales están ordenadas de mayor a menor según su altura. Por otro lado, se parte de la premisa de que todas las piezas pueden ser procesadas por el equipo en cuestión, es decir, no exceden ni la altura máxima permitida ni el área máxima de la bandeja de entrada de la máquina, aunque, si por alguna razón se tratase de procesar alguna pieza inadmisibles en la máquina, el algoritmo lo detectaría y eliminaría la parte en cuestión. También es importante destacar que inicialmente el grupo de trabajos J asociado a la máquina está vacío.

En primer lugar, se tiene el método NFDH. En este algoritmo se lee pieza a pieza el conjunto I , de manera que si la pieza i cabe en el trabajo inicial j , es decir, si el área de la pieza i más el área de las partes que ya están en el grupo j es menor que el área máxima permitida en la máquina, dicha parte va asociada al grupo j ; de lo contrario se crea un nuevo trabajo $j+1$ y se añade la pieza a dicho trabajo nuevo. Este algoritmo también está explicado en forma de diagrama en la ilustración 5.

En segundo lugar, se comenta la técnica FFDH. En este algoritmo se recorre el conjunto de piezas I de mayor a menor altura, pero esta vez se comprueba si la pieza i cabe en alguno de los trabajos ya existentes, es decir, ahora se evalúa tanto si la pieza cabe en el trabajo j como en todos los trabajos existentes j_k del conjunto J . Si la pieza en cuestión cabe en algún trabajo j_k se le asigna a dicho trabajo. Dicha comprobación se realiza de manera ascendente según el orden de creación de los trabajos. Si por el contrario la pieza no cabe en ninguno de los elementos de J ya creados, se crea un nuevo trabajo j y se le añade la pieza i . Este método se expresa de manera más visual mediante un diagrama en la ilustración 6.

En tercer lugar, se trata el procedimiento BFDH. Aquí las piezas se recorren de la misma manera que en el método anterior comprobando si la pieza i cabe en los grupos ya existentes de J , y en caso de que haya varios trabajos j_k que puedan aceptar la parte i , la pieza va destinada a aquel que ocupe mayor área al haber incluido la nueva parte sin exceder los límites de la máquina. Si la parte i no cabe en ninguno de los trabajos ya creados se elabora un nuevo trabajo. Al igual que para las dos técnicas anteriores, se ofrece un diagrama para explicar de una manera esquemática el algoritmo en la ilustración 7.

Para programar los algoritmos comentados, se parte de tantas variables auxiliares en forma de lista como máquinas haya, que corresponden a los conjuntos J_m vacíos. A medida que se van leyendo las partes de cada máquina, dichas listas se van rellenando con otras listas que corresponden a cada j_k , a modo de ejemplo, se tiene una variable por cada máquina del estilo:

$$\left[\underbrace{(i_1, i_2, i_3)}_{j_1}, \underbrace{(i_4, i_6)}_{j_2}, \underbrace{(i_5) \dots}_{j_3} \right]$$

Una vez se tiene el grupo de variables auxiliares comentado en el párrafo anterior se unen en otra lista, de manera que nuestra solución codificada tiene una forma tal que así:

$$\left[\left[\underbrace{(i_1, i_2, i_3)}_{j_1}, \underbrace{(i_4, i_6)}_{j_2}, \underbrace{(i_5) \dots}_{j_3} \right], \left[\underbrace{(i_8, i_7, i_9)}_{j_4}, \underbrace{(i_{10})}_{j_5}, \underbrace{(i_{11}, i_{12}) \dots}_{j_6} \right] \right]$$

$$\underbrace{\hspace{15em}}_{M_1} \quad \underbrace{\hspace{15em}}_{M_2}$$

Donde las partes i_k realmente, a la hora de programar, son un tipo de variable especial correspondiente a una clase que contiene todas las características de la pieza en cuestión.

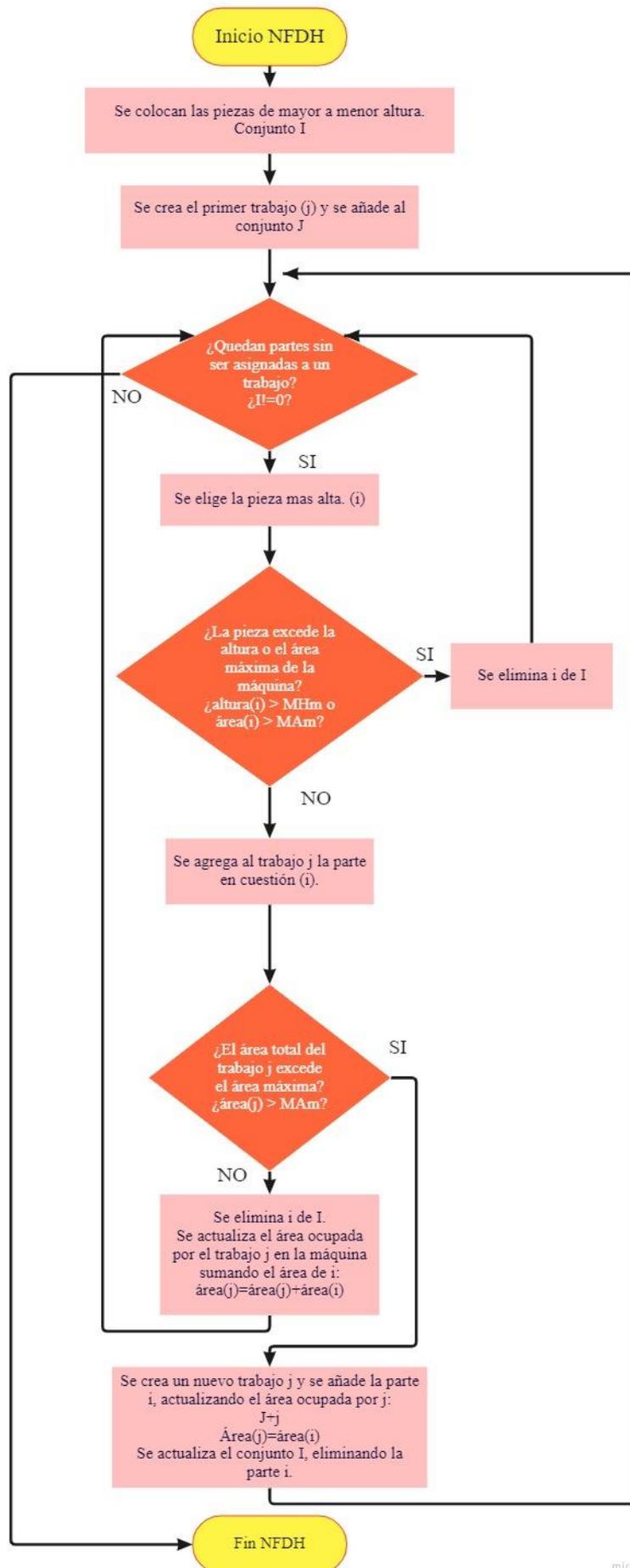


Ilustración 5. Diagrama del algoritmo NFDH. Elaboración propia.

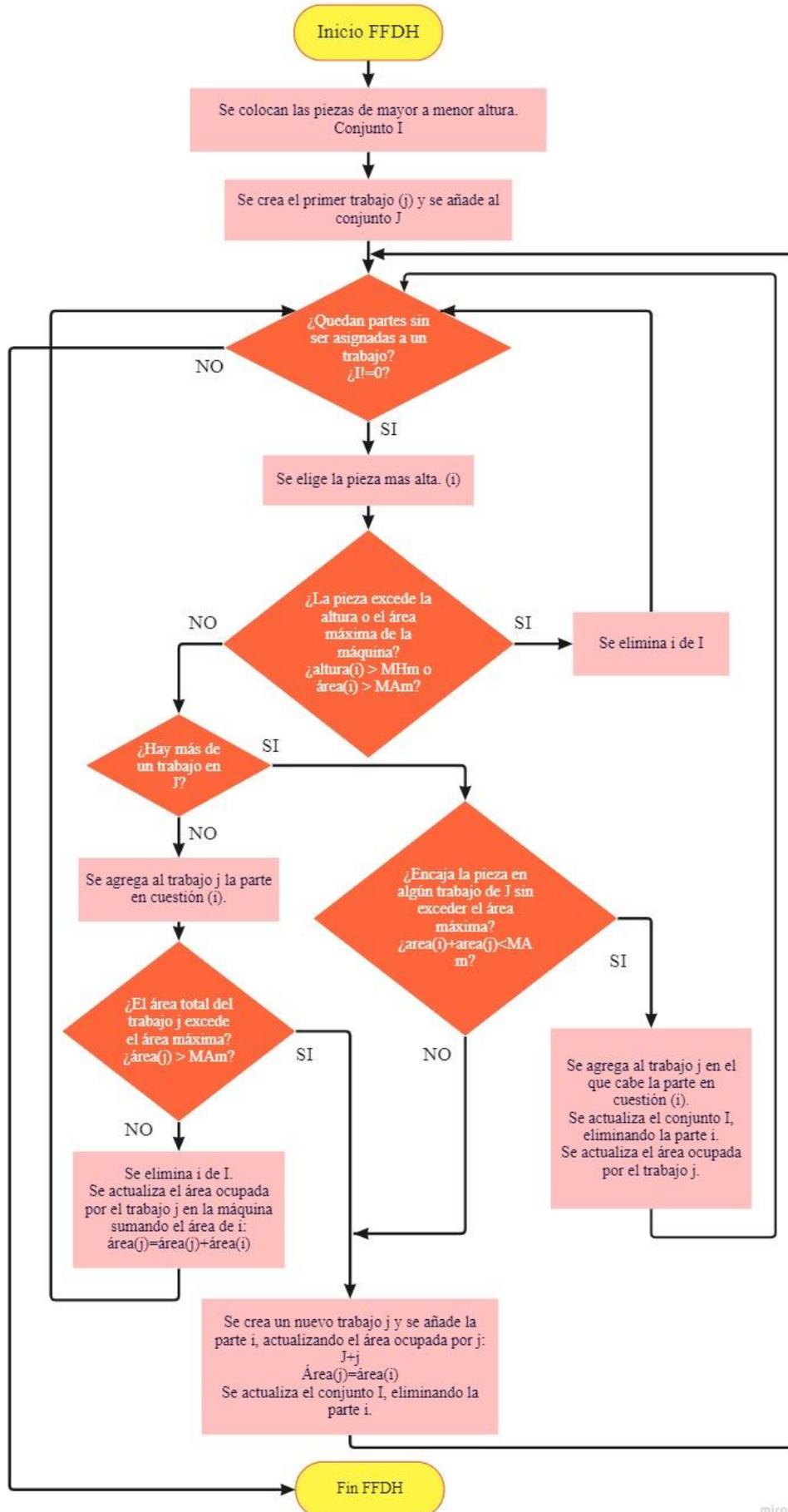


Ilustración 6. Diagrama del algoritmo FFDH. Elaboración propia.

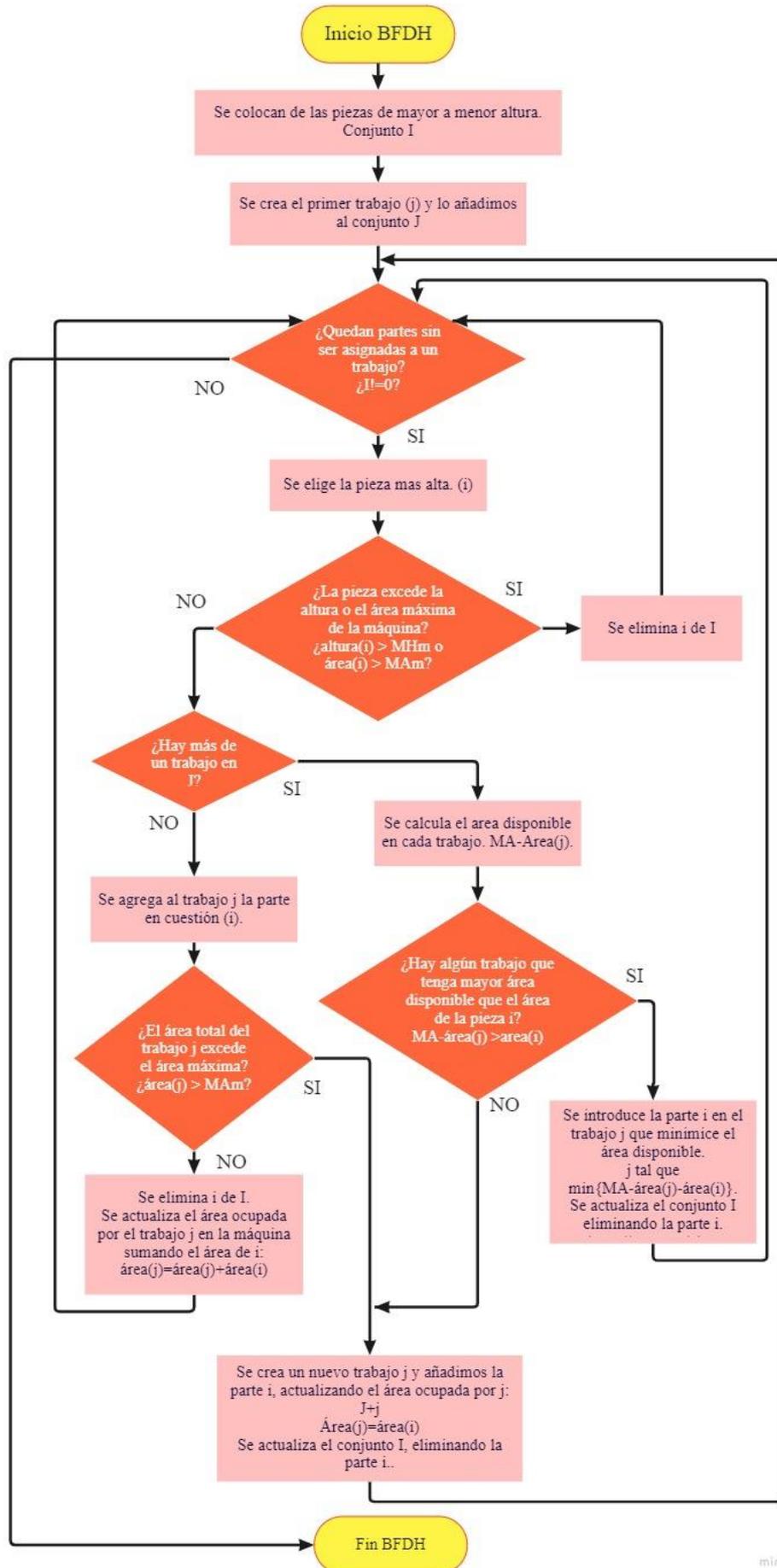


Ilustración 7. Diagrama del algoritmo BFDH. Elaboración propia.

3.3 Búsqueda de vecindario variable

Tras haber solucionado el problema de agrupamiento de piezas en grupos y haber obtenido una solución inicial, se procede a estudiar las opciones factibles similares a dicha solución, para ello se van a introducir los tres tipos de vecinos que se van a incluir en la metaheurística.

En primer lugar, se explica cómo funciona la vecindad de tipo Flip. En este método se traslada una parte de cualquier trabajo a otro, dando igual a que máquina pertenezcan los trabajos. Un ejemplo de la obtención de un vecino de este tipo es el correspondiente a la ilustración 8.

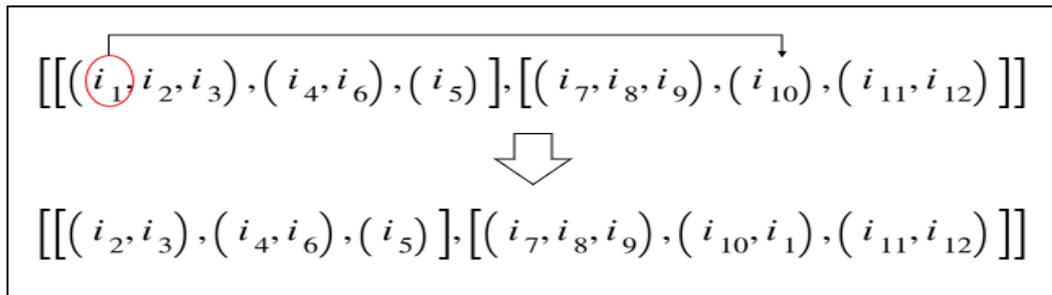


Ilustración 8. Ejemplo de vecino de tipo Flip. Elaboración propia.

En segundo lugar, se describe la vecindad de tipo Swap. Para formar un vecino de una solución mediante esta técnica, se escogen dos partes de dos trabajos distintos y se intercambian, dando igual la máquina a la que pertenezcan. Un ejemplo de este tipo sería el mostrado en la ilustración 9.

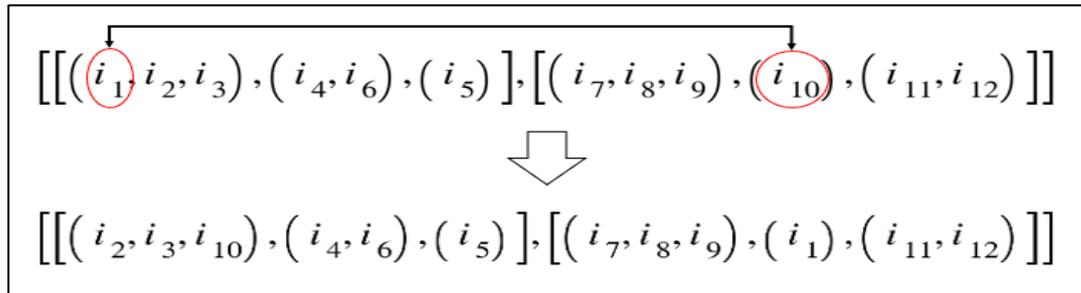


Ilustración 9. Ejemplo vecino tipo Swap. Elaboración Propia.

Finalmente se detalla el funcionamiento de la vecindad del tipo Job Alloc. En este caso se eligen dos partes, independientemente de en qué máquina se sitúen, y se trasladan a un nuevo trabajo, localizado en cualquiera de las máquinas que haya. En la ilustración 10 se puede observar un ejemplo de vecino de este tipo.

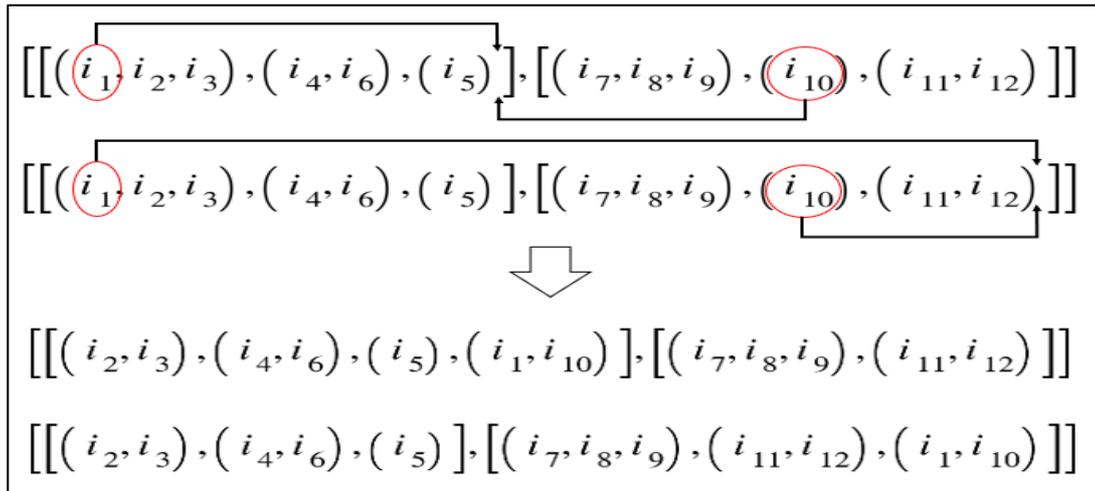


Ilustración 10. Ejemplo vecindad tipo Job Alloc 1. Elaboración propia.

Es necesario cuantificar la cantidad de vecinos que se obtienen dependiendo del tipo de vecindad para analizar en detalle las vecindades, por lo que se va a demostrar cuantos vecinos tiene una solución dependiendo de la técnica que se use para obtener nuevas soluciones similares.

En cuanto a la vecindad del tipo Flip se tiene un número total de vecinos igual a la ecuación 12 siendo n el número de partes de la instancia y k el número de trabajos de la solución. Esto se debe a que cada una de las partes puede ir destinada al resto de trabajos en los que no está situada.

$$n \cdot (k - 1) \quad (12)$$

La vecindad tipo Swap cuenta con tantos vecinos como se calculan en la ecuación 13 puesto que cada parte se puede intercambiar con el resto de las partes. Aunque hay vecinos irrelevantes debido a que cambiar partes dentro de un mismo trabajo no modifica la solución del problema.

$$n \cdot (n - 1) \quad (13)$$

La vecindad del tipo Job Alloc tiene tantos vecinos como se muestran en la ecuación 14, siendo m el número de máquinas al que puede ir destinado el nuevo trabajo y haciendo referencia el “2” de la ecuación a que se forman siempre parejas. En este caso no habrá vecinos que sean irrelevantes, porque eliminar dos partes de un mismo trabajo (cualesquiera de ellas) e introducirlas en uno nuevo influye en la solución.

$$\binom{n}{2} \cdot m. \quad (14)$$

3.4 Algoritmo de autoaprendizaje

Al tener distintos tipos de vecinos, es necesario implementar un algoritmo cuya función sea detectar qué tipo de vecindad es más rentable explorar, por lo tanto, se ha elegido el algoritmo Q-Learning [16]. Dicho algoritmo se basa en el autoaprendizaje y a medida que se van haciendo iteraciones indica que, con cierta probabilidad, con un tipo de vecindad se obtienen mejores resultados que con otra.

Para comprender el funcionamiento de dicho algoritmo se describen primero sus variables principales:

- Matriz de indicadores de mejora. Es una matriz cuadrada con un número de filas y columnas igual al número de tipos de vecindades que se tienen. El contenido de dicha matriz indica la mejora que se obtiene si vamos del estado i (fila) al estado j (columna), es decir, si se pasa de un tipo de vecindad a otra. Inicialmente todos los elementos de esta matriz son cero.
- $Q(i,j)$. Valor que ofrece una estimación del beneficio que se obtiene si se pasa de la vecindad i (fila) a la vecindad j (columna).
- Best_sol. Variable que contiene la mejor solución obtenida hasta el momento.

- A. Vector que indica qué tipos de vecindades pueden ser exploradas. Inicialmente todas sus componentes son 1, ya que todas las vecindades están disponibles.
- R. Diferencia entre el Makespan de Best_sol y el mejor vecino de la vecindad a.

El algoritmo está en funcionamiento siempre que haya vecindades que puedan mejorar la solución. Y se debe inicializar con un estado t igual a una vecindad que se considere oportuna. Cuando no haya vecindades disponibles, es decir, cuando el vector A esté vacío, el proceso de búsqueda de vecindario variable de la iteración i finaliza.

En primer lugar, dentro de la fila t , con probabilidad ε se elige una vecindad aleatoria (columna de la matriz de indicadores de mejora) de entre las que haya disponibles, y con probabilidad $1 - \varepsilon$ se selecciona aquella que contenga el máximo $Q(t, a)$. De manera que el estado a (la próxima vecindad para estudiar) depende de la probabilidad ε .

En segundo lugar, una vez seleccionada la vecindad a , se explora por completo y se elige aquel vecino que permita obtener una mejor función objetivo. Posteriormente se compara el valor de la función objetivo del vecino seleccionado con aquel que se tiene guardado con la mejor solución que se ha obtenido en las anteriores iteraciones (Best_sol).

En tercer lugar, si se mejora el valor de la función objetivo ($R < 0$), se guarda el vecino obtenido como la mejor solución en Best_sol y se procede a reiniciar el vector A , estableciendo todas sus componentes de nuevo a 1. Si por el contrario $R > 0$ se elimina la vecindad a de A haciendo 0 su correspondiente componente del vector A .

Finalmente, se establece el valor de a (vecindad que se ha explorado) a la variable t (vecindad en la que se sitúa), además se actualizan los valores de la matriz $Q(t, a)$ siguiendo la siguiente ecuación 15 :

$$Q(t, a) = Q(t, a) + \alpha[R + \gamma \cdot \max_{a' \in A} \{Q(a, a') - Q(t, a)\}] \quad (15)$$

Donde α es un parámetro entre 0 y 1 que indica la velocidad de aprendizaje y γ es una variable que indica el factor de descuento aplicado a la variable Q que acompaña en la fórmula.

El algoritmo de autoaprendizaje Q-Learning se explica de manera esquemática con un diagrama en la ilustración 11.

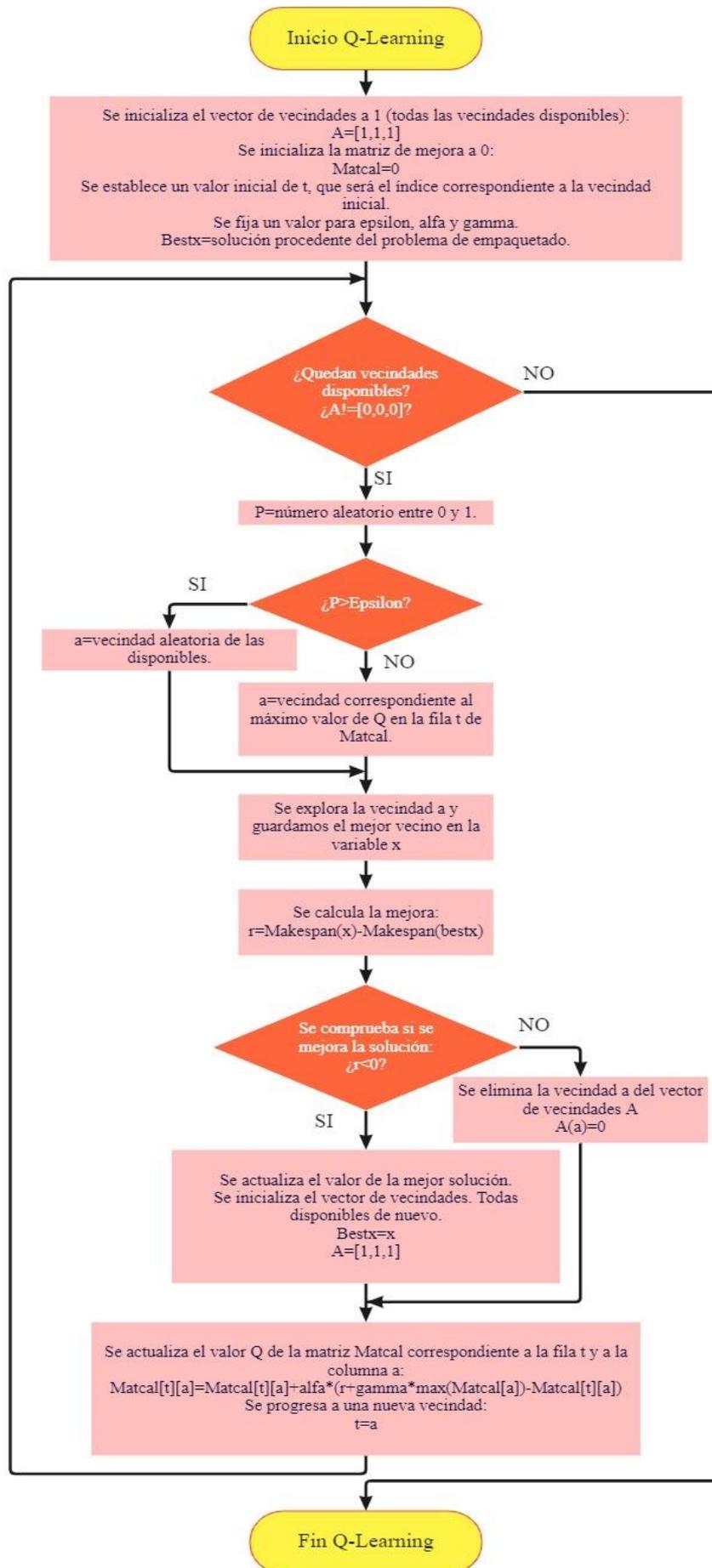


Ilustración 11. Diagrama de algoritmo Q-Learning. Elaboración propia.

3.5 Perturbación de la solución

A la hora de explorar distintas soluciones a partir de una solución inicial aleatoria mediante la búsqueda del mejor vecino posible usando el algoritmo de autoaprendizaje se corre el riesgo de encontrar una solución que sea un óptimo local, por lo tanto, es necesario modificar periódicamente, de una manera brusca, la solución con la que se está trabajando para abarcar así un mayor número de soluciones posibles.

Con el objetivo de alcanzar la mayor cantidad de soluciones exploradas, el algoritmo se encarga de elaborar una solución totalmente aleatoria de la misma manera en que se inicializa la metaheurística: se forma una solución previa que se encargue de distribuir las partes a máquinas, posteriormente se ordenan las partes según su altura y finalmente se lleva a cabo el problema de empaquetado para obtener la nueva solución distinta.

3.6 Criterio de parada del algoritmo

La mayoría de las metaheurísticas basan su criterio de parada en un número máximo de iteraciones del algoritmo implementado en cada caso, sin embargo, tomando como referencia [1], en la metaheurística desarrollada se implementa un criterio de parada probabilístico.

El criterio propuesto tiene la finalidad de ahorrar tiempos computacionales altos de manera innecesaria, por una parte, y de asegurar, por otra, que la solución a la que se ha llegado es lo suficientemente buena como para dejar de seguir buscando mejores soluciones. Para ello, se diseña una técnica [19] en la cual se detecta la capacidad del algoritmo de conseguir soluciones mejores a las ya exploradas y, cuando dicha posibilidad de mejorar la solución es muy baja, se deja de iterar.

El fundamento teórico [19] de dicho criterio de parada se basa en que los valores de la función objetivo del problema (Makespan) siguen una distribución desconocida, pero se trata de conocer su comportamiento mediante otra variable aleatoria. Esto es debido a que, siendo Z un conjunto de los valores de Makespan, se asume que, al estar minimizando dicho valor, la mayor parte de los valores de Makespan estarán en el intervalo $[\mu, \max(Z)]$, siendo μ la media de dicho conjunto. Por lo tanto, se comprueba en [19] que, formando la variable aleatoria presentada en la ecuación 16, se comprueba que sigue una distribución probabilística positiva con una cola a la izquierda, asumiéndose que es una distribución Gamma de parámetros desconocidos (θ y k).

$$\hat{Z} = \max(Z) - z \quad (16)$$

Entonces, para calcular la probabilidad de que en una iteración se encuentre una solución mejor (z) a la aquella que ha aportado mejor resultado de todas las que se han explorado en todo el proceso anterior (z^*), se usa un conjunto de valores de z para estimar los parámetros de la función Gamma y posteriormente, a partir de dicha solución, la probabilidad de obtener una mejor solución sería:

$$\begin{aligned} p &= P[z < z^*] = P[\max(Z) - \hat{Z} \leq z^*] = P[\hat{Z} \geq \max(Z) - z^*] = 1 - P[\hat{Z} \leq \max(Z) - z^*] \\ &= 1 - F_{\hat{Z}}(\max(Z) - z^*) \quad (17) \end{aligned}$$

Por lo tanto, el procedimiento en el algoritmo es el siguiente:

Primero, se guarda un conjunto de valores de Makespan en Z correspondientes a un número predefinido de iteraciones q . Una vez que se completa el conjunto Z se construye el conjunto \hat{Z} cuyas componentes vendrán dadas por la ecuación 16. Tras formar el conjunto \hat{Z} se realiza una estimación de los parámetros de la distribución Gamma (θ y k) haciendo uso de los valores de dicho conjunto. Y finalmente, en cada iteración posterior a las q primeras en la cual se mejore la solución, se calcula la probabilidad de mejorar la solución en la siguiente iteración haciendo uso de la ecuación 17, tal que, $p = 1 - F_{\hat{Z}}(\max(Z) - z^*)$. De modo que, si al comienzo de la iteración la probabilidad de mejora es menor que el umbral que se establezca (τ), el algoritmo finalizará.

A parte del criterio de parada probabilístico se impone un tiempo computacional máximo. Esto se debe a que las instancias son diferentes, por lo que unas necesitan un tiempo más elevado que otras para hacer una iteración a consecuencia de la cantidad de máquinas y partes que se tienen en cada problema. Los grupos de instancias con sus características y su procedencia se explican detalladamente en la sección 4.2. Entonces, en aquellas instancias en las que se detecta que se requiere un tiempo alto para realizar una iteración (entre 10 y 20 segundos) se establece un límite de tiempo de 600 segundos que corresponden con el grupo R de instancias. En aquellas instancias que se requieren entre 5 y 10 segundos para realizar una iteración se establece un tope de 300 segundos. Y finalmente, para aquellas instancias que son más sencillas y tienen un número de partes y máquinas

bajo se establece un límite de 60 segundos. Estos dos últimos grupos de instancias contienen instancias tanto de los grupos B y C.

3.7 Resumen de la metaheurística

Con intención de aclarar todo el proceso, se muestra en la ilustración 12 un diagrama en el cual se muestra el transcurso que conlleva la metaheurística de principio a fin, en el cual aparecen todos los conceptos que se han comentado en las diferentes secciones del presente capítulo.

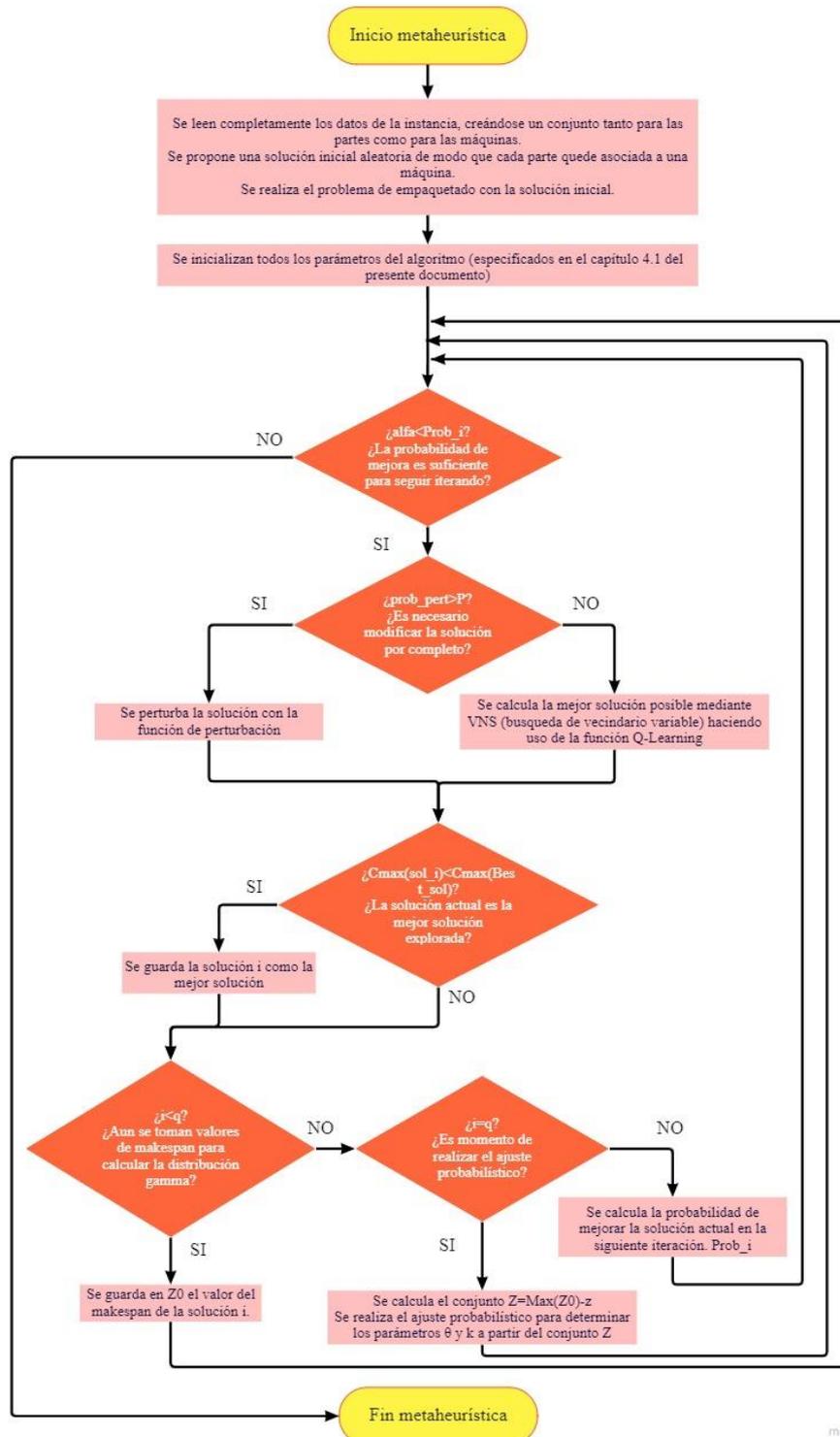


Ilustración 12. Metaheurística completa. Elaboración propia.

4 IMPLEMENTACIÓN DE LA METAHEURÍSTICA

En el presente capítulo se expone cómo funciona el algoritmo resolviendo un problema de ejemplo. Para ello, se detallan en las primeras secciones del capítulo los parámetros de gestión seleccionados para poner en marcha el programa y el modo de obtención de estos, además de los datos de entrada que tendría el problema, es decir, las características de las piezas y las máquinas. Posteriormente, para explicar cómo funciona la metaheurística, se expone un caso práctico a modo de ejemplo donde se puede ver el comportamiento de todos los algoritmos implementados.

4.1 Equipo y herramientas utilizados

Los algoritmos han sido implementados en lenguaje Python, con la herramienta de entorno de desarrollo integrado Spyder® [20]. La computadora donde se ha programado y se han puesto en marcha los experimentos es un Acer Aspire A515-51 con un procesador Intel™ Core™ i7-8550U CPU @ 1.80GHz, 1992 Mhz con 4 procesadores principales y 8 procesadores lógicos con 12 GB de memoria RAM. Además, se ha tomado como referencia a la hora de la programación en lenguaje Python la librería Scheptk [4], que también ha sido utilizada a la hora de la representación de las soluciones en diagramas de Gantt.

4.2 Datos de entrada del problema

Para ajustar los parámetros de gestión del algoritmo se ha tomado como referencia la configuración explicada en [1] que está basada en [21]. Para la configuración de estos se han utilizado 25 instancias aleatorias del conjunto con un criterio de parada de 60 segundos de tiempo computacional.

En la tabla 1 se explican los parámetros y los valores usados para practicar con la metaheurística.

Tabla 1. Configuración de parámetros.

Parámetro	Significado	Valor
ρ	Probabilidad de que se ponga en acción la función de perturbación.	0.35
ϵ	Probabilidad de seleccionar una vecindad de manera aleatoria en el algoritmo de autoaprendizaje.	0.11
α	Parámetro que indica la velocidad de aprendizaje. Explicado en la sección 3.4.	0.88
γ	Factor de descuento. Explicado en la sección 3.4.	0.86
τ	Probabilidad que se debe superar para que pare el algoritmo. Explicado en la sección 3.6.	0.20

En cuanto a lo referente a la información del entorno, se toman las instancias usadas por los autores de [1]. Dicha batería de instancias se encuentra clasificada en 4 partes según el tipo de entorno. Aquellas que comienzan por B [13] se asocian a un entorno de 2 o 3 máquinas paralelas con idénticas especificaciones y entre 15 y 46 partes. Las que comienzan por C [13] tienen el mismo número de partes y máquinas que las de tipo B, pero con diferentes especificaciones en las máquinas. Finalmente, el grupo que comienza con la letra R tiene entre 5 y 19 máquinas de diferentes especificaciones y de 21 a 48 partes [1]. A esta batería de instancias se puede acceder mediante el enlace situado en la sección 4.3.

4.3 Algoritmo e instancias

Se proporciona el algoritmo desarrollado con sus correspondientes variantes, explicadas en el capítulo 5 de este documento, al igual que la batería de instancias utilizadas en el proyecto. El archivo que contiene todos estos documentos se encuentra en el siguiente enlace:

https://1drv.ms/u/s!AuD_dWbIcGyKguE0ZI9wP3-kuuw2aw?e=dYMyRg

El archivo en cuestión contiene dos carpetas, la denominada “dataset” contiene todas aquellas instancias utilizadas en el desarrollo de los resultados presentados en el presente documento, y aquella denominada “programs” contiene todos aquellos documentos referentes a los códigos escritos en lenguaje Python relacionados, con las diferentes variantes y la metaheurística elaborada. Además, para cada variante, se encuentran subcarpetas las cuales contienen, por un lado, los programas usados para su desarrollo con el nombre “ALGORITMO_FINAL_METAHEURÍSTICAX”, por otro lado, aquellos usados para calcular los resultados denominados “ALGORITMO_FINAL_METAHEURÍSTICAX-RESULTADOS” y también otros programas usados para comprobar el funcionamiento de las funciones modificadas en cada variante, llamados “ALGORITMO_FINAL_METAHEURÍSTICAX_PRUEBAX”.

4.4 Ejemplo numérico

Una vez se conoce todo lo necesario para solucionar el problema de programación de operaciones se explica el procedimiento expuesto en el bloque 3 de manera numérica con una instancia denominada C-P39 en la batería explicada en el punto 4.1. En las tablas 2 y 3 se muestran tanto las características de las máquinas como las de las partes del problema a resolver.

Tabla 2. Máquinas del ejemplo numérico.

Nombre	VTm (Cm ³ /h)	HTm (Cm/h)	STm (h)	MAm (Cm ²)	MHm (Cm)
M1	0.030864	0.7	1.2	800	40
M2	0.030864	0.7	1.0	1000	40

Tabla 3. Partes del ejemplo numérico.

Nombre	Altura (Cm)	Área (Cm ²)	Volumen (Cm ³)
P1	4.27	122.62	102.83
P2	2.18	178.34	214.79
P3	29.58	273.83	840.17
P4	18.99	89.68	683.06
P5	10.77	269.75	1928.6
P6	26.67	258.54	1375.9
P7	14.38	114.56	989.53
P8	3.5	454.89	683.48
P9	3.0	615.12	722.91
P10	17.04	99.53	703.08
P11	27.94	56.85	272.92
P12	17.38	50.02	125.7
P13	17.61	146.57	617.98
P14	9.18	728.33	1193.36
P15	13.22	179.75	641.7

4.4.1 Ejemplo numérico. Problema de empaquetado

Para comenzar con el problema en cuestión, se propone una solución inicial generada aleatoriamente, que, como se comenta en el segundo párrafo de la sección 3.1, consta de una asociación de partes a máquinas. Dicha solución, en nuestro ejemplo será [2,1,2,1,1,2,2,2,1,2,1,2,1,1,2], lo que significa que la parte P1 va asignada a la máquina M2, la parte P2 se procesa en la máquina M1 y de igual manera con las partes que restan.

Una vez se tiene la solución inicial, se pone en marcha el algoritmo relacionado con el problema de empaquetado, resolviendo los problemas NFDH, FFDH y BFDH y seleccionando aquel que proporcione el menor Makespan. Las soluciones para dichos problemas son las expuestas en las tablas 4, 5 y 6, además, el procedimiento para llegar a dichos resultados se puede observar en el anexo A.

Tabla 4. Ejemplo numérico. Solución del problema de empaquetado NFDH.

Máquina 1		Máquina 2	
Trabajo	Partes	Trabajo	Partes
J1	P11, P4, P13, P5	J4	P3, P6, P10, P12, P15, P7
J2	P14	J5	P1, P8
J3	P9, P2		

Tabla 5. Ejemplo numérico. Solución del problema de empaquetado FFDH.

Máquina 1		Máquina 2	
Trabajo	Partes	Trabajo	Partes
J1	P11, P4, P13, P5, P2	J4	P3, P6, P10, P12, P15, P7
J2	P14	J5	P1, P8
J3	P9		

Tabla 6. Ejemplo numérico. Solución del problema de empaquetado BFDH.

Máquina 1		Máquina 2	
Trabajo	Partes	Trabajo	Partes
J1	P11, P4, P13, P5	J4	P3, P6, P10, P12, P15, P7
J2	P14	J5	P1, P8
J3	P9, P2		

Una vez que se tienen las soluciones de los 3 tipos de modos de empaquetado, se calcula el Makespan de las mismas, cuyos resultados aparecen en la tabla 7, para compararlas y seleccionar aquella que sea mejor, es decir, la que proporcione un menor tiempo de terminación total. Para ello se hace uso de la ecuación 1:

$$PT_j = SET_m \cdot Z_{mj} + VT_m \cdot \sum_{i \in I} v_i \cdot X_{mij} + HT_m \cdot \max\{h_i \cdot X_{mij}\} \quad (1)$$

Tabla 7. Ejemplo numérico. Tiempos de terminación de problema de empaquetado.

Solución	Makespan (h)
Solución NFDH	205.56
Solución FFDH	205.56
Solución BFDH	205.56

Por lo tanto, en base a los resultados expuestos en la tabla 7, el algoritmo se podría decantar por cualquiera de las soluciones calculadas mediante los distintos métodos de empaquetado. En el programa propuesto, en caso de que los valores del Makespan fuesen iguales, se elige de manera descendente según la tabla 7, por lo que, en este caso la solución que se escoge es la proporcionada mediante el algoritmo NFDH.

4.4.2 Ejemplo numérico. Vecindades y búsqueda de vecindario

A partir de la solución obtenida mediante el problema de empaquetado, se muestran los resultados tras llevar a cabo la búsqueda de vecindario de cada tipo de vecindad, explicadas en la sección 3.3.

Se comienza mostrando los resultados factibles de la búsqueda en la vecindad de tipo Flip en la tabla 8. Como se comenta en la sección 3.3, concretamente en la ecuación 12, en la vecindad de este tipo existen un total de $n \cdot (k - 1)$ vecinos, por lo que, en este caso, al haber 5 trabajos y 15 partes hay 60 vecinos, aunque esto no quiera decir que todos sean soluciones factibles. La condición de si una solución es factible o no la marca el área ocupada en cada trabajo, si el área de un trabajo pasa a exceder el área máxima de la máquina, entonces el vecino es no factible.

Tabla 8. Ejemplo numérico. Vecindad de tipo Flip.

Máquina 1			Máquina 2		Makespan (h)
Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5	
P11, P4, P13, P5	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8	205.56
P4, P13, P5	P14, P11	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8	212.42
P4, P13, P5	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8, P11	219.27
P11, P13, P5	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8, P4	225.67
P11, P4, P5	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8, P13	222.69
P11, P4, P13	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8, P5	258.36
P11, P4, P13, P5, P2	P14	P9	P3, P6, P10, P12, P15, P7	P1, P8	205.56
P11, P4, P13, P5	P14	P9	P3, P6, P10, P12, P15, P7	P1, P8, P2	200.92
P11, P4, P13, P5	P14	P9, P2	P6, P10, P12, P15, P7	P1, P8, P3	209.96
P11, P4, P13, P5	P14	P9, P2	P3, P10, P12, P15, P7	P1, P8, P6	209.96
P11, P4, P13, P5, P10	P14	P9, P2	P3, P6, P12, P15, P7	P1, P8	209.44
P11, P4, P13, P5	P14, P10	P9, P2	P3, P6, P12, P15, P7	P1, P8	215.18
P11, P4, P13, P5	P14	P9, P2	P3, P6, P12, P15, P7	P1, P8, P10	205.56
P11, P4, P13, P5, P12	P14	P9, P2	P3, P6, P10, P15, P7	P1, P8	227.26
P11, P4, P13, P5	P14	P9, P2	P3, P6, P10, P15, P7	P1, P8, P12	205.56
P11, P4, P13, P5, P15	P14	P9, P2	P3, P6, P10, P12, P7	P1, P8	236.10
P11, P4, P13, P5	P14	P9, P2	P3, P6, P10, P12, P7	P1, P8, P15	205.56
P11, P4, P13, P5, P7	P14	P9, P2	P3, P6, P10, P12, P15	P1, P8	225.37
P11, P4, P13, P5	P14	P9, P2	P3, P6, P10, P12, P15	P1, P8, P7	205.56
P11, P4, P13, P5, P1	P14	P9, P2	P3, P6, P10, P12, P15, P7	P8	208.73

Como se puede observar, el vecino del cual se obtiene mejor solución es aquel que proporciona un Makespan de 200.92 h, situado en la fila 10 de la tabla 8.

En segundo lugar, se exponen los resultados obtenidos de explorar la vecindad de tipo Swap en la tabla 9. En el caso del presente ejemplo se hallan un total de 164 vecinos, de los cuales 66 son factibles. Debido al gran número de soluciones factibles en este tipo de vecindad se muestran en la tabla 9 solo las 20 primeras soluciones factibles.

Tabla 9. Ejemplo numérico. Vecindad de tipo Swap.

Máquina 1			Máquina 2		Makespan (h)
Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5	
P11, P4, P13, P5	P14	P9, P2	P3, P6, P10, P12, P15, P7	P1, P8	205.56
P4, P13, P5, P2	P14	P9, P11	P3, P6, P10, P12, P15, P7	P1, P8	216.75
P4, P13, P5, P3	P14	P9, P2	P6, P10, P12, P15, P7, P11	P1, P8	224.21
P4, P13, P5, P6	P14	P9, P2	P3, P10, P12, P15, P7, P11	P1, P8	238.71
P4, P13, P5, P10	P14	P9, P2	P3, P6, P12, P15, P7, P11	P1, P8	198.83
P4, P13, P5, P12	P14	P9, P2	P3, P6, P10, P15, P7, P11	P1, P8	212.57
P4, P13, P5, P15	P14	P9, P2	P3, P6, P10, P12, P7, P11	P1, P8	221.41
P4, P13, P5, P7	P14	P9, P2	P3, P6, P10, P12, P15, P11	P1, P8	210.68
P4, P13, P5, P1	P14	P9, P2	P3, P6, P10, P12, P15, P7	P8, P11	216.10
P11, P13, P5, P2	P14	P9, P4	P3, P6, P10, P12, P15, P7	P1, P8	216.75
P11, P13, P5, P3	P14	P9, P2	P6, P10, P12, P15, P7, P4	P1, P8	211.56
P11, P13, P5, P6	P14	P9, P2	P3, P10, P12, P15, P7, P4	P1, P8	226.94
P11, P13, P5, P12	P14	P9, P2	P3, P6, P10, P15, P7, P4	P1, P8	206.17
P11, P13, P5, P15	P14	P9, P2	P3, P6, P10, P12, P7, P4	P1, P8	215.02
P11, P13, P5, P7	P14	P9, P2	P3, P6, P10, P12, P15, P4	P1, P8	204.28
P11, P13, P5, P1	P14	P9, P2	P3, P6, P10, P12, P15, P7	P8, P4	222.49
P11, P4, P5, P2	P14	P9, P13	P3, P6, P10, P12, P15, P7	P1, P8	215.79
P11, P4, P5, P3	P14	P9, P2	P6, P10, P12, P15, P7, P13	P1, P8	213.57
P11, P4, P5, P6	P14	P9, P2	P3, , P10, P12, P15, P7, P13	P1, P8	228.95
P11, P4, P5, P7	P14	P9, P2	P3, P6, P10, P12, P15, P13	P1, P8	206.29

Tras explorar por completo la vecindad de tipo swap, se encuentra que el mejor vecino es el situado en la fila 5 de la tabla 9, el cual proporciona un Makespan de 198.83 h.

Finalmente, se exponen las soluciones obtenidas mediante la exploración de la vecindad de tipo Job Alloc. Ésta, como se explica en la sección 3.3, tiene tantos vecinos como se muestran en la ecuación 14 por lo que en el caso del ejemplo numérico se tienen un total de 420 vecinos, de los cuales 380 de ellos son factibles. En la tabla 10, se pueden observar 19 vecinos factibles de este tipo.

Tabla 10. Ejemplo numérico. Vecindad de tipo Job Alloc.

Máquina 1				Máquina 2			Makespan (h)
Trabajo 1	Trabajo 2	Trabajo 3	Trabajo aux1	Trabajo 4	Trabajo 5	Trabajo aux2	
P11, P4, P13, P5	P14	P9, P2		P3, P6, P10, P12, P15, P7	P1, P8		205.56
P11, P4, P13, P5	P14	P9, P2		P3, P10, P12, P15, P7	P1	P8, P6	213.95
P4, P13, P5	P14	P9, P2	P10, P11	P3, P6, P12, P15, P7	P1, P8		223.93
P11, P13, P5	P14	P9, P2	P10, P4	P3, P6, P12, P15, P7	P1, P8		223.93
P11, P13, P5	P14	P9, P2		P3, P6, P12, P15, P7	P1, P8	P10, P4	229.66
P11, P4, P5	P14	P9, P2	P13, P10	P3, P6, P12, P15, P7	P1, P8		222.97
P11, P4, P5	P14	P9, P2		P3, P6, P12, P15, P7	P1, P8	P13, P10	226.686
P11, P4, P13	P14	P9, P2	P10, P5	P3, P6, P12, P15, P7	P1, P8		222.80
P11, P4, P13	P14	P9, P2		P3, P6, P12, P15, P7	P1, P8	P10, P5	266.97
P11, P4, P13, P5		P2, P9	P10, P14	P3, P6, P12, P15, P7	P1, P8		216.38
P11, P4, P13, P5		P2, P9		P3, P6, P12, P15, P7	P1, P8	P10, P14	244.28
P11, P4, P13, P5	P14	P2	P10, P9	P3, P6, P12, P15, P7	P1, P8		222.23
P11, P4, P13, P5	P14	P2		P3, P6, P12, P15, P7	P1, P8	P10, P9	229.76
P11, P4, P13, P5	P14	P9	P10, P2	P3, P6, P12, P15, P7	P1, P8		222.80
P11, P4, P13, P5	P14	P9		P3, P6, P12, P15, P7	P1, P8	P10, P2	214.08
P11, P4, P13, P5	P14	P9, P2	P10, P3	P6, P12, P15, P7	P1, P8		257.27
P11, P4, P13, P5	P14	P9, P2		P6, P12, P15, P7	P1, P8	P10, P3	213.96
P11, P4, P13, P5	P14	P9, P2	P10, P6	P3, P12, P15, P7	P1, P8		271.77
P11, P4, P13, P5	P14	P9, P2		P3, P12, P15, P7	P1, P8	P10, P6	213.96
P11, P4, P13, P5	P14	P9, P2	P10, P12	P3, P6, P15, P7	P1, P8		244.50
P11, P4, P13, P5	P14	P9		P3, P6, P10, P12, P15, P7	P8	P2, P1	204.37

Es importante recalcar que los trabajos que están vacíos, a la hora de programar, no se consideran como tales, es decir, habría un trabajo menos en la solución. Por ejemplo, en los vecinos de la fila 10 y 11 solo habría 5 trabajos, en el caso del vecino de la fila 10, por ejemplo, 3 trabajos en la primera máquina y 2 en la segunda.

Tras explorar la vecindad al completo, se consigue un vecino que ofrece un Makespan de 204.37 h, el cual es el

último que aparece en la tabla 10.

4.4.3 Ejemplo numérico. Resultados

Tras poner en funcionamiento el programa al completo, el mejor valor de Makespan obtenido tras 43 segundos de tiempo computacional y 200 iteraciones es 195.435 h. Dicho valor del Makespan resulta de la solución contenida en la tabla 11.

Tabla 11. Ejemplo numérico. Resultados.

Máquina 1		Máquina 2	
Trabajo	Partes	Trabajo	Partes
J1	P15, P7, P5	J4	P11, P6, P4, P13, P12, P3, P10
J2	P14	J5	P2, P8
J3	P9, P1		

El diagrama de Gantt correspondiente a la solución obtenida en el entorno de Unrelated Machines en el que se encuentra es el que aparece en la imagen siguiente (ilustración 13).

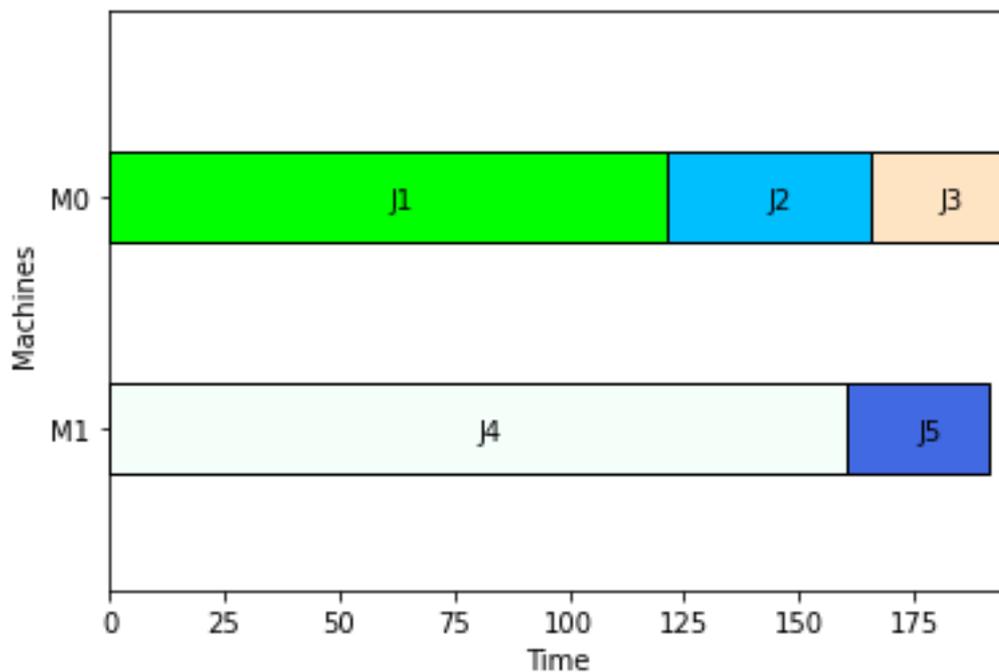


Ilustración 13. Diagrama de Gantt del ejemplo numérico. Elaboración propia.

5 VARIANTES PROPUESTAS

Una vez analizados los resultados obtenidos con la metaheurística, incluidos en el anexo B, teniendo como referencia los proporcionados en [1], se observa que los tiempos computacionales son altos, mientras que las soluciones obtenidas son relativamente semejantes, tanto para instancias de tipo pequeño como de tipo grande (dicha caracterización de instancias se explica en el segundo párrafo del capítulo 6). Los resultados se estudian de una manera más exhaustiva en el capítulo 6 de este documento, en concreto en la sección 6.1.

Teniendo la metaheurística confeccionada como base, se intenta reducir el tiempo computacional renunciando a una solución más cercana al óptimo. Para conseguir esto, se pretende reducir el número de vecinos investigados mediante una modificación de los tres tipos de vecindades haciendo uso de reglas de despacho cuya funcionalidad es estudiada con la obtención de los resultados para cada una de las variantes y la posterior comparación con los calculados con el algoritmo base.

5.1 Variante 1. Modificación 1 de la vecindad “Flip”

Con el objetivo de reducir la cantidad de vecinos de tipo Flip sin que la solución final se vea en gran medida afectada, dicha vecindad se modifica de manera que, a diferencia de la original, en la cual, como se explica en la sección 3.3, una parte de un trabajo se traslada a otro trabajo cualquiera, independientemente de la máquina, ahora dicha parte va destinada al trabajo con menor número de partes de cada máquina. En el caso en el que en una máquina haya dos o más trabajos con el mismo número de partes y este resulte ser el mínimo, la parte puede ir destinada a cada uno de estos trabajos, es decir, se exploran, por cada parte, tantos vecinos como trabajos con ese número mínimo de partes se tengan.

Dicha regla de despacho se ha fundamentado en que cambiar una parte de un trabajo a otro dentro de una misma máquina tiene poca repercusión en el tiempo de procesamiento. Por lo tanto, se trata de focalizar dicha vecindad en estudiar en que trabajo sería más rentable situar la parte dentro de una máquina, para así, reducir el número de vecinos de este tipo haciendo que cada parte pueda trasladarse únicamente a un trabajo de cada máquina. La conclusión a la que se llega es que lo más lógico es hacer uso de los trabajos con menor número de partes, ya que éstos, en su mayoría, serán aquellos que tengan menor volumen ocupado, lo que significa que la parte va a aquel trabajo con menor tiempo de procesamiento asociado al invertido en el volumen total de las piezas. Para explicar de un modo más esquemático el cambio de la vecindad se utiliza un ejemplo.

En una solución formada por tres grupos de trabajos asociado cada uno a una máquina (ilustración 14) se tiene una cantidad total de tres vecinos por parte, lo que significa que el total de vecinos es de 48, que resulta ser $n \cdot m$ siendo n el número de partes y m el número de máquinas, esto se debe a que solo se le puede añadir la parte correspondiente a un trabajo por máquina (el de menor número de partes). Por el contrario, con la vecindad de tipo Flip original se tiene una cantidad de vecinos bastante mayor, ya que cada parte tendría la libertad de ser añadida a cualquier trabajo. El número total de vecinos para el ejemplo es de 112, correspondiente a la multiplicación del número de partes por el número de trabajos a los que pueden ir añadidas. Por lo tanto, en un

problema en el cual se tenga una solución como la del ejemplo (ilustración 14), se reduce en gran medida el tiempo invertido en explorar la vecindad de tipo swap, lo que conlleva a un ahorro de tiempo computacional y de trabajo del equipo.

En la ilustración 14 se muestra el funcionamiento de la vecindad una vez introducida la modificación, por ello, la parte i_1 solamente tendría tres destinos posibles, el trabajo 3, el trabajo 5 y el trabajo 8, aquellos con menor número de partes en cada máquina.

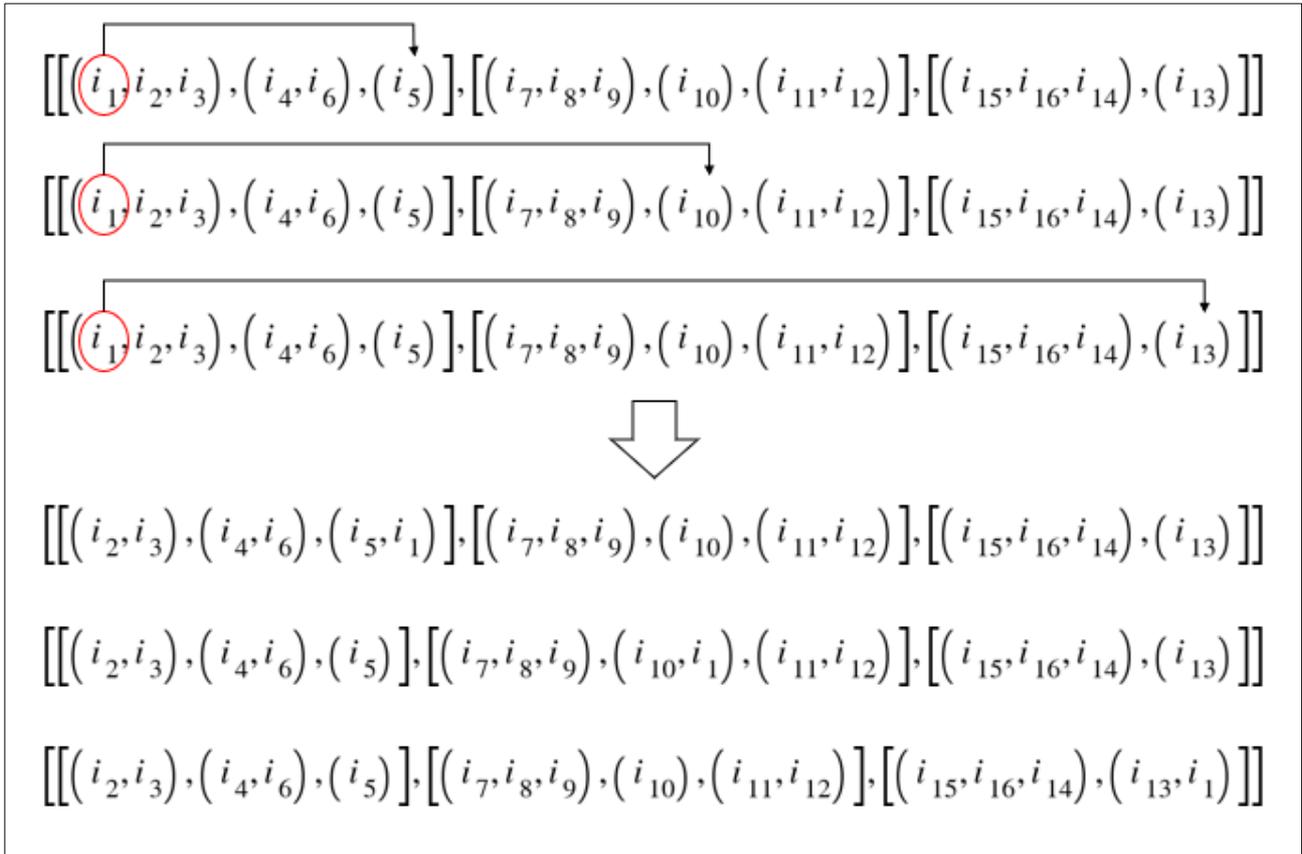


Ilustración 14. Modificación de la vecindad Flip. Elaboración propia.

5.2 Variante 2. Modificación 2 de la vecindad “Flip”

Por el mismo motivo que se explica en el primer párrafo de la sección 5.1, se lleva a cabo una segunda modificación de la vecindad de tipo Flip. Se trata en este caso, a diferencia del anterior, de situar la parte intercambiada de trabajos en aquel trabajo que tenga menor área ocupada para así, de una manera distinta, intentar situar la parte en aquellos trabajos que tengan el menor volumen ocupado.

Con esta variación de la vecindad se consigue reducir el número de vecinos de la misma manera que en la variante explicada en el punto 5.1 ya que también se tienen tantos vecinos por parte como máquinas haya, es decir, $n \cdot m$ vecinos. Por lo tanto, puede hacerse uso del ejemplo explicado al final la sección 5.1 y de la ilustración 14 para explicar por qué se produce una reducción importante en la cantidad de vecinos explorados. Para hacer uso de la figura 14, se debe suponer que los trabajos 3, 5 y 8 son aquellos que menor área ocupada tienen, ya que serán los únicos trabajos posibles a los que pueden ir destinadas las piezas.

5.3 Variante 3. Modificación de la vecindad “Swap”

Para reducir el número de soluciones exploradas en la búsqueda de vecindario variable, también se modifica la vecindad de tipo Swap. Dicha vecindad, originalmente, intercambia dos partes de distintos trabajos independientemente de la máquina a la que estén asociados, de manera que dos partes de distintos trabajos de

una misma máquina pueden ser intercambiados. En la variante propuesta se intenta reducir el número de vecinos posibles introduciendo la restricción de que las máquinas tienen que ser distintas también.

La restricción añadida se fundamenta en que los cambios en el tiempo total de funcionamiento más significativos se darán si se intercambian partes entre distintas máquinas, ya que el tiempo de procesamiento se ve afectado tanto por las características de las partes como por las especificaciones de las máquinas, por lo que intercambiando partes entre trabajos de la misma máquina no se modifican las componentes del tiempo de procesamiento asociadas a las características de las máquinas. Para explicar en qué consiste la modificación de dicha vecindad, al igual que en la sección 5.1, se hace uso de un ejemplo en el que se comparan la vecindad original y la vecindad con el cambio propuesto.

En una solución formada por dieciséis partes agrupadas en ocho trabajos, que a su vez están distribuidos en tres máquinas (ilustración 15), por un lado, con la modificación, se tienen un total de 168 vecinos, que se corresponden con cambiar cada parte de una máquina con el resto de las partes de distintas máquinas. Por ejemplo, de intercambiar la parte uno con otra parte se obtendrían 10 vecinos, tantos como partes haya en distintas máquinas. Por otro lado, con el tipo de vecindad Swap original se producen un total de 218 vecinos ya que ahora cada parte puede ser intercambiada con otra de un trabajo distinto de la misma máquina. Por ejemplo, la vecindad obtendría de intercambiar la parte uno con el resto de las partes posibles un total de 13 vecinos.

En la ilustración 15 se puede observar, en primer lugar, la demostración de la cantidad de vecinos que se obtienen a partir de la parte 1 en caso de usar la vecindad de manera original y, en segundo lugar, los vecinos formados a partir de la primera parte una vez introducida la modificación de la vecindad.

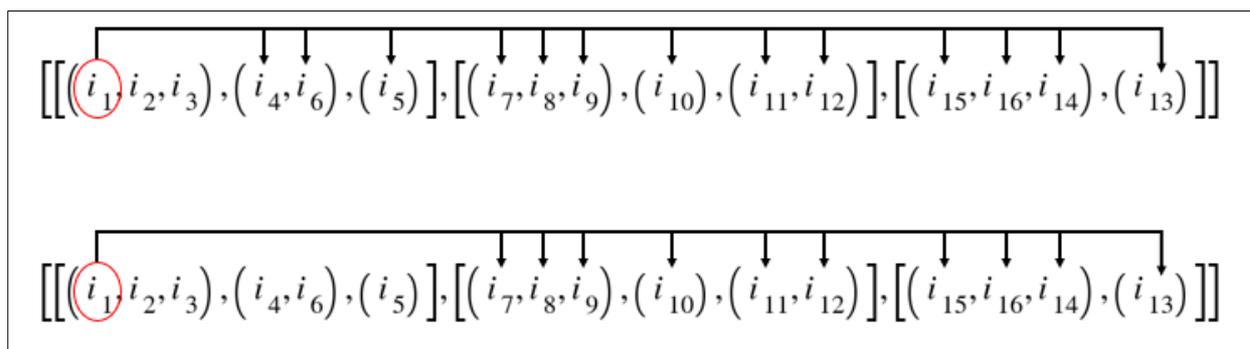


Ilustración 15. Comparación de vecindades Swap. Elaboración propia.

5.4 Variante 4. Modificación de la vecindad “Job Alloc”

Para reducir significativamente las soluciones formadas en la vecindad de tipo Job Alloc se modifica dicha vecindad improvisando una regla de despacho la cual consiste en que cada par de partes que van a ir destinadas a un nuevo trabajo se sitúen en la máquina que menos trabajos tenga.

La vecindad original tiene, por cada pareja de partes distintas, tantos vecinos como máquinas haya, sin embargo, al introducir la modificación explicada, el número de vecinos por cada par de partes únicamente es uno. El objetivo de dicha regla de despacho es situar el nuevo trabajo creado en aquella máquina que menos trabajos tenga para así, tener un equilibrio entre todas las máquinas de inversión en tiempos de set-up, ya que, mientras menos trabajos haya en una máquina menos tiempo se invierte en preparar la máquina para el procesamiento de las piezas. En caso de que haya 2 máquinas que contengan el mismo número de trabajos y este resulte ser el mismo, la pareja nueva de trabajos va destinada a ambas, es decir, se exploran los dos vecinos distintos.

Al igual que se ha hecho en las anteriores variantes, la modificación de esta vecindad se explica con un ejemplo:

Suponiendo una solución formada por dieciséis partes, agrupadas en ocho trabajos, a su vez divididos en tres máquinas, la vecindad proporciona tantos vecinos como parejas de partes se puedan formar, es decir, 78, ya que todos los nuevos trabajos formados con este par de partes van destinados únicamente a una máquina. Sin embargo, si se usa la vecindad sin la modificación propuesta habrá $78 \cdot 3$ vecinos, es decir, 234, ya que cada pareja de partes puede situarse en cualquiera de las 3 máquinas que existen. En el caso de usar la vecindad original se tiene un número de vecinos correspondientes a la ecuación 14, situada en la sección 3.3, mientras que

en el caso de usar la alternativa propuesta solamente habría un total de $\binom{n}{2}$ vecinos.

En la ilustración 16 se demuestra como una pareja de partes forma menos vecinos en la variación de la vecindad Job Alloc propuesta y se compara con los vecinos que resultan de una pareja haciendo uso de la vecindad sin modificación (ilustración 17).

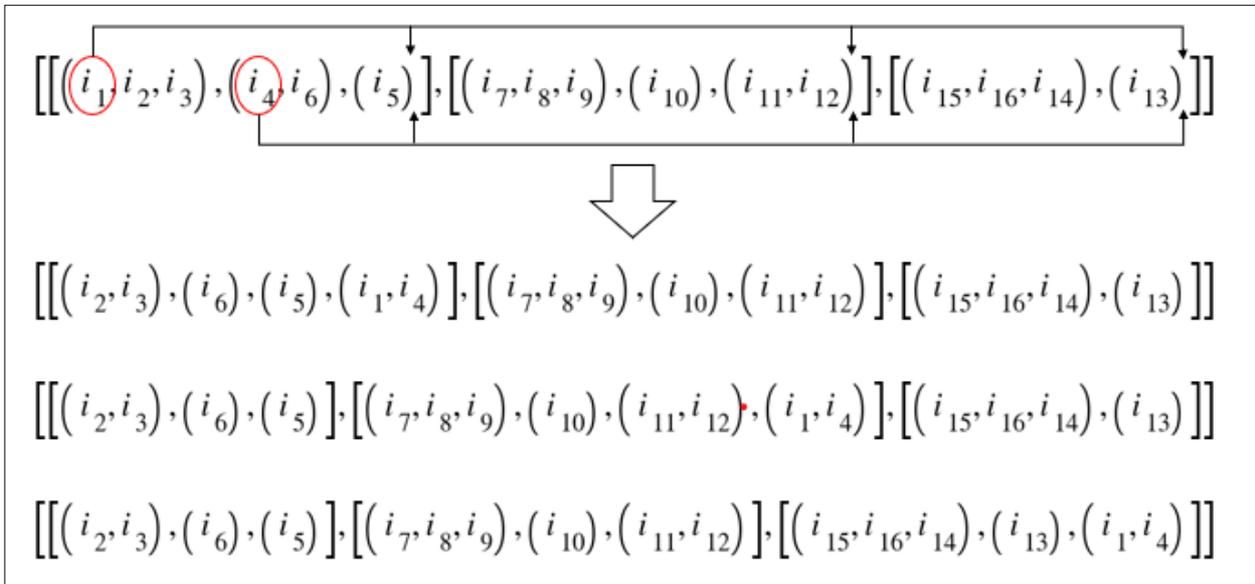


Ilustración 16. Ejemplo de vecindad tipo Job Alloc 2. Elaboración propia.

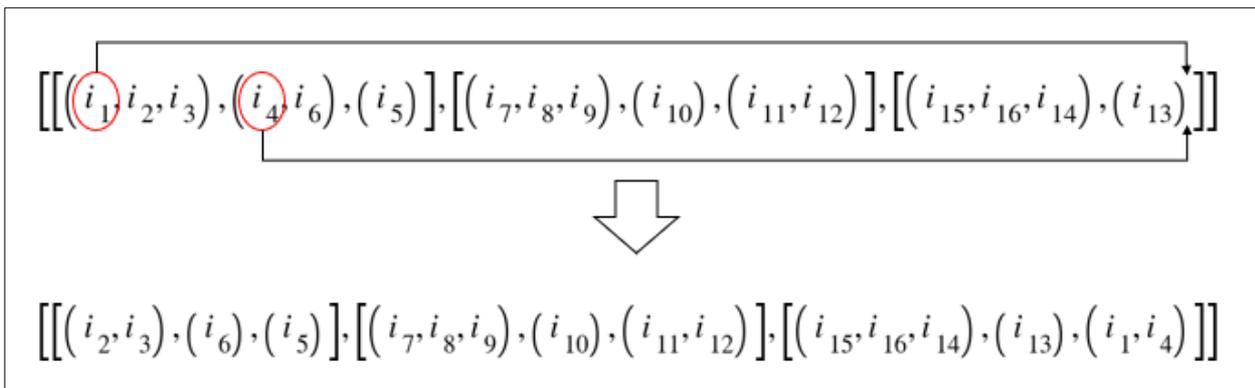


Ilustración 17. Ejemplo de vecindad tipo Job Alloc modificada. Elaboración propia.

5.5 Variante 5. Modificación del problema de empaquetado

La metaheurística, a la hora de realizar el problema de empaquetado de las partes, se organiza siguiendo tres algoritmos distintos y posteriormente seleccionando aquel con el cual se obtiene mejor Makespan. Sin embargo, según [22] la calidad de la solución de estos algoritmos se puede cuantificar mediante un indicador de rendimiento y se llega a la conclusión de que el algoritmo NFDH, con un rendimiento de 2 es peor que el FFDH y el BFDH. Los algoritmos FFDH y BFDH satisfacen la misma relación de rendimiento (17/10), sin embargo, el algoritmo BFDH trata de maximizar la utilización del trabajo tal y como se explica en la sección 3.2 del presente documento.

Debido a todo lo comentado en el primer párrafo, se considera interesante estudiar el efecto que causa hacer uso de solamente un algoritmo para el problema de empaquetado ya que supone un impacto en el tiempo computacional a la hora de realizar las iteraciones en las que se perturba la solución, que es en el momento en el que el algoritmo de empaquetado entra en acción. Por lo tanto, debido a todo lo explicado en el primer párrafo, en esta variante se selecciona la técnica BFDH como único algoritmo para realizar el problema de empaquetado.

5.6 Variantes 6 y 7. Combinación de diferentes modificaciones

Una vez analizado el comportamiento del algoritmo y los resultados obtenidos se cree conveniente unificar las diferentes modificaciones explicadas en las secciones anteriores del presente capítulo para, finalmente sacar una conclusión firme acerca de la metaheurística elaborada junto con las diferentes alternativas propuestas.

La variante 6 unifica las modificaciones correspondientes a las variantes 1, 3, 4 y 5, es decir, los tres tipos de vecindades se ven afectados al igual que el algoritmo de empaquetado. Mientras que la variante 7 incorpora los cambios de las variantes 2, 3, 4 y 5, la diferencia con la variante 6 es que la modificación de la vecindad de tipo Flip incorporada es la de la alternativa 2.

6 RESULTADOS Y CONCLUSIONES

Los resultados obtenidos se han dividido, para cada metaheurística, en distintos bloques según el tipo de instancias con las que se ha trabajado, por lo tanto, se han elaborado 3 tablas de resultados en cada caso. Las tablas asociadas a los grupos de instancias de tipo B y C se asocian a aquellas instancias consideradas de tamaño pequeño y mediano, y aquellas tablas relacionadas con las instancias de tipo R se refieren a instancias de tamaño grande. Dicha distinción entre tipos de instancias se lleva a cabo mediante un periodo de prueba en la que el algoritmo se pone en funcionamiento para cada instancia y se estudia cómo reacciona frente a ella, de modo que a aquellas instancias en las cuales se ha requerido mayor tiempo computacional se les ha denominado grandes, y de igual manera, en aquellas en las que la metaheurística ha tardado poco tiempo en calcular la solución se les caracteriza como pequeñas, de igual manera se desarrolla para las de tamaño mediano.

En la sección 3.6 se explica que se establecen distintos tiempos computacionales límites para cada tipo de instancias dependiendo de cuánto tarda en realizar una iteración el programa, por lo tanto, la determinación de que tipo de instancia se trata en cada caso tiene base en este criterio también para la decisión de que instancias se consideran pequeñas, medianas y grandes. Aquellas en las que se ha establecido un tiempo límite de 60 segundos se consideran pequeñas y se corresponden con las instancias incluidas entre las B15 y B27 y aquellas situadas entre C39 y C51 de las baterías de instancias B y C respectivamente. Cuando una instancia precisa de 300 segundos de tiempo límite se considera mediana, y son aquellas comprendidas entre B28 y B38 y aquellas dispuestas entre C52 y C62. Mientras que las instancias establecidas como grandes son todas aquellas contenidas en el grupo R, y para las mismas se establece un límite de tiempo computacional de 600 segundos.

Con el objetivo de conocer si resulta de interés el uso de la metaheurística elaborada se lleva a cabo una comparación entre las soluciones obtenidas de entre todas las elaboradas en este documento y las ya propuestas en la literatura. Además de cómo solucionar los posibles errores cometidos o mejorar la solución propuesta.

En [1] se comparan resultados de una metaheurística elaborada semejante a la confeccionada en este proyecto con un solucionador exacto denominado CPLEX y se llega a la conclusión de que, para este tipo de problemas, la aplicación CPLEX resulta demasiado tediosa de usar debido a que las soluciones obtenidas necesitan demasiado tiempo computacional. Debido a ello, la metaheurística trabajada toma como referencia los resultados obtenidos en dicho artículo mediante únicamente la metaheurística confeccionada por los autores.

Para la comparación mencionada se utilizan diversos indicadores, el más importante es el promedio de DEV, que mide la diferencia relativa, expresada en tanto por ciento, entre un valor de la función objetivo y el mejor valor obtenido por cualquiera de los algoritmos, en este caso entre los resultados de la metaheurística elaborada y los proporcionados en [1]. Además, se calcula la desviación típica del conjunto de valores obtenidos (indicada a continuación del valor medio), con el objetivo de obtener la dispersión que tienen las variaciones de Makespan. La ecuación con la cual se obtiene dicho indicador es la siguiente:

$$DEV = \frac{OFV - MEJOR\ OFV}{MEJOR\ OFV} \cdot 100 \quad (18)$$

También se calcula tanto el promedio de las funciones objetivos calculadas mediante cada método, como el promedio del tiempo que se tarda en llegar a la mejor solución (Ttb, time to best), expresado en segundos.

Para el análisis de los resultados, todas las comparaciones llevadas a cabo se dividen, por una parte, en el uso de la metaheurística para instancias pequeñas y medianas, y por otra, para instancias grandes, con la finalidad de tener en cuenta la posibilidad de que una variante sea muy precisa para un tipo de instancias, pero para otro tipo de problemas no.

Además, para estudiar el comportamiento de las distintas vecindades en cada algoritmo, se lleva a cabo un recuento de las mejoras que se obtienen en la solución en todo el procedimiento iterativo para cada instancia, así se puede observar que tanto por ciento de las mejoras obtenidas se deben a una vecindad u a otra.

6.1 Resultados de la metaheurística

En esta primera sección del capítulo final del proyecto se comparan los datos obtenidos con la metaheurística confeccionada con aquellos proporcionados en [1]. Haciendo uso del Anexo B y la información obtenida de [1] se lleva a cabo el cálculo de los indicadores comentados al inicio del presente capítulo.

Por un lado, se exponen los resultados obtenidos para las instancias de tamaño pequeño y mediano en la tabla 12.

Tabla 12. Comparativa en instancias pequeñas y medianas.

Metaheurística elaborada		Metaheurística literatura	
Makespan promedio (h)	369.06	Makespan promedio (h)	365.93
Ttb promedio (s)	110.74	Ttb promedio (s)	indefinido
Instancias en las cuales se obtiene el mejor resultado	4	Instancias en las cuales se obtiene el mejor resultado	44
Promedio DEV (%)			0.86 ± 0.66

Analizando la tabla 12, en concreto el indicador de variación de la solución DEV, se puede observar cómo, con la metaheurística elaborada, se obtienen resultados bastante próximos a los conseguidos por los autores de [1], sin embargo, para este tipo de instancias, observando la fila correspondiente al “Ttb promedio” resulta imposible comparar de manera exacta cuanto tiempo supone llegar a las soluciones obtenidas debido a que se carece de la información necesaria referente a la propuesta ya confeccionada en la literatura. No obstante, llevando a cabo una extrapolación o comparativa con los tiempos obtenidos para las instancias de tipo grande y teniendo en cuenta que en la metaheurística propuesta por los autores el límite de tiempo establecido para todas las instancias es de 60 segundos, el programa desarrollado es considerablemente peor en términos de tiempos computacionales, por lo tanto, mediante las distintas variantes propuestas se trata de mejorar la calidad de la metaheurística en este aspecto.

La desviación estándar de DEV (indicada en tanto por ciento a continuación del valor promedio) indica que, a partir de los resultados obtenidos mediante las instancias con las cuales se ha experimentado, para cualquier instancia la metaheurística se espera que aporte valores de incluso 0.2 % de desviación en el mejor de los casos o hasta un 1.52 % en la peor de las situaciones.

En cuanto al comportamiento de las vecindades en este tipo de instancias, se estudian cuantas mejoras se obtienen con cada vecindad en todo el proceso iterativo hasta llegar a la solución. Según los resultados obtenidos, un 40.86% de las mejoras se obtienen con los vecinos de tipo Flip, un 42.40% de las soluciones se ven mejoradas al explorar la vecindad de tipo Swap y el 16.74% restante resultan de la exploración en la vecindad de tipo Job Alloc. Estos datos resultan de interés para ver como varían en función de las modificaciones que se introducen a la metaheurística, dichos cambios se analizan en la sección 6.2 del presente capítulo.

Por otro lado, se presentan los valores de los indicadores obtenidos mediante los resultados calculados con la metaheurística para instancias de tamaño grande en la tabla 13.

Tabla 13. Comparativa en instancias grandes.

Metaheurística elaborada		Metaheurística literatura	
Makespan promedio (h)	912.38	Makespan promedio (h)	884.07
Ttb promedio (s)	568.25	Ttb promedio (s)	23.45
Instancias en las cuales se obtiene el mejor resultado	2	Instancias en las cuales se obtiene el mejor resultado	38
Promedio DEV (%)	5.73 ± 8.04		

En la tabla 13 se puede observar que la metaheurística se comporta de una manera más errónea, tanto en términos de tiempo invertido en encontrar la solución (Ttb), como en términos de valores de Makespan obtenidos. Sin embargo, al ser el tiempo computacional el parámetro que más afectado se ve, mediante las distintas variantes de la metaheurística propuesta, se intenta solucionar dicho problema y así reducir, en la medida de lo posible, el tiempo que se tarda en la búsqueda de las soluciones.

A diferencia del caso de las instancias medianas y pequeñas, ahora la desviación estándar referida a los valores del parámetro DEV calculados es un parámetro alarmante, debido a que adquiere un valor demasiado alto. Analizando los resultados del indicador DEV obtenidos para las instancias grandes, contenidos en el anexo B, se observa que se debe a que, en algunas instancias, el valor del Makespan obtenido está muy distanciado del proporcionado, por lo tanto, incrementa el valor de tanto la media, como la desviación típica de DEV. Estas instancias que provocan lo comentado son, por ejemplo, las de los archivos R06, R17, R23, R24 o R29, en las cuales se obtiene un DEV por encima de un 15 %. Examinando las características de estas instancias y cómo se comporta la metaheurística al ponerse en funcionamiento para calcular su solución, se llega a la conclusión de que estos valores tan lejanos se deben a que el algoritmo tarda mucho tiempo en realizar iteraciones para las mencionadas instancias, por lo tanto, al tener establecido el límite de 600 segundos el algoritmo finaliza antes de llegar a la mejor solución posible. Por ejemplo, para la instancia R06, en la cual, establecido el límite de tiempo de 600 segundos, se obtiene un DEV de 33.15 % y un Makespan de 567.23 h, sin embargo, si se deja el programa funcionando 1800 segundos se obtiene un Makespan de 512.89 h lo que significa reducir el porcentaje de desviación a un 20.39 %, valor bastante menor que en el caso inicial. Por lo tanto, en conclusión, el valor tan alto de la desviación estándar del conjunto de valores de DEV se debe a que en este grupo de instancias hay una gran variación entre las características de estas, pudiendo ser algunas de solamente de 5 máquinas y otras de hasta 19 máquinas con entre 21 y 48 partes, esto hace que en las instancias más complejas se requiera un mayor tiempo computacional y se obtengan peores resultados en el tiempo límite establecido en la metaheurística.

No obstante, aun sin tener en cuenta los resultados obtenidos para las instancias mencionadas en el párrafo anterior, tanto la media como la desviación estándar del parámetro DEV es bastante mayor que en el caso de las instancias pequeñas y medianas. Esto se debe a que, todas las instancias son más complejas y tienen mayor número de máquinas y partes, por lo tanto, requieren de soluciones más complicadas que tienen un mayor número de vecinos y en consecuencia sus vecindades serán más difíciles y tediosas de explorar.

Estudiando a que vecindad corresponden las mejoras obtenidas en el proceso de búsqueda de la solución, mediante el análisis de los resultados se obtiene que un 29.11 % de las mejoras obtenidas se obtuvieron en el proceso de búsqueda en la vecindad de tipo Flip, un 36.34 % de las mejorías en el Makespan se deben a la vecindad de tipo Swap, y el 34.35 % que resta se debe a las exploraciones en la vecindad de tipo Job Alloc.

Analizando las diferencias del comportamiento de la metaheurística en instancias pequeñas y medianas con respecto a las grandes destacan el empeoramiento del parámetro DEV con respecto a los resultados de [1], la mayor tardanza en obtener las soluciones y su gran desviación frente al tiempo que se tarda en la metaheurística propuesta en [1], y la incidencia de la vecindad Job Alloc en las instancias de tamaño grande, siendo el porcentaje de mejoras obtenidas con esta vecindad en torno al doble en instancias grandes.

6.2 Comparativa de las variantes con la metaheurística

Una vez han sido estudiados los resultados obtenidos mediante la metaheurística elaborada con respecto a la ya confeccionada en la literatura [1], se lleva a cabo una comparativa entre las distintas variantes propuestas, cuyos resultados se muestran en el anexo C, y el algoritmo confeccionado según este documento con el objetivo de analizar si dichas variantes son de interés para mejorar los resultados en términos de tiempos computacionales. Por lo tanto, los valores de DEV obtenidos para las diferentes variantes se obtienen con los resultados de la metaheurística elaborada y los obtenidos con las distintas variantes.

Por un lado, en la tabla 14 se encuentra un resumen de todos los resultados obtenidos con cada variante y la correspondiente comparativa con la metaheurística principal. Además, en la tabla 15, se incluye un resumen de la influencia que tienen los distintos tipos de vecindades en cada metaheurística a nivel del número de mejoras, indicándose en cada fila el tanto por ciento de mejoras sobre el total de mejoras que se obtienen con cada vecindad en todo el proceso iterativo hasta llegar a la solución. Dicha tabla es de gran interés para comentar las variaciones de los resultados puesto que las diferencias entre la metaheurística original y las variantes reside en las vecindades excepto en la variante 5, en la cual se modifica el problema de empaquetado. Las explicaciones correspondientes a las distintas modificaciones que afectan a la metaheurística en cada variante se sitúan en el capítulo 5.

Tabla 14. Comparativa de variantes. Instancias pequeñas y medianas.

Variante	Makespan promedio (h)	Ttb promedio (s)	DEV promedio (%)
Original	369.05	110.74	-
1. Modificación 1 de vecindad Flip	374.17	55.19	1.41 ± 1.07
2. Modificación 2 de vecindad Flip	373.91	64.85	1.34 ± 1.05
3. Modificación de vecindad Swap	373.28	75.14	1.17 ± 1.07
4. Modificación de vecindad Job Alloc	372.06	57.24	0.85 ± 0.87
5. Modificación del problema de empaquetado	372.17	95.10	0.85 ± 0.70
6. Combinación de las variantes 1, 3, 4 y 5	375.80	38.65	1.80 ± 1.16
7. Combinación de las variantes 2, 3, 4 y 5	375.34	48.15	1.69 ± 1.24

Tabla 15. Comparativa de comportamiento de vecindades. Instancias pequeñas y medianas.

Variante	Mejoras Flip (%)	Mejoras Swap (%)	Mejoras Job Alloc (%)
Original	40.86	42.40	16.74
1. Modificación 1 de vecindad Flip	34.3	44.40	21.31
2. Modificación 2 de vecindad Flip	31.37	43.53	25.10
3. Modificación de vecindad Swap	45.14	35.97	18.89
4. Modificación de vecindad Job Alloc	43.88	44.91	11.20
5. Modificación del problema de empaquetado	40.55	43.28	16.18
6. Combinación de las variantes 1, 3, 4 y 5	36.52	50.20	13.28
7. Combinación de las variantes 2, 3, 4 y 5	40.88	45.82	13.31

Analizando los datos representados en la tabla 14 se pueden llegar a varias conclusiones. En primer lugar, se observa que se produce un ahorro de tiempo a la hora de calcular las soluciones en todas y cada una de las variantes, sin embargo, en aquellas que se produce una mejora mayor en términos de tiempo también se generan empeoramientos en la solución. Estos empeoramientos se ven definidos por el parámetro DEV, el cual es mayor en aquellas variantes donde se incorporan más modificaciones, al igual que la dispersión de dichos resultados.

En segundo lugar, se observa que las variantes 6 y 7 tienen unos tiempos promedio hasta la mejor solución bastante inferiores a las demás variantes, esto se debe a que dichas metaheurísticas incorporan todas las variaciones de las demás alternativas.

Examinando por otro lado los datos proporcionados en la tabla 15, en cuanto a las vecindades, se puede destacar que, en este tipo de instancias, la vecindad de tipo Job Alloc tiene muy poca repercusión ya que en todas las variantes es la vecindad que menor número de mejoras ofrece en la búsqueda de vecindario variable, además aquella variante en la que se modifica la vecindad de este tipo, la variante 4, es la que proporciona los resultados más similares a la metaheurística principal, al igual que la variante 5, en la cual no se modifica ninguna vecindad. También es importante destacar que, en este tipo de instancias la vecindad más importante es la de tipo Flip, ya que, observando la tabla 15 se deduce que para las variantes en las que se modifica esta vecindad, se empeoran en mayor medida los resultados obtenidos en términos de Makespan y se mejoran más los tiempos de búsqueda de la solución.

Finalmente, se puede concluir con que, dependiendo del objetivo que se tenga, conviene usar un tipo de variante u otra, de modo que, si el objetivo del usuario es obtener las soluciones en el menor tiempo posible, éste se decantará por la variante 6, en la cual se obtienen, de media, resultados un 1.80 % peores que en la metaheurística final, pero se ahorra más de la mitad del tiempo en llegar a la solución.

Por otro lado, en las tablas 16 y 17 se muestran los resultados de las comparativas entre las distintas variantes para las instancias de tipo grande de la misma manera que se ha hecho anteriormente con las instancias de tamaño pequeño y mediano.

Tabla 16. Comparativa de variantes. Instancias grandes.

Variante	Makespan promedio (h)	Ttb promedio (s)	DEV promedio (%)
Original	912.70	568.25	-
1. Modificación 1 de vecindad Flip	938.78	447.77	5.70 ± 6.26
2. Modificación 2 de vecindad Flip	937.37	466.53	4.87 ± 5.25
3. Modificación de vecindad Swap	936.53	453.21	5.91 ± 13.57
4. Modificación de vecindad Job Alloc	943.46	434.11	6.18 ± 11.65
5. Modificación del problema de empaquetado	931.04	477.34	4.49 ± 8.17
6. Combinación de las variantes 1, 3, 4 y 5	959.50	378.30	9.06 ± 16.51
7. Combinación de las variantes 2, 3, 4 y 5	949.90	413.43	8.02 ± 15.92

Tabla 17. Comparativa de comportamiento de vecindades. Instancias grandes.

Variante	Mejoras Flip (%)	Mejoras Swap (%)	Mejoras Job Alloc (%)
Original	29.11	36.34	34.55
1. Modificación 1 de vecindad Flip	24.89	39.05	36.06
2. Modificación 2 de vecindad Flip	23.83	37.64	38.53
3. Modificación de vecindad Swap	29.53	30.12	40.35
4. Modificación de vecindad Job Alloc	36.86	38.94	24.20
5. Modificación del problema de empaquetado	27.80	35.39	36.80
6. Combinación de las variantes 1, 3, 4 y 5	35.48	39.96	24.55
7. Combinación de las variantes 2, 3, 4 y 5	31.95	39.53	28.52

Primeramente, en cuanto a la tabla 16, al igual que para las instancias de tamaño pequeño y mediano, se consigue una mejoría considerable en términos de tiempo a la hora de buscar la mejor solución posible con la consecuencia de empeorar los resultados de Makespan. Al igual que en la conclusión de las instancias pequeñas y medianas,

las variantes en las cuales se llevan a cabo las mejoras más importantes en términos de tiempo son aquellas en las que se combinan todas las variaciones llevadas a cabo en las vecindades y el problema de empaquetado, siendo de nuevo la variante 6 la metaheurística en la cual se llega a la solución con un menor tiempo computacional.

Al igual que cuando se comparan los resultados de la metaheurística principal con los de la proporcionada en [1], las desviaciones estándar (para el conjunto de valores de DEV para este tipo de instancias) de las variantes también toman valores demasiado altos, además se detecta que a medida que aumenta la media también lo hace la desviación. Tal y como se comenta en la sección 6.1 esto se debe a que las instancias que hay en el conjunto explorado tienen diferentes características y, por ende, unos resultados resultan peores que otros.

Tras observar la tabla 17, cabe destacar que la influencia de la vecindad Job Alloc en este tipo de instancias ha crecido con respecto a las instancias de tamaño pequeño y mediano, esto se puede comprobar haciendo hincapié en la variante cuatro, la cual al reducir de manera importante los vecinos de tipo Job Alloc se produce un empeoramiento superior al de aquellas variantes en las cuales solamente se modifica una sola vecindad (1, 2 y 3), y también un ahorro de tiempo superior al conseguido en las mencionadas vecindades.

Finalmente, de igual manera que se razona para las instancias de tipo pequeño y mediano, dependiendo del objetivo que tenga el usuario del programa, será conveniente el uso de una variante u otra.

6.3 Conclusión

En este proyecto se ha analizado el método de fabricación aditiva del tipo SLM haciendo énfasis en las desventajas que supone dicha técnica industrial. Una vez se detecta que el principal problema es el tiempo invertido en el proceso de fabricación, se decide estudiar en detalle que variables intervienen a la hora de elaborar las piezas mediante dicho método y el procedimiento que se sigue con el objetivo de minimizar el tiempo final de fabricación, también conocido como Makespan. Además de focalizar el proyecto en dicha técnica de fabricación, también se considera interesante el estudio del marco más complicado relacionado con la misma, siendo este un entorno de varias máquinas de diferentes características que actúan en paralelo con la restricción de tener que agrupar las distintas piezas en lotes debido a las características de este tipo de máquinas. Para abordar dicha cuestión, primeramente, se hace una revisión de la literatura para conocer en profundidad el problema y su modelo matemático, al igual que las distintas técnicas disponibles que existen para la resolución de este. Y posteriormente, una vez elegido el método de resolución, que es una metaheurística con búsqueda de vecindario variable, se desarrolla un programa informático que trata de obtener las mejores soluciones posibles en el menor tiempo.

Una vez elaborada la metaheurística, se comprueba que funciona correctamente comparando los resultados con algunos obtenidos de algoritmos similares proporcionados por la literatura. Finalmente se detecta la principal desventaja que posee el programa, relacionada con el tiempo computacional, y, mediante distintas alternativas se intenta mejorar dicho factor.

Comentado todo aquello que se ha llevado a cabo, finalmente se concluye que, dependiendo del objetivo del usuario, ya sea priorizar la calidad de las soluciones o el tiempo de computación, se puede hacer uso de una variante u otra. Se considera oportuno hacer uso de la metaheurística original si nuestro objetivo es llegar a soluciones mejores sin que haya interés en el tiempo transcurrido para ello y el uso de la variante 6 si lo que se busca es obtener la solución lo antes posible sin importar que haya pequeñas variaciones en cuanto al Makespan de la solución. Una combinación interesante sería utilizar la metaheurística básica, sin ninguna modificación, para las instancias pequeñas y medianas, ya que los tiempos computacionales para este tipo de instancias son mucho menores que en los de instancias grandes, y para aquellas instancias en las que haya más partes y máquinas, consideradas de tamaño grande, utilizar la variante 6. Además, como solución a aquellas instancias en las cuales la metaheurística no consigue una solución buena debido a su alta complejidad (tomando como referencia los resultados obtenidos en [1]), problema el cual se ha detectado analizando la desviación típica de los conjuntos de valores de DEV obtenidos de la comparación entre las metaheurísticas, se propone que, si se detecta que el tiempo computacional entre iteraciones es demasiado alto (del orden de entre 100 y 200 segundos), se establezca un criterio de parada distinto, ya sea imponiendo un número mínimo de iteraciones o un tiempo computacional máximo mayor, aunque todo ello supone un gran aumento en el tiempo invertido en encontrar la solución.

Como se puede observar en la primera sección del presente capítulo, los resultados obtenidos son buenos, pero

tienen margen de mejora, en el presente trabajo se ha intentado mejorar la propuesta en términos de tiempos computacionales, son su correspondiente éxito, por lo tanto, en cuanto a las líneas futuras de trabajo con las que se podría continuar con dicho proyecto, cabe destacar que es interesante centrarse en mejorar otros aspectos de las soluciones, como por ejemplo el Makespan, mediante, por ejemplo, la introducción de nuevos tipos de vecindades, trabajar en otro lenguaje de programación que aporte mejores resultados o depurar el programa de una manera diferente. Investigando como mejorar estos aspectos también se solucionará en paralelo el problema del tiempo computacional tan alto que supone el uso de la metaheurística en instancias complejas.

Una manera interesante de mejorar el Makespan de las soluciones sería hacer un estudio exhaustivo de la ecuación del tiempo de procesamiento y analizarla término a término, observando que términos incrementan más el tiempo de fabricación de los trabajos y cuales menos para así focalizar el problema en el parámetro relacionado con los términos más influyentes.

Otro método con el cual se podrían mejorar los resultados obtenidos a la hora de fabricar piezas en este tipo de máquinas es el desarrollo tecnológico de la maquinaria. Aumentando la capacidad de las máquinas, acrecentando el área máxima de las mismas se permitiría formar lotes de trabajo más grandes y se reduciría considerablemente el tiempo de procesamiento al disminuir los tiempos de Set-Up.

REFERENCIAS

- [1] Alicastro, M., Ferone, D., Festa, P., Fugaro, S., & Pastore, T. (2021). A reinforcement learning iterated local search for makespan minimization in additive manufacturing machine scheduling problems. *Computers & Operations Research*, 131, 105272.
- [2] Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.
- [3] Anaconda Software Distribution (2016). *Conda (Version 2-2.4.0)* [Computer Software].
- [4] Framiñán Torres, J. M. (2022). *Librería Scheptk (Nº de versión 0.0.6)*. Windows-Mac. Sevilla.
- [5] Haro González, J. V. (2012). *Desarrollo de un procedimiento de diseño óptimo para fabricación de componentes mediante tecnologías aditivas*.
- [6] Ligon, S. C., Liska, R., Stampfl, J., Gurr, M., & Mülhaupt, R. (2017). Polymers for 3D printing and customized additive manufacturing. *Chemical reviews*, 117(15), 10212-10290.
- [7] Zahera, M. (2012). *La fabricación aditiva, tecnología avanzada para el diseño y el desarrollo de productos*. XVI Congreso Internacional de Ingeniería de Proyectos Valencia, 11-13 de julio de 2012. Fundación Cotec.
- [8] Hernández-Castellano, P. M., Martínez-Rivero, M., Gutiérrez-Barcenilla, A., Suárez-García, L., & Marrero-Alemán, M. (2019). *Fabricación Aditiva: material didáctico interactivo (No. COMPON-2019-CINAIC-0057)*.
- [9] Lago, J. (2020). *Revisión de las Técnicas de Fabricación Aditiva y sus Aplicaciones*. Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna.
- [10] Tofail, S. A., Koumoulos, E. P., Bandyopadhyay, A., Bose, S., O'Donoghue, L., & Charitidis, C. (2018). Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials today*, 21(1), 22-37.
- [11] Hartmanis, J. (1982). *Computers and intractability: a guide to the theory of np-completeness (michael r. Garey and david s. Johnson)*. *Siam Review*, 24(1), 90.
- [12] Li, Q., Kucukkoc, I., & Zhang, D. Z. (2017). Production planning in additive manufacturing and 3D printing. *Computers & Operations Research*, 83, 157-172.
- [13] Kucukkoc, I. (2019). MILP models to minimise makespan in additive manufacturing machine scheduling problems. *Computers & Operations Research*, 105, 58-67.
- [14] Pérez González, P., Fernández-Viagas Escudero, V., & Framiñán Torres, J. M. (2020). *Programación de la Producción*. Sevilla: Sección de Publicadores, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla.

- [15] Cplex, I. I. (2009). V12. 1: User's Manual for CPLEX. International Business Machines Corporation, 46(53), 157.
- [16] Dos Santos, J. P. Q., de Melo, J. D., Neto, A. D. D., & Aloise, D. (2014). Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search. *Expert Systems with Applications*, 41(10), 4939-4949.
- [17] Martello, S., & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.
- [18] Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11), 1097-1100.
- [19] Felici, G., Ferone, D., Festa, P., Napolitano, A., & Pastore, T. (2017, June). A grasp for the minimum cost sat problem. In *International Conference on Learning and Intelligent Optimization* (pp. 64-78). Springer, Cham.
- [20] Raybaut, P. (2022). *Spyder (Nº de versión 5)*. Windows-Mac. MIT: Spyder Project Contributors.
- [21] López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43-58.
- [22] Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R., & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on computing*, 3(4), 299-325.

7 ANEXO A. EJEMPLO NUMÉRICO. PROCEDIMIENTO DE ALGORITMOS DE EMPAQUETADO

Partes ordenadas de mayor a menor altura:

Tabla 18. Ejemplo numérico. Partes ordenadas de menor a mayor altura por máquina.

Partes asociadas a la máquina 1:	Partes asociadas a la máquina 2:
Parte - Altura - Área - Volumen	Parte - Altura - Área - Volumen
1 - 27.94 - 56.85 - 272.92	1 - 29.58 - 273.83 - 840.17
2 - 18.99 - 89.68 - 683.06	2 - 26.67 - 258.54 - 1375.9
3 - 17.61 - 146.57 - 617.98	3 - 17.38 - 50.02 - 125.7
4 - 10.77 - 269.75 - 1928.6	4 - 17.04 - 99.53 - 703.08
5 - 9.18 - 728.33 - 1193.36	5 - 14.38 - 114.56 - 989.53
6 - 3.0 - 615.12 - 722.91	6 - 13.22 - 179.75 - 641.7
7 - 2.18 - 178.34 - 214.79	7 - 4.27 - 122.62 - 102.83
	8 - 3.5 - 454.89 - 683.48

7.1 Procedimiento del algoritmo NFDH

---Se inicia el algoritmo NFDH para la máquina 1 ---

Área máxima permitida en la máquina: 800.0

Área ocupada en el trabajo 1: 0

Área que se ocuparía en el trabajo 1 introduciendo la parte 1: 56.85

Se añade la parte 1 al trabajo 1

Área ocupada en el trabajo 1: 56.85

Área que se ocuparía en el trabajo 1 introduciendo la parte 2: 146.53

Se añade la parte 2 al trabajo 1

Área ocupada en el trabajo 1 : 146.53

Área que se ocuparía en el trabajo 1 introduciendo la parte 3 : 293.1

Se añade la parte 3 al trabajo 1

Área ocupada en el trabajo 1 : 293.1

Área que se ocuparía en el trabajo 1 introduciendo la parte 4 : 562.85

Se añade la parte 4 al trabajo 1

Área ocupada en el trabajo 1 : 562.85

Área que se ocuparía en el trabajo 1 introduciendo la parte 5 : 1291.18

Se excede el área máxima de la máquina

Se añade la parte 5 a un nuevo trabajo 2

Área ocupada en el trabajo 2 : 728.33

Área que se ocuparía en el trabajo 2 introduciendo la parte 6 : 1343.45

Se excede el área máxima de la máquina

Se añade la parte 6 a un nuevo trabajo 3

Área ocupada en el trabajo 3 : 615.12

Área que se ocuparía en el trabajo 3 introduciendo la parte 7 : 793.46

Se añade la parte 7 al trabajo 3

--Finaliza el algoritmo NFDH en la máquina 1—

Resultados del algoritmo NFDH en la máquina 1:

Tabla 19. Resultados del algoritmo NFDH en la máquina 1.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2	Partes asociadas al trabajo 3
Altura - Área - Volumen	Altura - Área - Volumen	Altura - Área - Volumen
27.94 - 56.85 - 272.92	9.18 - 728.33 - 1193.36	3.0 - 615.12 - 722.91
18.99 - 89.68 - 683.06		2.18 - 178.34 - 214.79
17.61 - 146.57 - 617.98		
10.77 - 269.75 - 1928.6		

---Se inicia el algoritmo NFDH en la máquina 2 ---

Área máxima permitida en la máquina: 1000.0

Área ocupada en el trabajo 1: 0

Área que se ocuparía en el trabajo 1 introduciendo la parte 1: 273.83

Se añade la parte 1 al trabajo 1

Área ocupada en el trabajo 1: 273.83

Área que se ocuparía en el trabajo 1 introduciendo la parte 2: 532.37

Se añade la parte 2 al trabajo 1

Área ocupada en el trabajo 1: 532.37

Área que se ocuparía en el trabajo 1 introduciendo la parte 3: 582.39

Se añade la parte 3 al trabajo 1

Área ocupada en el trabajo 1: 582.39

Área que se ocuparía en el trabajo 1 introduciendo la parte 4: 681.92

Se añade la parte 4 al trabajo 1

Área ocupada en el trabajo 1 : 681.92

Área que se ocuparía en el trabajo 1 introduciendo la parte 5: 796.48

Se añade la parte 5 al trabajo 1

Área ocupada en el trabajo 1: 796.48

Área que se ocuparía en el trabajo 1 introduciendo la parte 6: 976.23

Se añade la parte 6 al trabajo 1

Área ocupada en el trabajo 1 : 976.23

Área que se ocuparía en el trabajo 1 introduciendo la parte 7 : 1098.85

Se excede el área máxima de la máquina

Se añade la parte 7 a un nuevo trabajo 2

Área ocupada en el trabajo 2: 122.62

Área que se ocuparía en el trabajo 2 introduciendo la parte 8: 577.51

Se añade la parte 8 al trabajo 2

---Fin del algoritmo NFDH en la máquina 2 ---

Resultados del algoritmo NFDH en la máquina 2:

Tabla 20. Resultados del algoritmo NFDH en la máquina 2.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2
Altura - Área - Volumen	Altura - Área - Volumen
29.58 - 273.83 - 840.17	4.27 - 122.62 - 102.83
26.67 - 258.54 - 1375.9	3.5 - 454.89 - 683.48
17.38 - 50.02 - 125.7	
17.04 - 99.53 - 703.08	
14.38 - 114.56 - 989.53	
13.22 - 179.75 - 641.7	

7.2 Procedimiento del algoritmo FFDH

---Inicio del algoritmo FFDH en la máquina 1 ---

Área máxima permitida en la máquina: 800.0

Área ocupada en cada trabajo: [0]

Se comprueba si la parte 1 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 56.85

Se añade la parte 1 al trabajo 1

Área ocupada en cada trabajo: [56.85]

Se comprueba si la parte 2 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 146.53

Se añade la parte 2 al trabajo 1

Área ocupada en cada trabajo: [146.53]

Se comprueba si la parte 3 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 293.1

Se añade la parte 3 al trabajo 1

Área ocupada en cada trabajo: [293.1]

Se comprueba si la parte 4 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 562.85

Se añade la parte 4 al trabajo 1

Área ocupada en cada trabajo: [562.85]

Se comprueba si la parte 5 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1: 1291.18

Se crea un nuevo trabajo, el trabajo 2

Área ocupada en cada trabajo: [562.85, 728.33]

Se comprueba si la parte 6 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 1177.97

Se comprueba si la parte 6 cabe en el trabajo 2

Área que se ocuparía en el trabajo 2 : 1343.45

Se crea un nuevo trabajo, el trabajo 3 y se añade la parte 6

Área ocupada en cada trabajo: [562.85, 728.33, 615.12]

Se comprueba si la parte 7 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 741.19

Se añade la parte 7 al trabajo 1

---Fin algoritmo FFDH en la máquina 1 ---

Tabla 21. Resultados del algoritmo FFDH en la máquina 1.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2	Partes asociadas al trabajo 3
Altura - Área - Volumen	Altura - Área - Volumen	Altura - Área - Volumen
27.94 - 56.85 - 272.92	9.18 - 728.33 - 1193.36	3.0 - 615.12 - 722.91
18.99 - 89.68 - 683.06		
17.61 - 146.57 - 617.98		
10.77 - 269.75 - 1928.6		
2.18 - 178.34 - 214.79		

---Inicio del algoritmo FFDH en la máquina 2 ---

Área máxima permitida en la máquina: 1000.0

Área ocupada en cada trabajo: [0]

Se comprueba si la parte 1 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 273.83

Se añade la parte 1 al trabajo 1

Área ocupada en cada trabajo: [273.83]

Se comprueba si la parte 2 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 532.37

Se añade la parte 2 al trabajo 1

Área ocupada en cada trabajo: [532.37]

Se comprueba si la parte 3 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 582.39

Se añade la parte 3 al trabajo 1

Área ocupada en cada trabajo: [582.39]

Se comprueba si la parte 4 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 681.92

Se añade la parte 4 al trabajo 1

Área ocupada en cada trabajo: [681.92]

Se comprueba si la parte 5 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 796.48

Se añade la parte 5 al trabajo 1

Área ocupada en cada trabajo: [796.48]

Se comprueba si la parte 6 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 976.23

Se añade la parte 6 al trabajo 1

Área ocupada en cada trabajo: [976.23]

Se comprueba si la parte 7 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 1098.85

Se crea un nuevo trabajo, el trabajo 2 y se añade la parte 7

Área ocupada en cada trabajo: [976.23, 122.62]

Se comprueba si la parte 8 cabe en el trabajo 1

Área que se ocuparía en el trabajo 1 : 1431.12

Se comprueba si la parte 8 cabe en el trabajo 2

Área que se ocuparía en el trabajo 2 : 577.51

Se añade la parte 8 al trabajo 2

---Fin del algoritmo FFDH en la máquina 2 ---

Resultados del algoritmo FFDH en la máquina 2:

Tabla 22. Resultados del algoritmo FFDH en la máquina 2.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2
Altura - Área - Volumen	Altura - Área - Volumen
29.58 - 273.83 - 840.17	4.27 - 122.62 - 102.83
26.67 - 258.54 - 1375.9	3.5 - 454.89 - 683.48
17.38 - 50.02 - 125.7	
17.04 - 99.53 - 703.08	
14.38 - 114.56 - 989.53	
13.22 - 179.75 - 641.7	

7.3 Procedimiento del algoritmo BFDH

---Inicio del algoritmo BFDH en la máquina 1 ---

Área máxima permitida en la máquina: 800.0

El área ocupada de cada trabajo es: [0]

El área libre de cada trabajo es: [800.0]

La diferencia mínima es: 800.0

Se añade la parte 1 al trabajo 1

El área ocupada de cada trabajo es: [56.85]

El área libre de cada trabajo es: [743.15]

La diferencia mínima es: 743.15

Se añade la parte 2 al trabajo 1

El área ocupada de cada trabajo es: [146.53]

El área libre de cada trabajo es: [653.47]

La diferencia mínima es: 653.47

Se añade la parte 3 al trabajo 1

El área ocupada de cada trabajo es: [293.1]

El área libre de cada trabajo es: [506.9]

La diferencia mínima es: 506.9

Se añade la parte 4 al trabajo 1

El área ocupada de cada trabajo es: [562.85]

El área libre de cada trabajo es: [237.14999999999998]

La parte no cabe en ningún trabajo

Se crea un nuevo trabajo, el trabajo 2 y se añade la parte 5

El área ocupada de cada trabajo es: [562.85, 728.33]

El área libre de cada trabajo es: [237.14999999999998, 71.66999999999996]

La parte no cabe en ningún trabajo

Se crea un nuevo trabajo, el trabajo 3 y se añade la parte 6

El área ocupada de cada trabajo es: [562.85, 728.33, 615.12]

El área libre de cada trabajo es: [237.14999999999998, 71.66999999999996, 184.88]

La diferencia mínima es: 184.88

Se añade la parte 7 al trabajo 3

---Fin algoritmo BFDH en máquina 1 ---

Tabla 23. Resultados del algoritmo BFDH en la máquina 1.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2	Partes asociadas al trabajo 3
Altura - Área - Volumen	Altura - Área - Volumen	Altura - Área - Volumen
27.94 - 56.85 - 272.92	9.18 - 728.33 - 1193.36	3.0 - 615.12 - 722.91
18.99 - 89.68 - 683.06		2.18 - 178.34 - 214.79
17.61 - 146.57 - 617.98		
10.77 - 269.75 - 1928.6		

---Inicio del algoritmo BFDH en la máquina 2 ---

Área máxima permitida en la máquina: 1000.0

El área ocupada de cada trabajo es: [0]

El área libre de cada trabajo es: [1000.0]

La diferencia mínima es: 1000.0

Se añade la parte 1 al trabajo 1

El área ocupada de cada trabajo es: [273.83]

El área libre de cada trabajo es: [726.1700000000001]

La diferencia mínima es: 726.1700000000001

Se añade la parte 2 al trabajo 1

El área ocupada de cada trabajo es: [532.37]

El área libre de cada trabajo es: [467.63]

La diferencia mínima es: 467.63

Se añade la parte 3 al trabajo 1

El área ocupada de cada trabajo es: [582.39]

El área libre de cada trabajo es: [417.61]

La diferencia mínima es: 417.61

Se añade la parte 4 al trabajo 1

El área ocupada de cada trabajo es: [681.92]

El área libre de cada trabajo es: [318.08000000000004]

La diferencia mínima es: 318.08000000000004

Se añade la parte 5 al trabajo 1

El área ocupada de cada trabajo es: [796.48]

El área libre de cada trabajo es: [203.51999999999998]

La diferencia mínima es: 203.51999999999998

Se añade la parte 6 al trabajo 1

El área ocupada de cada trabajo es: [976.23]

El área libre de cada trabajo es: [23.769999999999982]

La parte no cabe en ningún trabajo

Se crea un nuevo trabajo, el trabajo 2 y se añade la parte 7

El área ocupada de cada trabajo es: [976.23, 122.62]

El área libre de cada trabajo es: [23.769999999999982, 877.38]

La diferencia mínima es: 877.38

Se añade la parte 8 al trabajo 2

---Fin del algoritmo BFDH en la máquina 2 ---

Tabla 24. Resultados del algoritmo BFDH en la máquina 2.

Partes asociadas al trabajo 1	Partes asociadas al trabajo 2
Altura - Área - Volumen	Altura - Área - Volumen
29.58 - 273.83 - 840.17	4.27 - 122.62 - 102.83
26.67 - 258.54 - 1375.9	3.5 - 454.89 - 683.48
17.38 - 50.02 - 125.7	
17.04 - 99.53 - 703.08	
14.38 - 114.56 - 989.53	
13.22 - 179.75 - 641.7	

8 ANEXO B. RESULTADOS DE LA METAHEURÍSTICA

Tomando como referencia las instancias proporcionadas por los autores de [1] se presentan en las siguientes tablas los resultados obtenidos con la metaheurística programada y las herramientas descritas en el punto 4.1. De manera que, se presenta en la primera columna la instancia correspondiente, en la segunda columna, el valor de la función objetivo (Makespan) obtenido, en la tercera, el tiempo computacional necesario hasta llegar a la mejor solución, y en las columnas 4, 5 y 6 se presenta el número de mejoras que se han obtenido explorando las distintas vecindades. En la columna 7 se presenta el valor de la función objetivo obtenido mediante el uso de la metaheurística realizada por los autores de [1] para la comparación llevada a cabo en la sección 6.1. Finalmente, en la columna 8 se presenta el valor del indicador DEV para la solución obtenida en la instancia correspondiente a la fila, dicho parámetro se calcula haciendo uso de los resultados proporcionados por los autores de [1] y los obtenidos con el programa desarrollado. En la tabla 27, correspondiente a la instancia de tipo R también se incluye, en la columna 8, el tiempo en llegar a la mejor solución usando el algoritmo de [1]. Además, en la última fila, se presenta el total de mejoras que se han conseguido en todas las instancias para cada tipo de vecindad.

Tabla 25. Metaheurística básica. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Número de mejoras con Flip	Número de mejoras con Swap	Número de mejoras con Job Alloc	Makespan [1] (h)	DEV (%)
B15	197.51	8.73	1	4	2	197.51	0.00
B16	205.08	19.87	4	4	2	203.89	0.58
B17	390.47	20.94	4	8	3	389.97	0.13
B18	397.89	17.50	8	7	0	397.78	0.03
B19	380.23	9.56	7	5	2	378.76	0.39
B20	386.71	11.21	9	6	2	385.09	0.42
B21	283.90	48.43	5	8	3	280.61	1.17
B22	296.98	51.96	8	7	4	294.95	0.69
B23	416.31	24.89	8	4	1	414.25	0.50
B24	437.52	18.47	5	6	2	433.10	1.02
B25	439.18	33.03	12	8	3	434.74	1.02
B26	457.16	50.09	18	21	2	454.85	0.51
B27	440.10	42.67	5	6	3	436.52	0.82
B28	458.44	28.20	5	4	1	456.55	0.41
B29	352.67	66.54	5	5	4	348.83	1.10
B30	364.71	55.22	5	9	6	359.01	1.59
B31	342.59	109.37	6	6	5	340.68	0.56
B32	358.06	142.38	12	11	3	350.61	2.12
B33	371.53	300.00	9	10	6	370.81	0.19
B34	386.17	300.00	13	10	5	381.19	1.31
B35	367.91	300.00	11	13	1	363.18	1.30
B36	378.21	300.00	12	11	5	374.07	1.11
B37	442.95	300,00	10	11	4	438,58	1.00
B38	456.95	300.00	18	18	5	451.82	1.13
			200	202	74		

Tabla 26. Metaheurística básica. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Número de mejoras con Flip	Número de mejoras con Swap	Número de mejoras con Job Alloc	Makespan [1] (h)	DEV (%)
C39	195.44	8.94	1	5	1	195.44	0.00
C40	201.22	5.03	3	4	2	199.45	0.89
C41	388.63	5.12	7	6	3	385.59	0.79
C42	381.39	12.20	6	3	3	381.39	0.00
C43	373.68	46.81	4	4	2	372.58	0.29
C44	388.75	34.56	11	10	5	380.22	2.24
C45	290.47	21.21	4	3	2	286.53	1.38
C46	296.54	9.02	9	3	1	291.78	1.63
C47	424.97	32.54	9	8	5	423.05	0.45
C48	433.82	13.47	7	8	4	430.46	0.78
C49	446.28	23.95	10	10	2	444.46	0.41
C50	454.94	18.43	9	8	4	451.51	0.76
C51	300.49	60.00	10	10	7	296.07	1.49
C52	305.57	89.43	3	5	6	300.49	1.69
C53	357.49	92.33	7	11	5	352.41	1.44
C54	362.74	102.34	10	11	4	356.93	1.63
C55	349.65	168.54	12	23	5	343.62	1.76
C56	349.91	300.00	22	23	4	347.68	0.64
C57	380.12	212.55	7	14	3	375.22	1.30
C58	372.84	300.00	10	7	3	379.12	-1.66
C59	371.73	300.00	11	7	5	367.82	1.06
C60	377.38	300.00	6	7	4	372.18	1.40
C61	447.09	300.00	12	11	4	444.22	0.64
C62	454.32	300.00	8	10	5	449.10	1.16
			198	211	89		

Tabla 27. Metaheurística básica. Resultados de instancias de tipo R.

Instancia	Makespan	Ttb (s)	Nº Mejoras Flip	Nº Mejoras Swap	Nº Mejoras Job Alloc	Makespan [1]	Ttb [1]	DEV (%)
R01	379.42	292.63	12	9	4	367.67	0.23	3.19
R02	1014.78	600.00	32	10	12	1004.25	34.52	1.05
R03	487.42	600.00	18	21	15	480.98	1.66	1.34
R04	1926.21	532.84	15	18	13	1908.18	30.33	0.94
R05	870.52	600.00	4	7	6	861.54	23.38	1.04
R06	567.23	600.00	12	12	19	426.02	36.42	33.15
R07	4126.57	530.00	2	18	0	4112.98	23.33	0.33
R08	287.36	600.00	10	7	24	289.06	18.41	-0.59
R09	548.84	600.00	13	16	18	514.06	29.24	6.76
R10	979.81	600.00	11	8	16	931.26	38.42	5.21
R11	1483.53	600.00	14	5	13	1465.57	28.81	1.23
R12	257.01	600.00	6	16	23	232.69	28.00	10.45
R13	236.33	556.22	15	17	23	232.96	1.60	1.45
R14	687.23	600.00	3	6	22	602.09	32.99	14.14
R15	209.85	600.00	9	21	13	203.98	26.71	2.88
R16	2408.60	242.00	10	16	2	2397.33	25.50	0.47
R17	403.54	600.00	13	5	20	306.46	25.92	31.68
R18	447.33	600.00	8	20	13	432.92	32.60	3.33
R19	1206.89	600.00	4	13	1	1195.61	16.75	0.94
R20	669.15	260.41	6	10	6	650.70	31.20	2.84
R21	2416.19	600.00	2	26	2	2404.41	29.10	0.49
R22	352.57	600.00	8	11	15	338.62	30.24	4.12
R23	463.85	600.00	9	4	21	398.80	14.31	16.31
R24	561.43	600.00	6	2	19	478.96	33.86	17.22
R25	488.10	600.00	9	12	3	488.10	0.03	0.00
R26	512.39	516.03	13	20	18	507.71	9.23	0.92
R27	773.03	600.00	13	11	24	706.12	29.39	9.48
R28	350.49	600.00	4	14	5	350.48	0.01	0.00
R29	567.16	600.00	6	5	26	466.19	23.34	21.66
R30	763.00	600.00	10	10	14	693.75	29.24	9.98
R31	698.84	600.00	5	7	14	675.53	28.15	3.45
R32	380.25	600.00	7	21	12	379.52	6.12	0.19
R33	2563.60	600.00	18	14	10	2560.00	26.70	0.14
R34	478.50	600.00	5	13	8	454.26	28.41	5.34
R35	437.49	600.00	9	10	7	428.43	25.56	2.12
R36	2278.04	600.00	14	30	1	2274.81	39.73	0.14
R37	132.40	600.00	31	27	24	128.12	0.18	3.34
R38	2003.44	600.00	21	27	6	1992.41	44.64	0.55
R39	292.33	600.00	25	13	13	280.55	33.00	4.20
R40	797.44	600.00	7	16	16	739.64	20.59	7.81
			439	548	521			

9 ANEXO C. RESULTADOS DE LAS VARIANTES PROPUESTAS

En este anexo se presentan los resultados obtenidos mediante las diferentes variantes propuestas a la metaheurística. El anexo se divide en siete secciones en las cuales los datos son representados mediante tablas, las cuales contienen, en la columna 1. el nombre de la instancia. en la columna 2, el Makespan obtenido, en la columna 3 el tiempo hasta llegar a la mejor solución, en las columnas 4. 5 y 6 se presentan las mejoras obtenidas mediante cada tipo de vecindad, en la columna 7 se presenta el Makespan obtenido mediante la metaheurística principal, dichos datos se presentan en las tablas del anexo B, para la comparación entre los distintos algoritmos, y en la columna 8 se presentan los valores del parámetro DEV, tomando como datos para su cálculo los valores del Makespan obtenidos con la metaheurística correspondiente y los obtenidos con la metaheurística principal.

9.1 Resultados de variante 1

Tabla 28. Variante 1. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	202.40	0.31	3	8	1	197.51	2.48
B16	207.63	4.61	3	6	2	205.08	1.25
B17	390.65	3.17	7	6	1	390.47	0.05
B18	409.42	2.58	8	4	2	397.89	2.90
B19	385.71	8.33	5	15	2	380.23	1.44
B20	392.46	7.00	4	11	3	386.71	1.49
B21	284.06	11.18	2	5	2	283.90	0.06
B22	299.67	7.80	3	6	2	296.98	0.91
B23	418.44	15.22	4	7	2	416.31	0.51
B24	441.39	3.17	3	9	3	437.52	0.88
B25	443.90	18.91	5	7	1	439.18	1.07
B26	460.90	21.62	4	15	5	457.16	0.82
B27	442.95	3.34	6	7	4	440.10	0.65
B28	467.03	28.06	5	9	1	458.44	1.87
B29	358.04	21.65	8	7	2	352.67	1.52
B30	372.15	56.07	4	3	1	364.71	2.04
B31	348.32	82.72	6	4	2	342.59	1.67
B32	361.15	137.50	11	5	3	358.06	0.87
B33	376.99	72.25	7	9	5	371.53	1.47
B34	394.19	206.00	9	2	5	386.17	2.08
B35	367.91	182.22	8	4	8	367.91	0.00
B36	383.72	50.20	9	14	5	378.21	1.46
B37	448.62	101.23	16	15	8	442.95	1.28
B38	457.85	143.89	13	18	9	456.95	0.20
			153	196	79		

Tabla 29. Variante 1. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	196.71	3.20	2	2	1	195.44	0.65
C40	204.77	1.03	5	5	1	201.22	1.76
C41	389.75	4.45	3	4	3	388.63	0.29
C42	401.26	4.67	2	5	3	381.39	5.21
C43	381.10	4.53	4	8	3	373.68	1.99
C44	388.75	1.76	5	12	5	388.75	0.00
C45	291.73	4.76	5	4	2	290.47	0.43
C46	298.16	2.96	6	7	4	296.54	0.55
C47	428.32	10.37	2	10	6	424.97	0.79
C48	436.47	6.02	4	2	5	433.82	0.61
C49	453.68	3.72	6	10	7	446.28	1.66
C50	458.68	8.63	9	6	7	454.94	0.82
C51	315.34	31.07	10	10	8	300.49	4.94
C52	311.20	47.25	5	5	5	305.57	1.84
C53	359.83	53.23	11	12	6	357.49	0.65
C54	370.27	53.24	10	9	6	362.74	2.08
C55	354.10	53.59	11	18	5	349.65	1.27
C56	356.46	37.47	14	19	3	349.91	1.87
C57	389.75	81.07	9	14	5	380.12	2.53
C58	384.39	100.18	6	8	4	372.84	3.10
C59	377.59	285.75	8	6	7	371.73	1.58
C60	384.78	61.36	7	7	4	377.38	1.96
C61	450.47	300.00	7	9	6	447.09	0.76
C62	461.23	300.00	5	12	7	454.32	1.52
			156	204	113		

Tabla 30. Variante 1. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	381.54	292.63	10	22	16	379.42	0.56
R02	1060.12	600.00	16	22	24	1014.78	4.47
R03	497.82	600.00	14	28	19	487.42	2.13
R04	1940.24	365.51	18	23	17	1926.21	0.73
R05	878.61	95.62	12	14	14	870.52	0.93
R06	589.04	600.00	4	10	9	567.23	3.84
R07	4134.11	530.00	7	13	10	4126.57	0.18
R08	345.27	600.00	7	15	5	287.36	20.15
R09	550.01	600.00	6	11	7	548.84	0.21
R10	1014.61	600.00	7	6	7	979.81	3.55
R11	1512.87	600.00	8	17	18	1483.53	1.98
R12	310.76	600.00	2	3	6	257.01	20.91
R13	266.05	386.08	7	19	11	236.33	12.58
R14	718.61	600.00	4	8	7	687.23	4.57
R15	232.27	600.00	11	7	12	209.85	10.68
R16	2425.60	138.94	3	10	10	2408.60	0.71
R17	466.35	600.00	4	13	13	403.54	15.57
R18	482.09	600.00	7	17	24	447.33	7.77
R19	1212.52	38.87	12	21	21	1206.89	0.47
R20	683.97	143.07	13	14	11	669.15	2.22
R21	2417.34	600.00	7	6	7	2416.19	0.05
R22	401.85	600.00	12	26	12	352.57	13.98
R23	494.40	600.00	2	9	11	463.85	6.59
R24	605.48	600.00	15	20	12	561.43	7.85
R25	488.10	182.22	7	15	15	488.10	0.00
R26	525.25	140.87	9	13	12	512.39	2.51
R27	816.37	600.00	5	7	6	773.03	5.61
R28	368.79	133.28	9	11	6	350.49	5.22
R29	652.79	600.00	4	7	4	567.16	15.10
R30	784.54	600.00	3	20	10	763.00	2.82
R31	734.12	600.00	5	14	19	698.84	5.05
R32	420.87	164.74	8	13	14	380.25	10.68
R33	2576.32	185.95	6	7	13	2563.60	0.50
R34	501.52	600.00	13	6	14	478.50	4.81
R35	448.11	331.81	5	17	15	437.49	2.43
R36	2300.86	174.20	16	20	7	2278.04	1.00
R37	164.35	516.35	12	17	18	132.40	24.14
R38	2033.62	456.44	8	14	12	2003.44	1.51
R39	297.33	434.40	22	7	20	292.33	1.71
R40	816.84	600.00	10	7	19	797.44	2.43
			350	549	507		

9.2 Resultados de variante 2

Tabla 31. Variante 2. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Tfb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	201.59	1.94	4	3	3	197.51	2.06
B16	209.76	1.31	2	6	1	205.08	2.28
B17	398.32	4.06	5	6	2	390.47	2.01
B18	404.45	1.59	4	4	3	397.89	1.65
B19	384.72	6.42	0	11	5	380.23	1.18
B20	392.00	3.31	4	3	3	386.71	1.37
B21	286.81	8.09	5	7	3	283.90	1.03
B22	298.79	8.12	4	5	0	296.98	0.61
B23	423.31	23.10	5	3	1	416.31	1.68
B24	441.62	9.44	6	4	0	437.52	0.94
B25	441.07	12.16	4	5	4	439.18	0.43
B26	457.16	1.17	5	4	2	457.16	0.00
B27	441.99	8.01	6	8	4	440.10	0.43
B28	467.54	7.78	3	4	3	458.44	1.99
B29	354.03	44.88	6	4	6	352.67	0.38
B30	367.63	35.27	3	15	5	364.71	0.80
B31	348.34	43.15	4	7	4	342.59	1.68
B32	359.97	145.31	8	9	4	358.06	0.54
B33	381.25	215.92	4	9	3	371.53	2.62
B34	390.31	81.28	7	12	4	386.17	1.07
B35	373.70	220.53	5	5	5	367.91	1.57
B36	380.62	193.21	11	10	5	378.21	0.64
B37	451.29	300.00	4	8	7	442.95	1.88
B38	467.82	300.00	9	18	10	456.95	2.38
			118	170	87		

Tabla 32. Variante 2. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	201.98	0.52	3	2	3	195.44	3.35
C40	201.48	0.59	2	6	1	201.22	0.13
C41	391.29	4.44	4	5	0	388.63	0.68
C42	401.73	4.12	2	5	2	381.39	5.33
C43	378.38	7.95	3	9	5	373.68	1.26
C44	388.75	1.94	4	5	2	388.75	0.00
C45	292.15	5.33	2	4	3	290.47	0.58
C46	297.19	5.19	1	4	0	296.54	0.22
C47	428.26	6.24	3	5	4	424.97	0.77
C48	439.03	3.50	5	5	3	433.82	1.20
C49	453.25	12.14	3	6	6	446.28	1.56
C50	459.64	5.89	7	11	3	454.94	1.03
C51	308.46	70.98	10	10	4	300.49	2.65
C52	309.50	36.36	6	2	6	305.57	1.29
C53	357.49	33.42	6	2	7	357.49	0.00
C54	370.30	31.42	5	7	4	362.74	2.08
C55	351.81	171.67	5	8	6	349.65	0.62
C56	354.26	131.67	7	13	3	349.91	1.24
C57	382.59	46.17	6	4	6	380.12	0.65
C58	388.32	83.55	8	6	5	372.84	4.15
C59	377.96	153.54	10	9	4	371.73	1.68
C60	378.23	242.24	9	16	10	377.38	0.23
C61	452.14	183.94	7	10	13	447.09	1.13
C62	459.36	193.74	9	16	9	454.32	1.11
			127	170	109		

Tabla 33. Variante 2. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	379.42	307.01	6	17	16	379.42	0.00
R02	1036.30	600.00	16	17	29	1014.78	2.12
R03	497.82	567.56	9	31	19	487.42	2.13
R04	1968.31	379.05	13	24	14	1926.21	2.19
R05	878.61	120.84	9	11	11	870.52	0.93
R06	575.24	600.00	6	9	14	567.23	1.41
R07	4126.57	600.00	8	16	14	4126.57	0.00
R08	326.49	532.44	4	20	10	287.36	13.62
R09	552.50	600.00	4	15	11	548.84	0.67
R10	1014.61	600.00	2	7	12	979.81	3.55
R11	1501.50	487.45	6	16	20	1483.53	1.21
R12	312.52	600.00	1	5	2	257.01	21.60
R13	246.39	600.00	12	23	13	236.33	4.26
R14	707.24	600.00	3	3	5	687.23	2.91
R15	232.27	600.00	8	7	15	209.85	10.68
R16	2463.52	204.67	8	11	13	2408.60	2.28
R17	439.58	600.00	9	12	15	403.54	8.93
R18	478.45	489.60	2	16	21	447.33	6.96
R19	1212.52	63.55	17	17	21	1206.89	0.47
R20	694.57	178.42	9	19	9	669.15	3.80
R21	2425.64	537.58	9	7	10	2416.19	0.39
R22	369.67	600.00	17	30	11	352.57	4.85
R23	489.38	600.00	5	5	13	463.85	5.50
R24	605.48	600.00	18	22	11	561.43	7.85
R25	511.13	257.60	3	12	11	488.10	4.72
R26	525.25	114.24	10	14	12	512.39	2.51
R27	788.34	600.00	9	2	10	773.03	1.98
R28	375.11	176.67	10	7	7	350.49	7.02
R29	637.41	600.00	3	7	5	567.16	12.39
R30	796.34	569.70	1	16	13	763.00	4.37
R31	716.74	600.00	6	19	22	698.84	2.56
R32	400.09	341.72	6	9	17	380.25	5.22
R33	2604.01	295.34	5	11	11	2563.60	1.58
R34	512.54	600.00	15	11	19	478.50	7.11
R35	448.11	295.59	3	15	20	437.49	2.43
R36	2288.35	301.31	16	24	12	2278.04	0.45
R37	164.35	516.35	15	15	18	132.40	24.14
R38	2071.46	600.00	11	13	15	2003.44	3.40
R39	305.83	524.41	24	10	19	292.33	4.62
R40	814.94	600.00	9	3	21	797.44	2.19
			347	548	561		

9.3 Resultados de variante 3

Tabla 34. Variante 3. Resultados de instancias tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	200.38	3.23	6	3	2	197.51	1.46
B16	206.09	1.28	1	2	2	205.08	0.49
B17	394.60	8.00	2	5	1	390.47	1.06
B18	399.18	10.19	11	5	6	397.89	0.33
B19	380.44	2.59	3	2	3	380.23	0.06
B20	387.83	4.91	3	2	5	386.71	0.29
B21	285.19	8.48	5	3	3	283.90	0.46
B22	299.80	4.67	8	1	2	296.98	0.95
B23	421.10	22.80	10	4	3	416.31	1.15
B24	442.12	7.11	1	5	2	437.52	1.05
B25	441.30	13.95	2	3	1	439.18	0.48
B26	460.29	4.81	0	2	1	457.16	0.69
B27	442.12	11.07	17	13	5	440.10	0.46
B28	460.50	3.87	6	4	1	458.44	0.45
B29	352.67	44.03	9	15	5	352.67	0.00
B30	365.58	50.06	15	14	4	364.71	0.24
B31	346.99	28.20	7	5	6	342.59	1.29
B32	359.44	23.57	9	11	2	358.06	0.39
B33	376.77	240.39	12	13	4	371.53	1.41
B34	386.17	169.28	19	16	6	386.17	0.00
B35	368.37	300.00	25	15	8	367.91	0.12
B36	381.60	167.47	13	10	6	378.21	0.90
B37	446.77	232.23	24	13	4	442.95	0.86
B38	456.95	300.00	9	7	3	456.95	0.00
			217	173	85		

Tabla 35. Variante 3. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	196.71	0.78	3	3	2	195.44	0.65
C40	203.83	4.94	5	4	4	201.22	1.29
C41	390.71	3.60	4	2	1	388.63	0.54
C42	391.39	6.27	3	4	1	381.39	2.62
C43	388.12	8.50	6	3	2	373.68	3.87
C44	391.22	8.00	5	4	2	388.75	0.64
C45	298.75	11.69	8	9	2	290.47	2.85
C46	305.36	8.20	5	4	2	296.54	2.97
C47	438.93	12.93	8	3	6	424.97	3.29
C48	449.63	7.65	6	3	1	433.82	3.64
C49	449.63	5.69	6	4	4	446.28	0.75
C50	456.34	16.24	5	6	2	454.94	0.31
C51	308.59	52.39	7	7	4	300.49	2.70
C52	307.44	18.84	3	6	5	305.57	0.61
C53	359.12	95.11	4	3	4	357.49	0.46
C54	369.16	98.35	3	4	5	362.74	1.77
C55	351.72	97.96	10	16	4	349.65	0.59
C56	358.39	104.25	9	8	4	349.91	2.42
C57	382.87	90.07	17	6	7	380.12	0.73
C58	386.16	300.00	12	12	3	372.84	3.57
C59	380.42	300.00	15	10	3	371.73	2.34
C60	382.75	300.00	11	10	5	377.38	1.42
C61	453.40	300.00	10	3	2	447.09	1.41
C62	454.32	92.92	17	11	7	454.32	0.00
			182	145	82		

Tabla 36. Variante 3. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	380.52	292.04	8	15	16	379.42	0.29
R02	1059.93	600.00	15	17	18	1014.78	4.45
R03	516.97	419.34	8	20	12	487.42	6.06
R04	1930.45	357.48	10	9	7	1926.21	0.22
R05	871.95	114.54	16	16	16	870.52	0.16
R06	610.23	600.00	5	5	7	567.23	7.58
R07	4135.22	200.03	14	4	21	4126.57	0.21
R08	529.02	600.00	5	11	6	287.36	84.10
R09	568.15	600.00	12	10	22	548.84	3.52
R10	979.81	600.00	11	6	10	979.81	0.00
R11	1514.41	600.00	10	7	18	1483.53	2.08
R12	321.95	600.00	12	8	19	257.01	25.27
R13	241.95	556.22	10	11	11	236.33	2.38
R14	687.23	600.00	8	7	21	687.23	0.00
R15	233.14	600.00	10	13	16	209.85	11.10
R16	2418.10	134.15	11	8	18	2408.60	0.39
R17	460.83	600.00	7	8	6	403.54	14.20
R18	475.26	600.00	13	15	13	447.33	6.24
R19	1206.89	76.69	16	9	4	1206.89	0.00
R20	689.73	163.82	13	12	10	669.15	3.08
R21	2420.80	600.00	11	8	12	2416.19	0.19
R22	401.90	600.00	8	12	15	352.57	13.99
R23	485.07	600.00	7	14	15	463.85	4.57
R24	564.16	600.00	2	8	10	561.43	0.49
R25	488.24	373.83	12	4	14	488.10	0.03
R26	521.56	118.27	5	14	25	512.39	1.79
R27	773.03	600.00	3	6	7	773.03	0.00
R28	353.59	254.03	10	9	18	350.49	0.89
R29	582.55	600.00	6	12	10	567.16	2.71
R30	788.04	600.00	20	15	6	763.00	3.28
R31	734.28	600.00	9	9	17	698.84	5.07
R32	391.59	178.14	5	6	14	380.25	2.98
R33	2575.53	222.73	12	10	9	2563.60	0.47
R34	485.23	600.00	11	23	19	478.50	1.41
R35	441.90	239.33	5	11	6	437.49	1.01
R36	2291.33	202.04	11	6	16	2278.04	0.58
R37	152.33	600.00	15	13	18	132.40	15.05
R38	2043.91	425.58	10	8	7	2003.44	2.02
R39	305.87	600.00	18	5	18	292.33	4.63
R40	828.55	600.00	7	5	21	797.44	3.90
			401	409	548		

9.4 Resultados de variante 4

Tabla 37. Variante 4. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	201.75	1.14	3	1	0	197.51	2.15
B16	205.08	0.25	4	4	2	205.08	0.00
B17	391.06	3.51	0	1	0	390.47	0.15
B18	406.03	8.26	4	5	2	397.89	2.05
B19	381.94	1.85	6	7	1	380.23	0.45
B20	391.81	1.68	1	1	4	386.71	1.32
B21	285.38	12.97	10	10	3	283.90	0.52
B22	298.39	9.51	2	7	3	296.98	0.48
B23	417.17	20.31	0	2	1	416.31	0.21
B24	441.52	7.47	11	11	1	437.52	0.91
B25	439.18	14.49	5	4	3	439.18	0.00
B26	461.93	19.24	2	9	4	457.16	1.04
B27	440.10	27.17	7	4	0	440.10	0.00
B28	464.87	14.42	11	17	1	458.44	1.40
B29	357.70	29.67	10	11	5	352.67	1.43
B30	366.00	40.57	5	3	3	364.71	0.35
B31	346.01	12.95	16	13	4	342.59	1.00
B32	359.67	31.36	8	9	0	358.06	0.45
B33	371.53	179.61	11	11	1	371.53	0.00
B34	388.77	58.96	16	14	5	386.17	0.67
B35	368.42	40.17	17	13	4	367.91	0.14
B36	378.21	193.98	31	17	7	378.21	0.00
B37	445.48	300.00	27	27	6	442.95	0.57
B38	458.28	300.00	11	12	0	456.95	0.29
			218	213	60		

Tabla 38. Variante 4. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	200.04	0.81	0	3	1	195.44	2.36
C40	205.10	2.72	3	8	1	201.22	1.92
C41	389.85	4.13	6	4	1	388.63	0.31
C42	395.91	5.77	9	7	0	381.39	3.81
C43	377.17	4.33	4	5	1	373.68	0.94
C44	389.12	5.04	8	6	3	388.75	0.10
C45	290.63	5.43	1	3	2	290.47	0.05
C46	298.10	3.93	1	4	0	296.54	0.53
C47	433.41	10.39	2	10	0	424.97	1.99
C48	436.16	5.77	4	3	0	433.82	0.54
C49	449.55	24.94	13	8	3	446.28	0.73
C50	455.00	5.18	2	4	0	454.94	0.01
C51	304.88	13.47	5	4	2	300.49	1.46
C52	309.22	12.89	4	4	2	305.57	1.19
C53	358.88	75.98	15	19	2	357.49	0.39
C54	367.57	72.45	14	15	4	362.74	1.33
C55	353.25	36.24	10	3	5	349.65	1.03
C56	349.91	40.85	7	12	1	349.91	0.00
C57	381.40	83.60	10	8	4	380.12	0.34
C58	386.55	98.30	14	12	3	372.84	3.68
C59	373.81	219.34	18	24	6	371.73	0.56
C60	380.06	215.39	20	23	5	377.38	0.71
C61	450.27	171.08	15	7	1	447.09	0.71
C62	456.91	300.00	24	28	2	454.32	0.57
			209	224	49		

Tabla 39. Variante 4. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	380.52	115.21	4	18	8	379.42	0.29
R02	1023.61	343.21	17	24	11	1014.78	0.87
R03	509.78	306.52	13	32	10	487.42	4.59
R04	1927.09	392.48	11	23	5	1926.21	0.05
R05	875.91	341.67	14	13	6	870.52	0.62
R06	858.23	600.00	21	16	10	567.23	51.30
R07	4129.27	178.80	9	8	4	4126.57	0.07
R08	357.39	600.00	11	5	14	287.36	24.37
R09	608.35	600.00	16	8	13	548.84	10.84
R10	988.35	600.00	4	14	9	979.81	0.87
R11	1486.48	600.00	26	29	7	1483.53	0.20
R12	331.25	600.00	11	9	10	257.01	28.89
R13	243.63	237.61	12	19	9	236.33	3.09
R14	861.60	600.00	10	9	12	687.23	25.37
R15	209.85	600.00	25	31	19	209.85	0.00
R16	2408.60	242.00	20	20	6	2408.60	0.00
R17	564.20	600.00	24	16	15	403.54	39.81
R18	450.42	600.00	21	23	12	447.33	0.69
R19	1207.80	260.73	8	18	2	1206.89	0.08
R20	669.15	173.76	27	19	10	669.15	0.00
R21	2416.19	488.90	13	17	5	2416.19	0.00
R22	374.33	600.00	20	21	12	352.57	6.17
R23	495.74	600.00	9	12	11	463.85	6.88
R24	647.81	600.00	9	9	14	561.43	15.39
R25	488.24	600.00	9	11	5	488.10	0.03
R26	523.82	114.79	15	14	14	512.39	2.23
R27	801.82	600.00	13	5	16	773.03	3.72
R28	350.48	106.95	7	12	8	350.49	0.00
R29	571.73	600.00	8	5	12	567.16	0.81
R30	860.27	600.00	11	6	15	763.00	12.75
R31	704.68	600.00	20	17	16	698.84	0.84
R32	380.25	600.00	9	12	13	380.25	0.00
R33	2566.05	85.41	19	24	3	2563.60	0.10
R34	484.26	172.83	16	13	14	478.50	1.20
R35	437.49	600.00	18	11	11	437.49	0.00
R36	2285.07	175.19	7	20	3	2278.04	0.31
R37	132.40	134.21	43	26	19	132.40	0.00
R38	2003.44	600.00	39	27	6	2003.44	0.00
R39	295.49	294.09	19	31	9	292.33	1.08
R40	827.53	600.00	12	8	9	797.44	3.77
			620	655	407		

9.5 Resultados de variante 5

Tabla 40. Variante 5. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	200.82	0.76	1	1	0	197.51	1.67
B16	205.20	2.30	3	3	2	205.08	0.06
B17	392.78	2.69	3	5	1	390.47	0.59
B18	402.23	2.56	4	2	1	397.89	1.09
B19	382.88	7.03	8	4	2	380.23	0.70
B20	394.34	10.01	10	5	5	386.71	1.97
B21	284.77	11.78	4	2	2	283.90	0.31
B22	296.98	9.30	2	1	2	296.98	0.00
B23	420.83	16.88	3	7	1	416.31	1.09
B24	440.03	10.44	5	3	2	437.52	0.57
B25	439.62	31.05	14	15	4	439.18	0.10
B26	464.26	47.69	13	10	3	457.16	1.55
B27	441.21	32.89	4	5	3	440.10	0.25
B28	465.92	30.21	5	10	1	458.44	1.63
B29	355.44	74.83	6	5	6	352.67	0.78
B30	367.48	24.41	2	6	1	364.71	0.76
B31	342.59	107.51	8	12	6	342.59	0.00
B32	360.30	117.88	8	15	2	358.06	0.63
B33	378.10	300.00	18	16	4	371.53	1.77
B34	391.66	300.00	14	20	6	386.17	1.42
B35	370.86	188.82	8	10	3	367.91	0.80
B36	384.76	300.00	21	22	5	378.21	1.73
B37	442.95	300.00	14	13	4	442.95	0.00
B38	462.86	300.00	26	23	4	456.95	1.29
			204	215	70		

Tabla 41. Variante 5. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	198.21	1.37	3	3	1	195.44	1.42
C40	202.64	2.08	2	4	0	201.22	0.70
C41	388.75	3.92	1	4	2	388.63	0.03
C42	391.46	3.10	3	3	3	381.39	2.64
C43	378.16	3.92	2	5	2	373.68	1.20
C44	388.75	44.20	6	4	3	388.75	0.00
C45	292.42	18.26	3	8	0	290.47	0.67
C46	299.74	1.35	1	1	0	296.54	1.08
C47	428.72	9.05	4	4	3	424.97	0.88
C48	433.82	4.59	3	1	2	433.82	0.00
C49	448.00	6.07	1	3	0	446.28	0.39
C50	460.19	9.96	5	4	1	454.94	1.15
C51	300.49	54.00	6	7	3	300.49	0.00
C52	311.01	69.69	4	12	5	305.57	1.78
C53	359.83	76.74	9	5	3	357.49	0.65
C54	368.67	69.16	6	7	5	362.74	1.63
C55	349.65	135.33	7	14	4	349.65	0.00
C56	353.72	300.00	31	33	10	349.91	1.09
C57	380.12	219.13	12	11	2	380.12	0.00
C58	381.37	192.52	8	6	5	372.84	2.29
C59	377.38	241.32	16	13	6	371.73	1.52
C60	377.38	269.98	9	1	8	377.38	0.00
C61	450.36	300.00	16	16	6	447.09	0.73
C62	454.69	300.00	24	28	10	454.32	0.08
			182	197	84		

Tabla 42. Variante 5. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	379.42	192.75	9	8	9	379.42	0.00
R02	1027.01	600.00	29	15	13	1014.78	1.20
R03	497.64	600.00	13	27	11	487.42	2.10
R04	1926.21	532.84	12	20	8	1926.21	0.00
R05	881.14	600.00	9	5	5	870.52	1.22
R06	602.38	600.00	7	10	18	567.23	6.20
R07	4126.57	410.83	4	14	4	4126.57	0.00
R08	420.72	600.00	8	12	30	287.36	46.41
R09	557.29	144.20	10	11	23	548.84	1.54
R10	1013.29	246.47	9	12	14	979.81	3.42
R11	1502.34	600.00	9	6	12	1483.53	1.27
R12	298.88	600.00	6	11	22	257.01	16.29
R13	241.91	478.99	11	17	20	236.33	2.36
R14	717.31	268.33	5	9	24	687.23	4.38
R15	225.20	600.00	9	16	13	209.85	7.32
R16	2408.60	109.10	6	11	3	2408.60	0.00
R17	492.50	600.00	10	10	20	403.54	22.05
R18	483.86	600.00	10	15	15	447.33	8.17
R19	1214.05	395.55	3	8	2	1206.89	0.59
R20	669.15	260.41	10	8	5	669.15	0.00
R21	2416.19	533.21	3	24	6	2416.19	0.00
R22	398.54	398.54	11	10	17	352.57	13.04
R23	493.22	600.00	9	5	18	463.85	6.33
R24	561.43	600.00	8	7	20	561.43	0.00
R25	488.24	588.24	6	11	5	488.10	0.03
R26	516.82	516.03	11	17	14	512.39	0.86
R27	779.96	600.00	12	6	22	773.03	0.90
R28	359.97	359.96	6	10	7	350.49	2.71
R29	567.16	600.00	10	4	21	567.16	0.00
R30	781.94	600.00	6	9	19	763.00	2.48
R31	719.88	600.00	9	10	14	698.84	3.01
R32	391.59	391.59	11	16	14	380.25	2.98
R33	2563.60	600.00	15	14	6	2563.60	0.00
R34	504.41	504.40	2	18	13	478.50	5.41
R35	442.95	442.95	12	12	12	437.49	1.25
R36	2278.04	469.70	9	30	6	2278.04	0.00
R37	142.62	143.62	34	25	21	132.40	7.72
R38	2025.59	600.00	20	24	10	2003.44	1.11
R39	306.04	306.04	21	18	15	292.33	4.69
R40	817.91	600.00	10	12	17	797.44	2.57
			414	527	548		

9.6 Resultados de variante 6

Tabla 43. Variante 6. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	203.40	0.98	4	1	1	197.51	2.98
B16	206.03	0.71	0	2	1	205.08	0.47
B17	396.27	1.86	3	5	0	390.47	1.49
B18	401.93	0.26	0	1	0	397.89	1.02
B19	385.71	2.13	0	4	3	380.23	1.44
B20	388.73	7.05	6	15	3	386.71	0.52
B21	284.98	3.83	1	3	2	283.90	0.38
B22	300.17	26.81	3	4	2	296.98	1.08
B23	428.84	5.27	4	4	1	416.31	3.01
B24	453.79	3.14	3	3	0	437.52	3.72
B25	440.26	8.12	2	4	1	439.18	0.25
B26	468.39	2.40	1	2	0	457.16	2.46
B27	449.43	11.89	6	5	2	440.10	2.12
B28	467.70	28.73	1	4	0	458.44	2.02
B29	358.03	16.76	7	8	0	352.67	1.52
B30	366.90	21.55	2	3	3	364.71	0.60
B31	353.54	19.87	5	8	0	342.59	3.20
B32	360.39	24.91	4	11	1	358.06	0.65
B33	378.58	44.04	4	8	1	371.53	1.90
B34	395.07	264.45	15	16	5	386.17	2.31
B35	369.76	93.37	2	9	3	367.91	0.50
B36	389.20	73.06	7	8	0	378.21	2.91
B37	447.22	99.76	3	2	1	442.95	0.96
B38	475.54	238.96	12	6	3	456.95	4.07
			95	136	33		

Tabla 44. Variante 6. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	196.60	0.85	2	3	0	195.44	0.60
C40	203.49	0.77	1	3	0	201.22	1.13
C41	394.52	2.01	5	5	1	388.63	1.52
C42	396.06	0.95	0	2	1	381.39	3.85
C43	377.57	4.45	3	1	1	373.68	1.04
C44	392.45	1.20	2	2	0	388.75	0.95
C45	293.45	1.15	2	1	0	290.47	1.03
C46	298.87	3.20	2	5	0	296.54	0.79
C47	427.07	8.04	2	2	0	424.97	0.50
C48	437.23	2.04	1	1	1	433.82	0.79
C49	454.36	25.70	2	1	3	446.28	1.81
C50	465.37	12.93	3	2	1	454.94	2.29
C51	312.31	15.40	6	7	5	300.49	3.93
C52	319.35	18.36	6	8	4	305.57	4.51
C53	365.29	58.45	10	13	2	357.49	2.18
C54	372.98	38.96	1	4	2	362.74	2.82
C55	351.95	9.87	1	5	0	349.65	0.66
C56	359.97	59.35	6	12	2	349.91	2.88
C57	386.27	41.69	6	6	4	380.12	1.62
C58	372.84	134.79	13	11	3	372.84	0.00
C59	382.46	47.11	1	4	0	371.73	2.89
C60	389.54	68.63	4	8	0	377.38	3.22
C61	458.13	189.37	6	7	4	447.09	2.47
C62	460.29	110.22	7	8	1	454.32	1.31
			92	121	35		

Tabla 45. Variante 6. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	380.52	109.41	12	13	9	379.42	0.29
R02	1018.38	388.04	27	44	12	1014.78	0.36
R03	493.49	214.99	13	16	12	487.42	1.25
R04	1927.55	283.21	12	17	6	1926.21	0.07
R05	874.39	69.63	9	8	5	870.52	0.44
R06	873.47	600.00	22	18	15	567.23	53.99
R07	4136.32	370.02	22	24	4	4126.57	0.24
R08	352.85	600.00	11	10	13	287.36	22.79
R09	912.55	600.00	17	7	13	548.84	66.27
R10	1017.31	600.00	12	5	16	979.81	3.83
R11	1486.65	600.00	23	28	7	1483.53	0.21
R12	419.12	600.00	14	8	13	257.01	63.08
R13	245.76	99.36	11	15	4	236.33	3.99
R14	733.88	600.00	7	7	11	687.23	6.79
R15	216.21	579.75	25	40	16	209.85	3.03
R16	2419.42	101.16	11	14	5	2408.60	0.45
R17	509.35	600.00	33	34	15	403.54	26.22
R18	449.57	600.00	25	27	16	447.33	0.50
R19	1209.08	41.92	8	12	1	1206.89	0.18
R20	681.46	101.73	18	14	7	669.15	1.84
R21	2421.79	386.57	16	21	7	2416.19	0.23
R22	355.53	446.58	16	39	8	352.57	0.84
R23	503.50	600.00	12	5	12	463.85	8.55
R24	652.47	600.00	11	8	19	561.43	16.22
R25	488.24	110.29	12	11	7	488.10	0.03
R26	521.12	69.39	7	9	10	512.39	1.70
R27	866.76	600.00	10	10	16	773.03	12.12
R28	350.58	121.20	7	11	8	350.49	0.03
R29	600.59	600.00	10	11	13	567.16	5.89
R30	937.77	600.00	11	5	15	763.00	22.90
R31	707.06	582.30	15	19	13	698.84	1.18
R32	388.27	75.50	8	16	6	380.25	2.11
R33	2573.40	107.83	8	20	1	2563.60	0.38
R34	480.15	431.24	16	12	12	478.50	0.34
R35	441.90	217.47	21	16	15	437.49	1.01
R36	2290.68	104.83	4	16	1	2278.04	0.56
R37	138.41	517.45	40	21	22	132.40	4.54
R38	2021.31	267.52	18	26	4	2003.44	0.89
R39	305.16	334.80	10	25	13	292.33	4.39
R40	977.98	600.00	10	7	9	797.44	22.64
			594	669	411		

9.7 Resultados de variante 7

Tabla 46. Variante 7. Resultados de instancias de tipo B.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
B15	204.97	1.12	1	3	1	197.51	3.78
B16	205.69	1.35	3	4	2	205.08	0.30
B17	396.79	2.54	4	4	2	390.47	1.62
B18	408.51	1.29	1	2	1	397.89	2.67
B19	380.44	1.86	4	4	1	380.23	0.06
B20	387.83	1.53	3	3	1	386.71	0.29
B21	285.10	3.64	3	4	1	283.90	0.42
B22	300.16	2.24	1	2	3	296.98	1.07
B23	421.00	5.12	5	3	0	416.31	1.13
B24	445.26	6.74	5	3	3	437.52	1.77
B25	442.04	28.49	1	4	3	439.18	0.65
B26	466.65	2.05	0	1	1	457.16	2.08
B27	446.55	40.46	10	5	2	440.10	1.47
B28	467.02	10.64	11	6	1	458.44	1.87
B29	356.28	44.96	5	8	0	352.67	1.02
B30	370.09	29.29	10	15	1	364.71	1.47
B31	350.27	41.85	8	13	5	342.59	2.24
B32	358.06	37.39	3	11	1	358.06	0.00
B33	379.06	46.54	5	12	1	371.53	2.03
B34	396.83	37.42	7	6	2	386.17	2.76
B35	376.29	78.90	5	8	2	367.91	2.28
B36	386.30	99.70	13	11	0	378.21	2.14
B37	451.00	214.59	12	12	3	442.95	1.82
B38	464.09	300.00	11	31	5	456.95	1.56
			131	175	42		

Tabla 47. Variante 7. Resultados de instancias de tipo C.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
C39	195.44	1.26	2	5	0	195.44	0.00
C40	207.16	3.29	8	3	2	201.22	2.95
C41	392.93	4.87	9	4	4	388.63	1.11
C42	400.05	1.54	5	1	0	381.39	4.89
C43	377.98	2.86	5	5	1	373.68	1.15
C44	390.53	1.40	1	3	1	388.75	0.46
C45	293.13	3.67	0	4	2	290.47	0.92
C46	297.80	2.39	1	2	1	296.54	0.43
C47	428.68	34.31	11	5	2	424.97	0.87
C48	445.09	5.58	5	3	3	433.82	2.60
C49	449.10	24.35	0	1	1	446.28	0.63
C50	457.67	4.14	3	3	1	454.94	0.60
C51	305.26	54.97	9	7	2	300.49	1.59
C52	309.32	43.60	9	6	2	305.57	1.23
C53	362.31	54.52	6	13	2	357.49	1.35
C54	381.97	69.12	13	8	6	362.74	5.30
C55	355.51	63.93	5	13	5	349.65	1.67
C56	358.49	51.32	8	7	2	349.91	2.45
C57	385.72	70.04	9	7	1	380.12	1.47
C58	393.51	154.41	16	17	2	372.84	5.55
C59	378.97	61.02	6	6	0	371.73	1.95
C60	381.91	175.69	15	20	6	377.38	1.20
C61	458.56	300.00	16	11	8	447.09	2.57
C62	463.02	83.07	5	5	1	454.32	1.91
			167	159	55		

Tabla 48. Variante 7. Resultados de instancias de tipo R.

Instancia	Makespan (h)	Ttb (s)	Nº de Mejoras Flip	Nº de Mejoras Swap	Nº de Mejoras Job Alloc	Makespan metaheurística ppal	DEV (%)
R01	388.77	388.77	12	11	8	379.42	2.47
R02	1021.48	102.21	24	33	6	1014.78	0.66
R03	509.78	264.49	9	24	14	487.42	4.59
R04	1928.35	326.21	14	25	6	1926.21	0.11
R05	871.62	600.00	8	15	7	870.52	0.13
R06	747.39	600.00	27	16	15	567.23	31.76
R07	4126.94	299.93	6	14	3	4126.57	0.01
R08	496.32	600.00	15	7	13	287.36	72.72
R09	789.35	600.00	21	17	13	548.84	43.82
R10	1018.24	600.00	7	9	12	979.81	3.92
R11	1490.42	600.00	34	37	11	1483.53	0.46
R12	405.17	600.00	13	15	5	257.01	57.65
R13	236.33	575.72	10	10	7	236.33	0.00
R14	799.90	600.00	10	6	14	687.23	16.40
R15	222.18	388.87	7	21	14	209.85	5.88
R16	2414.08	431.11	7	13	4	2408.60	0.23
R17	467.64	600.00	26	22	20	403.54	15.89
R18	452.30	499.47	18	30	20	447.33	1.11
R19	1210.79	322.68	11	22	2	1206.89	0.32
R20	677.25	112.10	14	20	10	669.15	1.21
R21	2420.86	600.00	20	26	12	2416.19	0.19
R22	359.04	436.94	19	28	8	352.57	1.84
R23	489.05	600.00	11	10	12	463.85	5.43
R24	623.20	600.00	11	6	14	561.43	11.00
R25	488.24	169.09	3	14	11	488.10	0.03
R26	517.41	204.59	11	17	7	512.39	0.98
R27	802.20	600.00	6	12	18	773.03	3.77
R28	350.48	114.10	6	11	11	350.49	0.00
R29	593.17	600.00	15	5	14	567.16	4.59
R30	899.39	600.00	7	16	8	763.00	17.87
R31	700.31	396.36	15	30	15	698.84	0.21
R32	388.27	139.69	14	5	12	380.25	2.11
R33	2572.57	139.64	6	12	13	2563.60	0.35
R34	481.35	341.98	7	9	6	478.50	0.59
R35	444.23	176.50	7	4	17	437.49	1.54
R36	2295.52	118.00	14	19	14	2278.04	0.77
R37	132.40	455.20	3	9	3	132.40	0.00
R38	2019.83	203.02	12	6	21	2003.44	0.82
R39	303.32	398.23	13	7	16	292.33	3.76
R40	841.03	532.27	9	8	12	797.44	5.47
			502	621	448		

