

# Trabajo Fin de Máster

## Organización Industrial y Gestión de Empresas

Aplicación de un algoritmo memético al problema de planificación de la producción en fabricación aditiva considerando orientaciones alternativas en las piezas

Autor: Pablo Kovács Martínez

Tutor: José Carlos Molina Gómez

**Dpto. Organización Industrial y Gestión de  
Empresas I**

**Escuela Técnica Superior de Ingeniería**

Sevilla, 2022





Trabajo Fin de Máster  
Máster en Organización Industrial y Gestión de Empresas

# **Aplicación de un algoritmo memético al problema de planificación de la producción en fabricación aditiva considerando orientaciones alternativas en las piezas**

Autor:

Pablo Kovács Martínez

Tutor:

José Carlos Molina Gómez

Profesor ayudante doctor

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Máster: Aplicación de un algoritmo memético al problema de planificación de la producción en fabricación aditiva considerando orientaciones alternativas en las piezas

Autor: Pablo Kovács Martínez

Tutor: José Carlos Molina Gómez

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

# Agradecimientos

---

*A mi familia y amigos*

*A mis compañeros y maestros*



# Resumen

---

El estudio realizado se ha basado en la experimentación de instancias de problemas de fabricación aditiva para la optimización de costes del problema de planificación de la producción basado en la asignación de piezas a estructuras para ser fabricadas en máquinas de impresión en 3D. El problema ha considerado la posibilidad de incorporar diferentes orientaciones en las piezas a introducir aumentando la complejidad del problema y acercándose a una situación real en la industria. Esta componente de orientaciones extra permite reducir los costes totales de producción sobre todo en problemas de gran complejidad.

Para la resolución de este problema, se utiliza un algoritmo memético compuesto por la hibridación de un algoritmo de Sistema de Colonia de Hormigas y un algoritmo de Búsqueda Descendente en Entornos Variables (VND) para la búsqueda local. Con el fin de obtener una solución inicial al problema, se ha desarrollado previamente una heurística de inserción semiparalela. Estos métodos son de gran interés científico actualmente pero no han sido evaluados hasta ahora de esta manera en los problemas de fabricación aditiva permitiendo varias orientaciones a las piezas.

Ha sido necesario encontrar los valores óptimos para la heurística semiparalela en una experimentación previa para reducir los tiempos de ejecución. Posteriormente se ha comprobado la efectividad de la búsqueda local realizando una comparación de los resultados obtenidos analizando el uso en solitario del ACS sin postprocesado respecto al Mem-ACS completo.

La comparación de resultados se ha analizado para los problemas considerando 1, 2 y 3 orientaciones y se ha comprobado que el Mem-ACS ofrece mejores resultados de coste para problemas que aumentan su complejidad computacional. Esta complejidad está marcada por el número de piezas, número de orientaciones disponibles y número de máquinas paralelas utilizadas.





# Abstract

---

The study carries out the experimentation of additive manufacturing problem instances based on cost optimisation over production problems collected and augmented from the literature review. It is based on the assignment of parts to structures in order to be manufactured on 3D printing machines. The problem has considered the possibility of several orientations for each part to be introduced. This characteristic increases the complexity of the problem, and it approaches the study to a real situation in the industry. This component of extra orientations allows reducing the total production costs of the production lines especially in problems of high complexity.

A memetic algorithm composed of an initial phase of semi-parallel heuristics is used to provide the initial solution and continues with the optimisation phase by using the Ant Colony System algorithm. It is later post-processed by a local VND search. These methods are nowadays of great scientific interest but have not been evaluated so far within the additive manufacturing problems allowing various part orientations.

It has been necessary to find the optimal values for the semi-parallel heuristic in a previous experimentation in order to reduce execution times. Subsequently, the use of the ACS alone without post-processing has been analysed and compared to the complete Mem-ACS.

The study results in better cost results by using Mem-ACS for problems that increase their computational complexity considering 1, 2 and 3 orientations. This complexity is measured by the number of parts, the number of available orientations and the number of parallel machines used.



# Índice

---

<b>Agradecimientos</b>	<b>vi</b>
<b>Resumen</b>	<b>viii</b>
<b>Abstract</b>	<b>x</b>
<b>Índice</b>	<b>xii</b>
<b>Índice de Tablas</b>	<b>xiv</b>
<b>Índice de Figuras</b>	<b>xvi</b>
<b>1 Introducción a la fabricación aditiva</b>	<b>1</b>
1.1 <i>Objetivos del trabajo</i>	1
1.2 <i>Estructura del trabajo</i>	2
1.3 <i>Origen de la fabricación aditiva</i>	3
1.4 <i>Categorías de fabricación aditiva</i>	4
<b>2 Estado del arte</b>	<b>5</b>
2.1 <i>Tipos de problemas</i>	5
2.2 <i>Clasificación de los problemas</i>	7
2.3 <i>Revisión bibliográfica</i>	9
<b>3 Descripción del problema</b>	<b>12</b>
3.1 <i>Orientación de las piezas</i>	13
3.2 <i>Características de las máquinas de FA</i>	13
3.3 <i>Modelo</i>	14
3.3.1 <i>Conjuntos e índices</i>	14
3.3.2 <i>Parámetros</i>	14
3.3.3 <i>Variable de decisión</i>	15
3.3.4 <i>Variabes auxiliares</i>	15
3.3.5 <i>Función objetivo</i>	15
3.3.6 <i>Restricciones</i>	16
<b>4 Método de resolución</b>	<b>17</b>
4.1 <i>Introducción a los métodos de resolución</i>	17
4.2 <i>Heurística constructiva semiparalela</i>	18
4.3 <i>Metaheurística (Mem-ACS)</i>	24
4.3.1 <i>Búsqueda global - Sistema de colonia de hormigas (ACS)</i>	25
4.3.2 <i>Búsqueda local - Descenso por vecindad variables (VND)</i>	28
<b>5 Experimentación</b>	<b>31</b>
5.1 <i>Generación de instancias</i>	31
5.2 <i>Ajuste de parámetros del algoritmo</i>	34
5.2.1 <i>Ajuste de parámetros de la función "Greedy"</i>	34
5.2.2 <i>Ajuste de parámetros del algoritmo memético</i>	35
5.3 <i>Resultados computacionales del algoritmo memético ACS</i>	38
5.3.1 <i>Comparación para problemas con 1 orientación</i>	40
5.3.2 <i>Comparación para problemas con 2 orientaciones</i>	45
5.3.3 <i>Comparación para problemas con 3 orientaciones</i>	46

5.3.4	Eficiencia del algoritmo memético	48
5.3.5	Efecto de la multiorientación de piezas	53
<b>6</b>	<b>Conclusiones</b>	<b>56</b>
6.1	<i>Líneas futuras de investigación</i>	56
<b>7</b>	<b>Anexos</b>	<b>57</b>
7.1	<i>Ajuste de parámetros alfa</i>	57
7.2	<i>Ajuste de parámetros primera fase (q0 y beta)</i>	58
7.3	<i>Ajuste de parámetros segunda fase (N y M)</i>	59
7.4	<i>Código de generación de instancias</i>	60
7.5	<i>Código de extracción de alfas para el ajuste de parámetros</i>	62
7.6	<i>Código de extracción de costes totales</i>	63
	<b>Referencias</b>	<b>65</b>

# ÍNDICE DE TABLAS

---

Tabla 1. Categorías y características de la FA (X. Zhang & Liou, 2021)	5
Tabla 2. Índices del modelo	14
Tabla 3. Parámetros del modelo	15
Tabla 4. Variable de decisión	15
Tabla 5. Variables auxiliares del modelo	15
Tabla 6. Valores para la generación de instancias	32
Tabla 7. Ejemplo cálculo de nuevas orientaciones para el problema n°1 de Li et al., (2017)	32
Tabla 8. Resumen de rangos de alfa	35
Tabla 9. Fase 1   Instancia 14   3 orientaciones	36
Tabla 10. Fase 1   Instancia 15   2 orientaciones	36
Tabla 11. Fase 1   Instancia 16   3 orientaciones	36
Tabla 12. Fase 1   Instancia 17   2 orientaciones	37
Tabla 13. Fase 1   Desviación media	37
Tabla 14. Fase 2   Instancia 14   3 orientaciones	37
Tabla 15. Fase 2   Instancia 15   2 orientaciones	38
Tabla 16. Fase 2   Instancia 16   3 orientaciones	38
Tabla 17. Fase 2   Instancia 17   2 orientaciones	38
Tabla 18. Fase 2   Desviación media	38
Tabla 19. Características de las instancias	39
Tabla 20. Análisis GAP Heurística Semiparalela   1 orientación	40
Tabla 21. Comparación resultados Heurística Semiparalela   1 orientación	41
Tabla 22. Análisis GAP Mem-ACS   1 orientación	42
Tabla 23. Comparación resultados Mem-ACS   1 orientación	43
Tabla 24. Análisis GAP Mem-ACS   2 orientaciones	45
Tabla 25. Comparación resultados   2 orientaciones	45
Tabla 26. Análisis GAP Mem-ACS   3 orientaciones	47
Tabla 27. Comparación resultados   3 orientaciones	47
Tabla 28. Análisis GAP Mem-ACS vs. ACS	49
Tabla 29. Eficiencia Mem-ACS	51
Tabla 30. Mejora de coste por unidad de volumen   Mem-ACS	53



# ÍNDICE DE FIGURAS

---

Ilustración 1. Ejemplo trabajo o estructura (Fera, Macchiaroli, et al., 2018)	6
Ilustración 2. Nesting and scheduling (Oh et al., 2020)	6
Ilustración 3. Subproblemas nesting and scheduling (Chergui et al., 2018)	7
Ilustración 4. Skyline nesting (Chergui et al., 2018)	7
Ilustración 5. Part/Level/Machine (Oh et al., 2020)	8
Ilustración 6. Orientaciones de pieza (Y. Zhang et al., 2017)	12
Ilustración 7. Posibles orientaciones de una pieza (Y. Zhang et al., 2017)	13
Ilustración 8. Formación de jobs 1	22
Ilustración 9. Formación de jobs 2	23
Ilustración 10. Formación de jobs 3	23
Ilustración 11. Formación de jobs 4	24
Ilustración 12. Formación de jobs 5	24
Ilustración 13. Cambio de orientación de las piezas dentro de un mismo job	29
Ilustración 14. Inserción de piezas en otro job de la misma máquina	29
Ilustración 15. Intercambio de dos piezas entre jobs distintos pertenecientes a la misma máquina	29
Ilustración 16. Intercambio de una pieza de un job por dos piezas de otro job	30
Ilustración 17. Intercambio de dos piezas de un job por otras dos de otro job	30
Ilustración 18. Fichero entrada sección inicial	33
Ilustración 19. Fichero entrada sección volumen	33
Ilustración 20. Fichero entrada sección orientaciones	33
Ilustración 21. Fichero entrada sección máquinas	34
Ilustración 22. GAP Mem-ACS vs. ACS   1 orientación	49
Ilustración 23. GAP Mem-ACS vs. ACS   2 orientaciones	50
Ilustración 24. GAP Mem-ACS vs. ACS   3 orientaciones	50
Ilustración 25. Representación u.m./u.vol. Mem-ACS	54
Ilustración 27. Representación u.m. Mem-ACS	55







# 1 INTRODUCCIÓN A LA FABRICACIÓN ADITIVA

---

La fabricación aditiva (FA), comúnmente conocida como impresión en tres dimensiones (3D), es el proceso productivo utilizado en las industrias actuales para la fabricación de piezas y productos mediante la superposición de capas de material. La FA consiste en la creación de estructuras tridimensionales mediante la sedimentación de capas de material. El modelo a fabricar se construye mediante diseño asistido por ordenador (CAD, *Computer-Aid Design*) o escaneado de la pieza original, y es incorporado a la máquina de impresión para su ejecución. Durante las últimas décadas del siglo XIX la FA comienza a desarrollarse y empiezan a verse las primeras aplicaciones manufactureras y comerciales tales como en la medicina, educación, arquitectura, cartografía, juguetes y ocio.

La tecnología de la FA es uno de los pilares de la revolución industrial que forma parte de la Industria 4.0. La 4ª revolución industrial que comenzó en 2011 en Alemania sienta las bases de los modernos métodos de fabricación actuales donde Internet de las Cosas (IoT, *Internet of Things*), Fabricación Aditiva, Robótica y Realidad Aumentada son algunos de los grandes pilares del nuevo entorno de fabricación.

Todos estos avances ya han llegado y están en plena ebullición. Esto hace que el presente estudio contribuya de manera significativa al progreso del conocimiento industrial y la optimización de procesos industriales.

Todas las tecnologías que influyen en el éxito de las fábricas son muy relevantes en la mejora de tiempos de ejecución, reducción de costes y aumento de beneficios empresariales, pero sin una correcta programación de la producción y secuenciación de operaciones no será posible explotar el total potencial que nos brindan. Es por ello que este trabajo desarrolla un problema de optimización de las operaciones involucradas en una cadena de producción que hace uso de máquinas heterogéneas de FA en paralelo, teniendo en cuenta la existencia de distintas orientaciones de las piezas a fabricar y su consiguiente optimización del proceso productivo.

## 1.1 Objetivos del trabajo

El principal objetivo de este trabajo es introducir y resolver el problema de planificación de la producción en FA en entornos de máquinas heterogéneas en paralelo con orientaciones alternativas en las piezas.

El problema consiste en asignar y localizar piezas en agrupaciones o estructuras (jobs) para ser fabricadas haciendo uso de máquinas de FA dispuestas en paralelo. El problema incorpora atributos presentes en entornos

industriales reales como son la consideración de máquinas heterogéneas o la posibilidad de disponer las piezas en diferentes orientaciones alternativas a la hora de ser fabricadas en el job, que añaden una componente de dificultad al problema

El problema se presenta mediante un modelo de programación lineal y es resuelto mediante un algoritmo memético basado en el Sistema de Colonias de Hormigas (ACS, *Ant Colony System*), metaheurística que ha sido ampliamente aplicada en problemas de optimización combinatoria.

Para llevar a cabo este objetivo, se ha realizado el estudio de múltiples artículos científicos que analizan el problema de planificación de la producción en FA partiendo de problemas básicos hasta los más complejos.

Dichos artículos han servido de base para plantear el problema mediante un modelo matemático formulado con técnicas de programación lineal. Por otra parte, debido a la novedad del problema presentado, se ha realizado una recopilación de instancias existentes que resuelven variantes del problema en la literatura científica, con el fin de elaborar una nueva batería de problemas. Con el fin de validar el comportamiento y rendimiento del algoritmo implementado, se han comparado las soluciones obtenidas por el algoritmo con soluciones óptimas o aproximadas de la literatura científica.

El motivo por el que se ha elegido este problema es por la necesidad de avanzar en el estudio de problemas de FA identificado en múltiples artículos de la literatura. El campo de estudio está sufriendo un fuerte desarrollo en los últimos años, pero no ha culminado de aprovechar todo su potencial. La componente de multiorientación estudiada supone un aporte distintivo en la consecución de los objetivos analizados.

Los objetivos del Trabajo Fin de Máster (TFM) son los siguientes:

1. Analizar el problema que se aborda y problemas relacionados en la literatura.
2. Adecuación y generación de instancias a resolver en el estudio mediante Excel, Visual Basic para Aplicaciones (VBA) y Python.
3. Ejecutar el algoritmo en Visual Studio mediante lenguaje C++.
4. Análisis de datos obtenidos de la resolución de las instancias.
5. Comparación de resultados y extracción de conclusiones.

La contribución principal de este TFM es la consideración de la multiorientación de las piezas para su agrupación en jobs o estructuras y la elaboración de un algoritmo memético que resuelva el problema de planificación de la producción en FA de forma eficiente.

## 1.2 Estructura del trabajo

La estructura del Trabajo Fin de Máster está dividido en 6 capítulos y un anexo, descritos a continuación:

- Capítulo 1: Introducción a la fabricación aditiva.

El primer apartado es un capítulo introductorio de descripción del propósito del TFM y se comenta brevemente su interés de estudio, su aplicabilidad, objetivos, ámbito, origen tecnológico y clasificación de la tecnología.

- Capítulo 2: Estado del arte.

Esta sección surge con el propósito de contextualizar la naturaleza del problema de FA dentro del marco de la programación de la producción. Se describen los tipos de problemas relacionados con el tratado, comentando los conceptos teóricos para la resolución del problema objeto de estudio. También incluye una revisión de la literatura relacionada y que se ha tomado como base del trabajo.

- Capítulo 3: Descripción del problema.

Este capítulo describe en profundidad el problema a resolver. Se exponen sus características principales en cuanto al entorno, restricciones y objetivos, además de comentar cualquier aspecto que ayude a comprender el problema en cuestión. Incluye el modelo de programación lineal sobre el que se ha basado el algoritmo del estudio.

- Capítulo 4: Método de resolución.

El cuarto capítulo describe con detalle las componentes del algoritmo memético desarrollado en el TFM para la resolución del problema.

- Capítulo 5: Experimentación.

Este apartado recoge una descripción detallada de los resultados computacionales obtenidos tras la ejecución del algoritmo en los casos de estudio. Se muestra el conjunto de pruebas realizadas, y se realiza la comparativa de los resultados obtenidos en cuanto al valor de coste analizado y al tiempo computacional del algoritmo

- Capítulo 6: Conclusiones.

Para finalizar, este último capítulo recoge las conclusiones extraídas tras el estudio, así como las posibles futuras líneas de investigación que se podrían plantear.

Al final del documento se encuentran las referencias bibliográficas, además de un Anexo que contiene características de las instancias utilizadas, resultados de experimentación y los códigos Python para el tratamiento de datos.

### 1.3 Origen de la fabricación aditiva

Los inicios de esta forma de fabricar se remontan a 1972, año en el que Pierre Ciraud obtiene la patente de un sistema basado en la creación de objetos a base de capas de metal obtenidas mediante el fundido selectivo de partículas de polvo utilizando electrones, láser o impulsos de plasma. En 1980, Hideo Kodama del Instituto Municipal de Investigación Industrial de Nagoya, en Japón, presentó el método de modelado sólido en 3D mediante el sistema conocido como Prototipado Rápido. En 1982 Herbert presentó el primer sistema láser de estereolitografía (SLA) basado en CAD. Un par de años después, en 1984, Charles Hull obtuvo la primera patente de estereolitografía y en 1986 fundó la compañía denominada 3D Systems, primera empresa de FA que utilizó la SLA en sus procesos productivos (Murr & Johnson, 2017). Esta empresa, a día de hoy, sigue desempeñando estas funciones con sede en Rock Hill, South Carolina.

El proceso de SLA consiste en el curado de resina polimérica líquida al exponerla a luz ultravioleta (Sreerag et al., 2021). Estos acontecimientos suponen las bases de los siguientes avances en FA que se desarrollarán durante la siguiente década y que continúan en pleno desarrollo hoy en día.

En 1991, se comercializaron tres tecnologías de FA, entre ellas el modelado de deposición fundida (FDM, *Fused Deposition Modelling*) de Stratasy, el curado en tierra sólida (SGC, *Solid Ground Curing*) de Cubital, y la fabricación de objetos laminados (LOM, *Laminated Object Manufacturing*) de Helisys. El sinterizado selectivo por láser (SLS, *Selective Laser Sintering*) de la empresa DTM (ahora parte de 3D Systems) y el sistema de estereolitografía Soliform de Teijin Seiki se pusieron en el mercado en 1992. Soligen comercializó en 1993 la fundición de producción directa de cáscaras (DSPC, *Direct Shell Production Casting*) mediante un mecanismo de inyección de tinta cuyo proceso fue patentado por el Instituto de Tecnología de Massachusetts (MIT). Ese mismo año el sistema SLA de Denken, que utilizaba láser de estado sólido, fue uno de los primeros en encajar en una mesa de trabajo y que además fue introducido a un precio bajo, en comparación con otros sistemas SLA que había en el mercado. 3D Systems y Ciba comercializaron también en 1993 su primer producto de resina epoxi.

Una gran variedad de productos comerciales acompaña a las invenciones mencionadas durante los siguientes años. El 1996 estuvo marcado como el año en el que se pusieron en el mercado las primeras impresoras 3D como Genesys, de Stratasy, basada en una tecnología desarrollada por el Centro de Investigación Watson de IBM; Actua 2100 de 3D Systems dedicada a la deposición por capas de cera; o Z402 de Z Corp. destinada al modelado conceptual.

Durante el año 2000 la compañía israelí denominada Objet Geometries lanzó una impresora 3D bajo el nombre de Quadra, que depositaba y solidificaba fotopolímero. También se pudo ver durante el mismo año grandes lanzamientos como los de Sanders Prototype (actualmente Solidscape) con PatterMaster, la de Precision Optical Manufacturing con su proceso de deposición directa de metal o la de Z Corp. como la primera impresora 3D multicolor. (Wohlens & Gornet, 2014).

El desarrollo de esta tecnología no ha parado y ha tenido un avance exponencial durante los siguientes años hasta nuestros días. Actualmente podemos ver una gran cantidad de versiones de sobremesa de distintas marcas a precios accesibles para todos los bolsillos. Esta evolución tecnológica ha permitido el desarrollo exponencial de estas máquinas y de los procesos productivos relacionados con ella suponiendo una componente diferencial en la industria de manufactura que requiere de exhaustivos métodos de optimización y análisis que permitan la evolución hacia la industria moderna o Industria 4.0.

## 1.4 Categorías de fabricación aditiva

Debido a la gran evolución que ha experimentado la FA desde sus inicios y a la diversidad de usos que puede tener, es necesario estandarizar y definir una clasificación clara que facilite la caracterización de esta tecnología y la comunicación entre los profesionales que hagan uso de ellas a nivel mundial. Analizando la terminología estándar desarrollada en la Norma Española UNE-EN ISO/ASTM 52900:2015, aprobada en España a Noviembre de 2017, podemos encontrar las siguientes siete categorías de FA en la página 9:

1. Proyección de aglutinante (BJ) - “proceso en el que un agente líquido aglutinante se deposita selectivamente para unir materiales en polvo”.
2. Deposición de energía focalizada (DED) - “proceso en el cual se utiliza energía térmica focalizada para unir materiales mediante fusión, a medida que se depositan. La energía térmica focalizada significa que una fuente de energía (por ejemplo, láser, haz de electrones o arco de plasma) se enfoca o concentra para fundir los materiales que se están depositando.”
3. Extrusión de material (ME, FDM) - “proceso en el cual el material se dispensa selectivamente a través de una boquilla o un orificio.”
4. Proyección de material (DOD) - “proceso en el cual se depositan selectivamente gotas del material de fabricación. Como ejemplo de estos materiales se incluyen los fotopolímeros y las ceras.”
5. Fusión de lecho de polvo (PBF, SLS, DMLS, SLM, EBM) - “proceso en el cual la energía térmica funde selectivamente ciertas zonas de un lecho de polvo.”
6. Laminado de hojas - “proceso en el cual el material en forma de láminas u hojas se une para formar un objeto.”
7. Fotopolimerización en tanque o cuba (VP, SLA, DLP) - “proceso en el que el fotopolímero líquido se cura selectivamente en una cuba mediante polimerización activada por luz.”

X. Zhang & Liou, (2021) realizan una tabla resumen de los procesos de FA y sus características basándose en las categorías definidas en la norma UNE-EN ISO/ASTM 52900:2015. A continuación, se presenta la tabla que expone las categorías junto con las tecnologías utilizadas, fuente de energía, materiales, ventajas y desventajas de cada una de ellas.

Tabla 1. Categorías y características de la FA (X. Zhang & Liou, 2021)

Categorías	Tecnologías	Fuente de energía	Materiales	Ventajas	Desventajas
Proyección de aglutinante	Proyección de aglutinante	Aglutinante/ Energía térmica	Polímero/ cerámica/ metal polvo	Amplia selección de materiales, impresión relativamente rápida	Largo postprocesamiento, porosidad en las piezas
Deposición de energía focalizada	Moldeado láser diseñado por red, deposición directa de metal, deposición de metal por láser, revestimiento metálico por láser, consolidación por láser	Láser	Metal/ cerámica polvo	Reparación de componentes desgastados, multimaterial (functionally graded material (FGM))	Baja precisión, bajo acabado superficial, tensión residual, requieren post mecanizado
Extrusión de material	Modelado por deposición fundida, fabricación de filamentos fundidos	Energía térmica	Termoplásticos (ABS, PLA, PC, nylon)	Económicos (tanto máquina como materia prima), multimaterial, fácil de operar	Mala resolución y acabado superficial, mala adherencia
Proyección de material	Gota bajo demanda, proyección de nanopartículas	PolyJet, Energía térmica	Resinas fotorpolímeras, metales, cerámica	Alta precisión, acabado superficial suave, multimaterial	Baja resistencia mecánica
Fusión de lecho de polvo	Sinterización directa de metales por láser, fusión por haz de electrones, fusión selectiva por láser, sinterización selectiva por láser	Láser, haz de electrones	Polímero/ cerámica/ metal polvo	Alta precisión, alta resolución, piezas totalmente densas, alta resistencia	Reciclaje de polvo, estructuras de soporte, material único, tensión residual
Laminado de hojas	Fabricación de objetos laminados	Láser	Plástico/ metal/ lámina de cerámica	Alto acabado superficial	Limitación del material
Fotopolimerización en tanque o cuba	Estereolitografía, procesamiento de luz digital, producción continua de interfaz líquida, impresión de polímeros a la luz del día	Luz ultravioleta	Resina fotosensible, cerámica	Alta precisión, buena resolución, automatización total	Sobrecurado prolongado post-procesamiento, composición única, alto coste de materiales





## 2 ESTADO DEL ARTE

---

El tema de la FA presenta un enfoque multidisciplinar ya que intervienen aspectos de diseño, ciencia de materiales, consumo de energía, gestión del ciclo de vida, tecnología láser, ciencia de computadores, gestión de la cadena de suministro y planificación de la producción (Costabile et al., 2016), (Fera et al., 2016, 2018). Como se menciona en el capítulo anterior, este trabajo se centra en la resolución de un problema de asignación de piezas a estructuras para ser fabricadas con máquinas de FA, permitiendo varias posibilidades de orientación de piezas y haciendo uso de máquinas en paralelo no idénticas. En esta sección se analizan los principales tipos de problemas en los que pueden clasificarse la programación de la producción en máquinas de FA y se hace un estudio de los artículos de la literatura que tratan de resolver este problema y han servido como referencia para la realización de este proyecto.

### 2.1 Tipos de problemas

El problema de programación de la producción en máquinas de FA consiste en fabricar un conjunto de partes (piezas) asociadas a pedidos de clientes en un conjunto de máquinas FA. Los estudios se realizan para el conjunto del total de partes, cada una de ellas con información de altura, superficie y volumen que ocupa. Todas las partes se agrupan en trabajos, también denominados estructuras, que serán realizados en una máquina FA. La agrupación de piezas en trabajos consiste en la ubicación y orientación de cada parte en una estructura de la máquina, de forma que no se pueda superar la superficie ni el volumen máximo de la estructura (Oh et al., 2020). Las partes de una estructura se suelen fabricar simultáneamente.

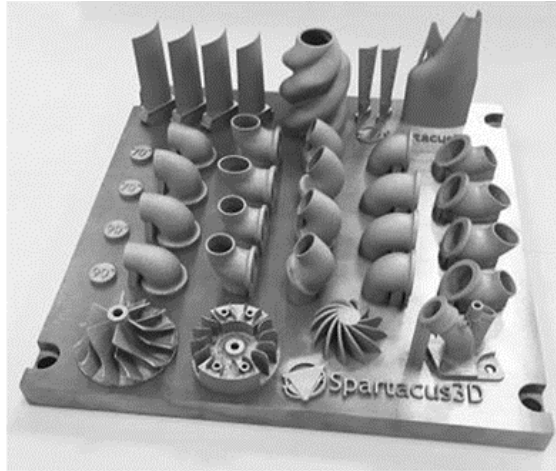


Ilustración 1. Ejemplo trabajo o estructura (Fera, Macchiaroli, et al., 2018)

El problema que estamos tratando se divide en dos subproblemas denominados “Nesting” y “Scheduling”.

El primer subproblema, denominado “Nesting”, consiste en agrupar el conjunto de las partes en un conjunto de trabajos. De esta forma, cada pieza resultará asignada con una determinada orientación a un determinado trabajo. Cada trabajo tendrá un tiempo de producción en función de las partes asignadas y una fecha de finalización en función de la máquina y el orden en la misma. En este primer subproblema se situarán de forma óptima las partes en los espacios de las estructuras específicas de cada máquina FA considerando las limitaciones de altura, superficie y volumen, además de los tiempos de procesado y las fechas de entrega.

Una vez formados los trabajos con la asignación de partes correspondientes, el segundo subproblema, denominado “Scheduling”, consiste en programar los trabajos en las máquinas. Cada máquina solo puede procesar un trabajo a la vez. Se define este problema como la asignación de cada trabajo a una única máquina y la ordenación en el tiempo (secuenciación) de cada trabajo en la máquina asignada considerando minimizar los retrasos.

El presente trabajo no aborda el problema de secuenciación como parte del “Scheduling” ya que su función objetivo está expresada en términos de coste de producción y no toma valores de tiempo en consideración. Por tanto, la componente de “Scheduling” no ha sido analizada en el trabajo. De esta forma, el problema abordado está enfocado en la optimización del problema de “Nesting”, agrupando las piezas en estructuras para la optimización de costes de producción en función de las características de las máquinas y las orientaciones de las piezas.

Las siguientes figuras resumen ambos subproblemas:

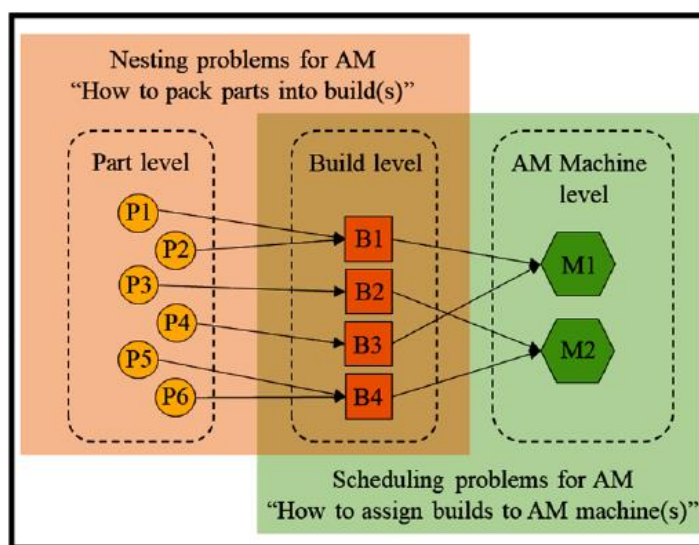


Ilustración 2. Nesting and scheduling (Oh et al., 2020)

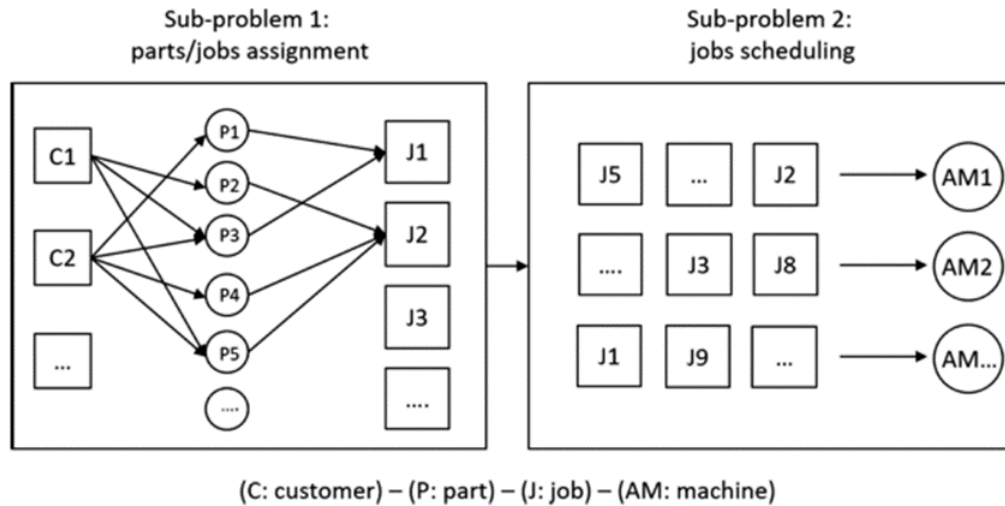


Ilustración 3. Subproblemas nesting and scheduling (Chergui et al., 2018)

La siguiente figura muestra un ejemplo de asignación de 4 partes a dos trabajos en 2D mediante el subproblema de “nesting” o “parts assignment”:

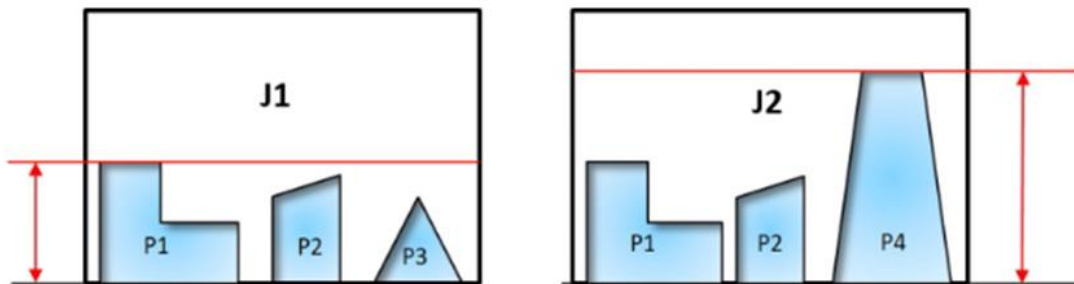


Ilustración 4. Skyline nesting (Chergui et al., 2018)

Los estudios realizados respecto a estos dos problemas varían respecto a gran variedad de parámetros y situaciones consideradas. En los apartados siguientes se analizará la literatura para observar los distintos casos de estudio desarrollados por los autores.

## 2.2 Clasificación de los problemas

Existen variantes para ambos problemas descritos en el apartado anterior. Según la clasificación de Oh et al., (2020), los problemas de “Nesting and scheduling” para FA tendrían 3 parámetros característicos: “Part level / Build level / AM machine level”.

- A nivel de Partes: todos los problemas consideran que existen “Multi-Part”, es decir, que hay múltiples tipos de partes a fabricar. Pero en algunas situaciones, cada máquina FA sólo puede fabricar 1 parte en cada estructura. En el primer caso se utiliza la clasificación “M” y en la segunda se utiliza “\_”.
- A nivel de Estructuras/”Builds”: los problemas se pueden clasificar en “Single-Build” cuando todas las estructuras son iguales (la misma agrupación para todas las partes), o “Multi-Build” cuando pueden ser diferentes estructuras.
- A nivel de Máquinas: los problemas se pueden clasificar en “Single-Machine” o “Multi-Machine”, si solo hay una máquina o son varias máquinas en paralelo. En este último caso, a su vez pueden ser:
  - “Identical Machines” si las máquinas son idénticas.
  - “Non-Identical Machines” si las máquinas son diferentes.

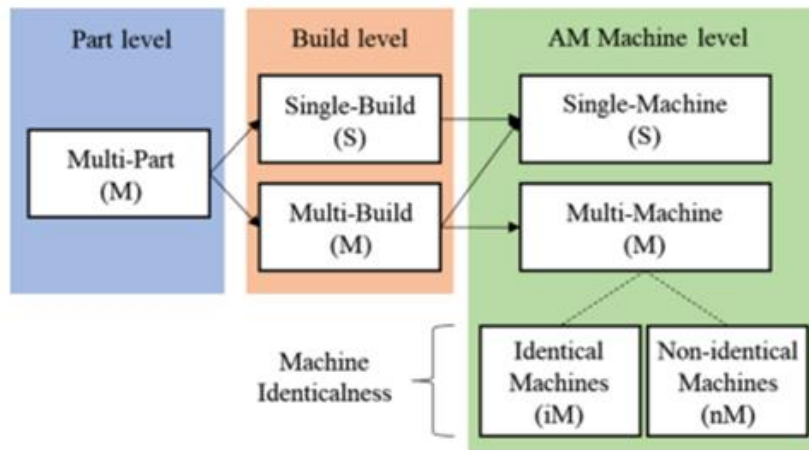


Ilustración 5. Part/Level/Machine (Oh et al., 2020)

La combinación de estas 3 dimensiones da como resultado la generación de seis clases de problemas a tratar, [M/S/S], [M/M/S], [M/M/iM], [M/M/nM], [-M/iM] and [-M/nM] (Oh et al., 2020).

- M/S/S: se centran en estudiar el problema “Nesting” en la parte de ubicación y orientación de las partes, similar al problema “bin-packing”.
- M/M/S: se centran en agrupar las partes en diferentes estructuras (nesting), que se realizan en la misma máquina FA, por lo que es importante la componente de scheduling.
- M/M/iM: como hay máquinas FA en paralelo idénticas, los “builds” pueden ser asignados a cualquier máquina, para luego secuenciarlas. Este problema se parece al caso de M/M/S ya que considera el mismo volumen máximo para los “builds”. En este caso se tendrán que tener en cuenta consideraciones de balanceo de carga.
- M/M/nM: como las máquinas FA en paralelo no son idénticas, este problema es el más complejo y tratará ambos enfoques. Ya que las estructuras varían en tamaño y forma, las partes se asignan a una estructura de una máquina concreta. Existe una gran variedad de criterios que se utilizan en este problema para asignar piezas a máquinas.
- -M/iM: cada máquina FA sólo procesa una parte en la estructura, por lo que no es necesario el nesting.
- -M/nM: hay que considerar que cada parte se procesa en una estructura, pero cada máquina es diferente.

Para cada tipo de problema existen criterios específicos que inciden en la resolución de los problemas. Los criterios típicos utilizados en los estudios son (Oh et al., 2020):

1. Dimensionalidad: 2D; 3D
2. Grado de libertad en rotación de las partes: ABC; C
3. Limitaciones en volumen de las estructuras: Bounded; Unbounded
4. Conjunto de partes a agrupar: Full; Subset
5. Métodos de generación de estructuras: Nested; Grouped; Created; Given
6. Modelos de secuenciación: Single-machine (SM); Parallel-machine (PM); Flow-shop (FS)
7. Criterio de secuenciación: Due-date (Du); Material (Ma); Quality (Qu)
8. Características de máquinas no idénticas: Location (Lo); Size (Si); Process parameter (Pr)

En los problemas exclusivos de Nesting (M/S/S) se aplican los 4 primeros criterios. En los problemas M/M/S y M/M/iM se suelen considerar los criterios 1, 2, 5, 6 y 7 con “Bounded” y “Full”. En los problemas M/M/nM además se considera el criterio 8. Finalmente, para los problemas -M/iM y -M/nM no se aplican los 4

primeros criterios ya que están enfocados en “scheduling”.

Respecto a los procesos de FA, los que se estudian en la literatura en este campo son (Oh et al., 2020):

- Binder jetting (BJ)
- Directed energy deposition (DED)
- Material extrusion (ME)
- Powder bed fusion (PBF)
- Vat photopolymerization (VP)

Respecto a metodologías que se aplican en la literatura para la resolución de los problemas, destacan:

- Metaheurísticas tales como Genetic algorithm (GA), Tabu search (TS) y Simulated annealing (SA)
- Heurísticas propias del problema
- Modelos de programación matemática
- SW comercial para Nesting

Finalmente, las funciones objetivo son muy diversas y dependen de los criterios seleccionados por los autores. En los casos de Nesting son objetivos relacionados con la maximización del “nesting rate” o la minimización del “build height”. En los casos de Scheduling, los objetivos se centran en la minimización de los retrasos/adelantos o de los costes o del makespan.

## 2.3 Revisión bibliográfica

Los estudios que se han llevado a cabo en la literatura han visto un incremento en los últimos años para analizar este problema desde diferentes enfoques y objetivos, con el fin de analizar, modelar y optimizar la cadena productiva. A pesar de esto, no se han visto gran cantidad de estudios que incluyan la máquina FA en el contexto del departamento de producción, quizás porque un mayor entendimiento respecto a esta nueva tecnología era necesaria previa a la experimentación directa (Fera et al., 2018). Actualmente está tomando gran relevancia debido a su extendido uso en la industria manufacturera y la implantación de modernos procesos de FA (Che et al., 2021), y un mayor número de artículos llevan a cabo experimentación en el ámbito de programación de la producción.

El estudio realizado por Li et al., (2017) es uno de los primeros trabajos sobre la planificación de la producción de máquinas FA distribuidas para satisfacer las demandas recibidas de clientes individuales en bajas cantidades. El objetivo principal de este artículo comprendía el subproblema de “scheduling” para una máquina única. Debido a la dificultad de resolución computacional del modelo propuesto en el estudio, se propusieron dos heurísticas de construcción semiparalela para resolver el problema, basándose en el objetivo del coste medio del material. Para el problema de “nesting” se utilizó un algoritmo sencillo de agrupación para generar trabajos/estructuras y asignar piezas. Los resultados obtenidos indicaron que ambas heurísticas propuestas se comportaron bien y proporcionaron valores de rendimiento prometedores dentro de tiempos computacionales razonables, sin una dominancia clara de una sobre la otra. Los resultados de las pruebas computacionales también demostraron la necesidad de desarrollar técnicas sofisticadas de planificación y programación para los problemas FA para 3DP.

Ese mismo año se publicó el artículo de Ransikarbum et al., (2017) que proponía un modelo de apoyo a las decisiones basado en la optimización multiobjetivo. Aplicaba el modelo a un conjunto de partes que se introducen al sistema con una orientación prefijada y que quieren ser distribuidas en un conjunto de impresoras 3D. Los objetivos tratados en el modelo son costes operativos, equilibrio de carga entre impresoras, retraso total y el número de piezas no fabricadas. Este modelo obtiene buenos resultados para problemas pequeños. Se recomienda en el estudio utilizar heurísticas y metaheurísticas en el caso de problemas más complejos.

Previamente se habían realizado estudios de la asignación de piezas a estructuras o “nesting” como el realizado por Zhang et al., (2016), que enfocaron su trabajo en la colocación de múltiples piezas de manera óptima en la estructura de una máquina teniendo en cuenta los objetivos definidos por el usuario. Este artículo se enfoca

únicamente al problema de “nesting” y evoluciona el problema de colocación en 2D. Su método constaba de un algoritmo en 2 etapas basado en un algoritmo genético. La primera de ellas es la selección de la orientación de la pieza, y la segunda, se encarga de la asignación a una estructura específica mediante una asignación más compacta reduciendo tiempo de fabricación y costes.

Posteriormente, sí podemos encontrar trabajos que aúnan ambos subproblemas. Chergui et al., (2018) presentan una heurística para analizar el problema de “nesting” y “scheduling” haciendo uso de máquinas idénticas en paralelo. Su algoritmo considera las fechas de entrega de los pedidos en la función objetivo del problema al minimizar el retraso total en la construcción de los trabajos con respecto a estas, a la vez que se maximiza el uso de las máquinas. Proponen una heurística propia basada en la regla de Earliest Due Date (EDD) que no ofrece resultados satisfactorios ya que exige de un elevado número de máquinas para conseguir un retraso del 0%. El estudio concluye con la necesidad de realizar métodos específicos que se adapten a los requerimientos de FA.

Fera et al., (2018) analizaron la optimización de la gestión de las operaciones con una única máquina planteando el problema matemático como una función multiobjetivo basada en el equilibrio de la optimización de la puntualidad/retraso y los costes de producción. Esta función multiobjetivo tiene dos restricciones principales: una para el volumen geométrico de la cámara de impresión y otra para respetar las fechas de entrega. Construido el modelo matemático como problema NP-HARD, en este trabajo se presentó una heurística tradicional modificada basada en Algoritmos Genéticos para respetar las características tecnológicas de FA. Este algoritmo de resolución ofreció resultados exitosos en resultados y tiempo computacional.

El trabajo realizado por Kucukkoc, (2019) introduce el problema de programación de la producción de máquinas FA con el objetivo de minimizar el tiempo de fabricación. No tiene en cuenta el proceso de nesting previo al scheduling. En cuanto a la configuración de las máquinas, se han considerado tres tipos de problemas: máquina única, máquinas idénticas en paralelo, y máquinas no idénticas en paralelo. Los problemas fueron definidos y modelados como problemas de programación lineal entera mixta (Mixed-Integer Linear Problem, MILP), y explorados mediante ejemplos numéricos para proponer modelos de resolución para estas situaciones. El estudio concluye con la necesidad de hacer uso de heurísticas y metaheurísticas debido a la dificultad computacional de los problemas al incrementar la dificultad de estos.

Fera et al., (2020), desarrollan una versión actualizada de un modelo de optimización presentado anteriormente (Fera et al., 2018) para la optimización de la gestión de las operaciones utilizando una sola máquina con la tecnología FA. En este estudio se presenta una nueva heurística basada en la búsqueda tabú (TS). Este algoritmo se aplicó sobre las mismas instancias utilizadas en el artículo previo para comparar los resultados obtenidos mediante los dos modelos comparando los indicadores de valor de la función objetivo, el valor de los costes de producción y el porcentaje de nivel de servicio. Los resultados ofrecen datos ventajosos a la búsqueda tabú en términos de rendimiento en la gestión de operaciones, en cambio el algoritmo genético (GA) obtiene mejores datos de tiempo de ejecución, significativamente al aumentar el número de piezas analizadas. Podríamos decir que el GA es más eficiente y aconsejable en problemas grandes, pero la TS nos ofrece mejores valores de función objetivo. Este autor propone la experimentación de otras heurísticas sobre estos problemas que aporten mayor visión sobre el método de resolución más indicado para la FA.

Zhang et al., (2020) en su trabajo desarrollan un algoritmo evolutivo mejorado para tres dimensiones (3DGA) combinando un algoritmo genético con una estrategia heurística para el “nesting”, haciendo uso de máquinas en paralelo. La eficiencia de la optimización se mejora mediante la introducción de un método de inicialización basado en las características del proceso de impresión 3D a través del desarrollo del correspondiente modelo de cálculo del tiempo. La experimentación realizada muestra mejores soluciones en comparación con otros algoritmos como el algoritmo genético simple, la optimización por enjambre de partículas y los algoritmos heurísticos.

Che et al., (2021) expusieron un artículo en el que se resolvía un problema de programación de máquinas de FA en paralelo no relacionadas con la selección de la orientación y las restricciones de empaquetamiento en 2D. El problema se presentó como MILP para el que se desarrolló un algoritmo de recocido simulado (SA) principalmente ejecutando el subproblema de “scheduling”. Respecto al subproblema de “nesting”, se desarrolló un algoritmo de búsqueda local aleatoria (RLS) con dos estrategias de ajuste basadas en la representación del skyline del patrón de empaquetado. Se hizo uso de la estructura de datos Trie para acelerar el procedimiento de empaquetado y se diseñaron cuatro métodos de post-optimización basados en greedy para

refinar la solución. El primero de ellos basado en makespan, el segundo en función del área utilizada de cada estructura, el tercero toma como referencia la variación de altura, y el cuarto utiliza el cambio de orientación de piezas dentro de un “build” para reducir su altura máxima. Se demuestra la eficacia de la selección de la orientación para ayudar a mejorar el tiempo de producción en FA.

Alicastro et al., (2021), centran su trabajo en el problema de programación de múltiples máquinas de FA sobre instancias generadas en los artículos de Li et al., (2017) y Kucukkoc, (2019), además de otros generados aleatoriamente. Proponen una metaheurística de Búsqueda Local Iterada de Aprendizaje con Refuerzo, basada en la implementación de una Búsqueda de Vecindario Variable (VNS) de Q-Learning, ya que, al aumentar el tamaño de los problemas, los tiempos computacionales también aumentan considerablemente. Llevan a cabo una comparación con las soluciones otorgadas por CPLEX y el algoritmo evolutivo propuesto por Zhang et al., (2020). Los resultados numéricos demuestran que la búsqueda local iterada de aprendizaje por refuerzo propuesta es capaz de obtener mejoras significativas en comparación con los otros enfoques de solución presentados. Los resultados del “nesting”, como, por ejemplo, las geometrías de las piezas o la orientación de los trabajos pueden afectar a los indicadores de rendimiento elegidos, como el tiempo de fabricación.

Este documento hace una contribución científica principal que consiste en el desarrollo de una metodología para resolver el problema de planificación de la producción en FA considerando varias orientaciones en las piezas. Para ello se ha desarrollado un algoritmo memético, que combina el Sistema de Colonia de Hormigas junto al Descenso de Vecindario Variable (VND, *Variable Neighbourhood Descent*) para la búsqueda local (Molina et al., 2020a). El problema se define formalmente mediante un modelo de programación lineal que considera las limitaciones del problema en función de las características de las máquinas y las orientaciones de las piezas de manera que define mejor las limitaciones de recursos en la planificación de la producción utilizando máquinas de FA en condiciones reales de la industria. Hasta donde sabemos, el algoritmo memético propuesto en este TFM aún no se ha introducido en la literatura científica.

### 3 DESCRIPCIÓN DEL PROBLEMA

---

A lo largo del capítulo se desarrolla la descripción del problema seleccionado para el estudio. La elección del tipo de problema a utilizar proviene del análisis de los artículos previos y la adaptación de problemas propuestos por otros autores. Se quiere ofrecer un mayor enfoque de la posibilidad de resolución de los problemas a gran escala, con gran número de piezas, que aporten un avance en los trabajos industriales referentes a la FA. El problema tratado en este trabajo se enmarca en la definición de M/M/nM, ya que introduce piezas distintas en estructuras con características diferentes y hace uso de máquinas FA en paralelo que no son idénticas. Debido a que las estructuras varían en tamaño y forma, las partes se asignan a una estructura de una máquina concreta. La característica peculiar del problema radica en la posibilidad de rotar las piezas dotándolas de capacidad para colocarlas en diferentes orientaciones, con su respectivo beneficio en términos de la función objetivo. Los criterios que se utilizan en este problema para asignar piezas a máquinas son: altura de la orientación, superficie proyectada en la base, volumen de la pieza y coste.

La investigación se basa en los modelos de Li et al., (2017) para el problema de la programación de trabajos cuando existen varias máquinas de FA no idénticas (non-identical AM machines), usando la función objetivo para minimizar los costes de producción y añadiendo una variante que se desarrolla en artículos como el de Zhang et al., (2017) que trata de buscar la mejor orientación de cada parte entre las posibles alternativas.

Las máquinas de FA realizan la fabricación de las partes capa a capa apoyadas sobre una bandeja, por lo que el modelo 3D debe ser orientado en una dirección concreta para definir cómo se fabricará cada capa. En la siguiente figura se muestra un ejemplo de 4 posibles orientaciones de una misma parte/pieza.

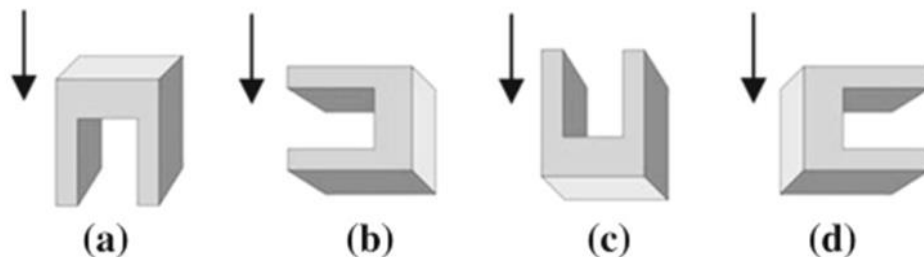


Ilustración 6. Orientaciones de pieza (Y. Zhang et al., 2017)



### 3.1 Orientación de las piezas

Puede existir un gran número de orientaciones posibles para colocar la pieza, pero únicamente un subconjunto previamente definido es utilizado para realizar la ejecución de instancias. La definición de este conjunto de orientaciones alternativas es la primera tarea a realizar basada en calidad, costes o tiempo. En nuestro problema será información de partida que utilice el algoritmo. Esto quiere decir que, existe un número de partes a fabricar con el mismo material de las que recibimos los datos de cada orientación como información de entrada. Cada orientación de una parte aporta la siguiente información:

- Superficie o área que ocupa la proyección sobre la bandeja de la estructura “A” (cm<sup>2</sup>)
- Altura que presenta la pieza desde la bandeja de la estructura “H” (cm)
- Volumen que ocupa en la estructura “V” (cm<sup>3</sup>). Este dato no depende de la orientación.

Se ofrece como ejemplo la siguiente imagen para facilitar la comprensión de las distintas orientaciones de una misma pieza. El ejemplo aporta la posibilidad de orientar la pieza en seis posiciones diferentes.

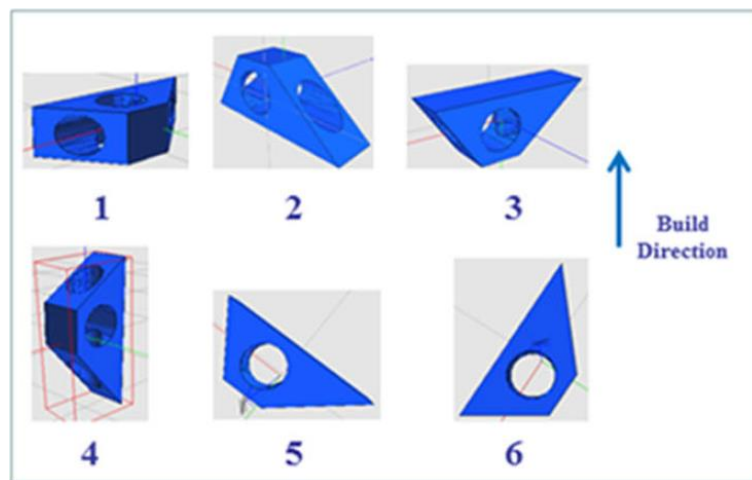


Ilustración 7. Posibles orientaciones de una pieza (Y. Zhang et al., 2017)

### 3.2 Características de las máquinas de FA

Para la fabricación de todas las partes existirá un número de máquinas aditivas que tengan unas características propias. Estas características específicas son las que determinarán si una pieza, en una orientación concreta, puede ser fabricada en la máquina. El problema estudiado consistirá en repartir todas las partes a fabricar entre diferentes estructuras, de forma que en cada estructura pueden ir una o varias partes, y cada parte se fabricará con una de sus orientaciones prácticas. Cada estructura se considera un trabajo a realizar en una de las máquinas disponibles, de forma que los tamaños de las bandejas y las alturas máximas en cada máquina pueden ser diferentes. Las características de cada máquina serán datos para introducir en nuestro problema y serán considerados información de entrada en nuestro algoritmo. Las características de tiempo y dimensiones utilizadas son las presentadas en el artículo publicado por Li et al., (2017):

- VTm (h/cm<sup>3</sup>): El tiempo que se emplea por unidad de volumen del material.
- HTm (h/cm): El tiempo que se emplea para cada capa de polvo por unidad de altura y que se repite hasta alcanzar la altura máxima.
- SETm (h): El tiempo de preparación de la máquina cada vez que se haga un trabajo.
- MAm (cm<sup>2</sup>): La superficie de la bandeja que sirve de base.
- MHm (cm): La altura máxima de procesamiento permitida en la máquina.

Además de agrupar las partes en trabajos, decidir la orientación de cada parte entre las alternativas y decidir en qué máquina va cada trabajo, hay que programar el orden de fabricación de cada trabajo en la máquina

asignada, aunque como se ha comentado anteriormente, los costes de planificación de la producción en AM son independientes de la secuencia de realización de los trabajos.

El análisis de la mejor solución se realiza en función del mejor resultado de coste. De esta manera la función objetivo utilizada está basada en minimizar el coste total de producción. El coste de producción de cada trabajo en cada máquina será la suma del coste de preparación de la máquina en cada cambio de trabajo (coste de set up), más el coste asociado a la aplicación de las capas de polvo, que depende de la altura máxima de las piezas asignadas a un trabajo, más el coste de material empleado y el de operación (coste de fusión de material) que dependerá del volumen total de las piezas. Para calcular los costes totales de producción, hay que comenzar definiendo los costes unitarios de producción (Li et al., 2017):

- $MC_m$  (u.m./cm<sup>3</sup>): El coste de material por unidad de volumen.
- $TC_m$  (u.m./h): El coste de operación por unidad de tiempo empleado en cada máquina.
- $HC_m$  (u.m./h): El coste de mano de obra por unidad de tiempo.

La selección de una determinada orientación para cada parte, la asignación de las partes a trabajos, y la selección de la máquina para hacer cada trabajo determinan finalmente el coste total de producción.

### 3.3 Modelo

El problema abordado se plantea como un modelo de programación lineal entera mixta (MILP, *Mixed Integer Linear Programming*) de planificación de la producción que minimiza los costes totales basándose en el problema propuesto por Li et al., (2017) y realizando las modificaciones oportunas para adaptarlo al formato de multiorientación.

#### 3.3.1 Conjuntos e índices

A continuación, se presenta la tabla que muestra los conjuntos utilizados en el modelo y sus respectivos índices.

Tabla 2. Índices del modelo

Conjunto	Definición	Índice
$G$	Piezas, partes o geometrías	$i = 1 \dots n_g$
$B$	Trabajos o estructuras ( <i>jobs</i> )	$j = 1 \dots n_b$
$M$	Máquinas de fabricación aditiva (FA)	$m = 1 \dots n_m$
$K$	Orientaciones prácticas de cada parte	$k = 1 \dots n_o(i)$

Sea una pieza  $i$  con orientación  $k$  a ser colocada dentro de job  $j$ , se evalúa la introducción de la pieza en función de parámetros específicos de la máquina  $m$ . La pieza  $i$  tendrá  $k$  orientaciones posibles para ser insertada en el job para conseguir la mejor colocación posible. Las posibles máquinas  $m$  tendrán distintas características que limitarán la forma de los jobs y será determinante para la inserción de las piezas.

#### 3.3.2 Parámetros

La siguiente tabla recoge los parámetros involucrados en el modelo junto a sus respectivas unidades de medida. Estos parámetros están relacionados con las características físicas de las piezas, jobs y máquinas de forma de definen la altura, área y volumen de éstas. Además, se añaden las componentes de tiempo y coste directamente aplicables sobre la evaluación de costes de producción.

Tabla 3. Parámetros del modelo

Parámetro	Definición	Unidades
$A_{ik}$	Área proyectada sobre la bandeja de la pieza $i$ con orientación $k$	cm
$H_{ik}$	Altura de la pieza $i$ con orientación $k$	cm <sup>2</sup>
$V_i$	Volumen de la pieza $i$	cm <sup>3</sup>
$VT_m$	Tiempo que se emplea por unidad de volumen del material en la máquina $m$	h/cm <sup>3</sup>
$HT_m$	Tiempo que se emplea para cada capa de polvo por unidad de altura en la máquina $m$	h/cm
$SET_m$	Tiempo de preparación de la máquina cada vez que se haga un trabajo en la máquina $m$	h
$MA_m$	Área máxima de la bandeja de cualquier estructura en la máquina $m$	cm <sup>2</sup>
$MH_m$	Altura máxima que se puede procesar en la máquina $m$	cm
MC	Coste por unidad de volumen del material	u.m./cm <sup>3</sup>
$TC_m$	Coste de operación por unidad de tiempo empleado en la máquina $m$	u.m./h
HC	Coste de mano de obra por unidad de tiempo	u.m./h

### 3.3.3 Variable de decisión

El modelo contiene la variable de decisión que relaciona los 4 conjuntos del modelo. Esta variable es de tipo binaria y adquiere el valor 1 en el caso de asignación de una pieza  $i$ , con orientación  $k$ , al trabajo  $j$  perteneciente a la máquina  $m$ .

Tabla 4. Variable de decisión

Variable	Definición	Tipo
$X_{ijmk}$	= 1 si la pieza $i$ se fabrica con orientación $k$ en el trabajo $j$ de la máquina $m$	Binaria

### 3.3.4 Variables auxiliares

Respecto a las variables auxiliares tendremos una variable de asignación job/máquina ( $Z_{jm}$ ) con valor 1 en el caso de que se esté utilizando el job en la máquina  $m$ . Esto está involucrado en la generación paralela de jobs previos a la selección del mejor candidato a ser utilizado e indica cuál es el elegido. La siguiente variable ( $Hmax_j$ ) hace referencia a la altura máxima de las piezas asignadas a un job. El coste de producción de un job en una máquina queda definido por la última variable ( $JPC_{jm}$ ).

Tabla 5. Variables auxiliares del modelo

Variable	Definición	Tipo
$Z_{jm}$	= 1 si alguna pieza es asignada al trabajo $j$ de la máquina $m$ (si $\sum_i X_{ij} \geq 1$ )	Binaria
$Hmax_j$	Máxima altura de cualquier pieza $i$ del trabajo $j$	Float
$JPC_{jm}$	Coste de producción de la estructura $j$ en la máquina $m$	Float

### 3.3.5 Función objetivo

La función objetivo utilizada en el estudio hace referencia a los costes de producción totales de la cadena de fabricación. Se busca minimizar el coste total de la cadena productiva de FA por unidad de volumen de manera análoga al utilizado por Li et al., (2017). La variable JPC (*job production cost*) queda definida en el siguiente apartado de restricciones.

$$\text{Minimizar } \left\{ \sum_{j=1}^{n_b} \sum_{m=1}^{n_m} JPC_{jm} \right\} \quad (1)$$

### 3.3.6 Restricciones

Un total de 7 ecuaciones son a las que queda sujeto el modelo tratado, enumeradas y descritas en este apartado. A continuación, se muestran las ecuaciones:

$$JPC_{jm} = SET_m \cdot HC \cdot Z_{jm} + (VT_m \cdot TC_m + MC) \cdot \sum_{i=1}^{n_g} \sum_{k=1}^{n_o(i)} V_i \cdot X_{ijmk} + HT_m \cdot TC_m \cdot Hmax_{jm} \quad (2)$$

$$\forall j = 1, \dots, n_b; \quad \forall m = 1, \dots, n_m$$

$$\sum_{j=1}^{n_b} \sum_{m=1}^{n_m} \sum_{k=1}^{n_o(i)} X_{ijmk} = 1 \quad \forall i = 1, \dots, n_g \quad (3)$$

$$\sum_{i=1}^{n_g} \sum_{k=1}^{n_o(i)} A_{ik} \cdot X_{ijmk} \leq MA_m \quad \forall j = 1, \dots, n_b; \quad \forall m = 1, \dots, n_m \quad (4)$$

$$\sum_{i=1}^{n_g} \sum_{k=1}^{n_o(i)} X_{ijmk} \leq n_g \cdot Z_{jm} \quad \forall j = 1, \dots, n_b; \quad \forall m = 1, \dots, n_m \quad (5)$$

$$\sum_{k=1}^{n_o(i)} H_{ik} \cdot X_{ijmk} \leq Hmax_{jm} \quad \forall i = 1, \dots, n_g; \quad \forall j = 1, \dots, n_b; \quad \forall m = 1, \dots, n_m \quad (6)$$

$$Hmax_{jm} \leq MH_m \quad \forall j = 1, \dots, n_b; \quad \forall m = 1, \dots, n_m \quad (7)$$

$$\sum_{j=1}^{n_b} \sum_{m=1}^{n_m} Z_{jm} \leq n_b \quad (8)$$

El primer conjunto de restricciones (2) supone el cálculo de las variables involucradas en la función objetivo, relacionada con los costes totales de producción. Estas ecuaciones están divididas en 3 términos de coste que describen los costes de puesta a punto de las máquinas cada vez que se va a fabricar un job; los costes de fabricación en cada máquina caracterizados por el volumen de material a fabricar; y los costes de fabricación por máquina relacionados con la altura máxima de las piezas a fabricar en el job. El siguiente conjunto de restricciones (3) restringe la asignación de piezas a jobs limitando y asegurando que cada pieza se utiliza una única vez. En cada uno de los jobs asignados a una máquina se detalla el límite de área disponible en cada job con la ecuación (4). En las restricciones (5) se indican que si para un job  $j$  de una máquina  $m$ ,  $Z_{jm} = 0$ , entonces  $\sum_{ik} x_{ijmk} = 0$ , es decir, si una estructura no se usa en una máquina, no se le asignan piezas. La componente de altura es una de las características primordiales en la asignación de piezas y está definida en los conjuntos de ecuaciones (6) y (7) obteniendo, respectivamente, el valor de las alturas máximas de cada estructura en cada máquina e impidiendo que éstas sean mayores que la altura máxima permitida en la máquina. La última ecuación (8) limita el número de jobs permitidos en el proceso productivo.

# 4 MÉTODO DE RESOLUCIÓN

---

El desarrollo de un algoritmo memético para la optimización del problema, basado en el Sistema de Colonia de Hormigas (ACS, según sus siglas en inglés), junto a una búsqueda local posterior, ha sido considerado para llevar a cabo la experimentación junto a una búsqueda local posterior para solucionar el problema de planificación de la producción en FA. Se trata de una técnica probabilística para resolver problemas computacionalmente complejos que pueden ser reducidos a la búsqueda de buenos caminos. El problema que tratamos es de tipo NP-Hard, lo que hace que su complejidad de resolución computacional justifique el uso de heurísticas y metaheurísticas debido a que no se puede solucionar el problema en un tiempo polinómico. Este algoritmo de resolución es conocido con el nombre de Mem-ACS y destaca por la capacidad de obtener soluciones de mayor calidad gracias a la combinación de una heurística constructiva probabilística y adaptativa mediante hormigas artificiales, junto con algoritmos de búsqueda local (Molina et al., 2020a). Concretamente el Mem-ACS utiliza un algoritmo de Descenso de Vecindad Variable (VND) para la explotación de la búsqueda local. Previo a la ejecución del algoritmo memético se han encontrado soluciones iniciales mediante una heurística de construcción semiparalela con el objetivo de seleccionar la combinación de parámetros óptima para el problema.

Esta sección presenta la metodología de resolución propuesta para el problema propuesto. En la sección 4.2 se describe la heurística de construcción semiparalela, y en la sección 4.3 se presentan los principales componentes del algoritmo memético desarrollado.

## 4.1 Introducción a los métodos de resolución

Los procedimientos de resolución utilizados en problemas relacionados de la literatura pueden ser clasificados en métodos exactos y métodos aproximados. Todos ellos forman parte de métodos de optimización para acelerar la búsqueda de soluciones.

Los métodos exactos aportan la solución óptima del problema mediante la evaluación de todas las posibles combinaciones para resolver el problema, ya sea de manera explícita o implícita.

Debido a la dificultad del problema analizado, es inviable obtener una solución óptima mediante método exacto en un tiempo razonable. Este es el motivo por el que es necesario el uso de los métodos aproximados para generar soluciones. No obtendremos la certeza de alcanzar una solución óptima, pero sí soluciones buenas en tiempo de computación aceptables. Los métodos de resolución aproximados que podemos encontrar son:

- Heurísticas: Las heurísticas son procedimientos simples que buscan proporcionar soluciones eficaces,

pero no necesariamente óptimas a problemas complejos de una manera fácil y rápida. Suelen emplearse en sustitución de métodos exactos, como paso previo a la ejecución de otro algoritmo o en el caso de datos con alta variabilidad.

- Metaheurísticas: Las metaheurísticas se utilizan como alternativa a las heurísticas en problemas más complejos. Encuentran soluciones eficaces de una manera eficiente y están compuestas de fases de intensificación y diversificación. Siendo estas dos características muy importantes en la definición de los algoritmos.
  - La intensificación está relacionada con la explotación del espacio de soluciones en la búsqueda local, para conseguir soluciones de mayor calidad como proceso posterior a la diversificación.
  - La diversificación se enfoca en la exploración mediante búsqueda global, abarcando un espacio mayor de soluciones. Teniendo en consideración el proceso de búsqueda anterior se evalúan regiones que contrastan fuertemente con las regiones exploradas hasta el momento, generando una variación del entorno de soluciones posibles.

Es fundamental buscar el equilibrio correcto entre ambas características en el proceso de diseño de una metaheurística efectiva, ya que una falta de intensificación provoca que la exploración sea demasiado larga, y una falta de diversificación produce una rápida convergencia a mínimos locales (Molina et al., 2020b).

Los métodos metaheurísticos pueden presentar los siguientes enfoques:

- Enfoque constructivo: basado en generar la solución paso a paso disminuyendo así el tamaño del problema en cada decisión al descartar posibles soluciones.
- Enfoque iterativo: se realiza una búsqueda local en cada iteración, estudiando las soluciones del entorno de la solución. Este entorno se denomina vecindario, y se corresponde con el conjunto de soluciones formadas a partir de una modificación de una solución. Uno de los métodos más conocidos de este tipo de enfoque es la Búsqueda Tabú (TS).
- Enfoque evolutivo: inspirado en mecanismos biológicos como son la selección y evolución natural. Genera nuevas soluciones seleccionando, cruzando, o mutando las soluciones existentes en un conjunto denominado población inicial, obteniendo así soluciones de mayor calidad “evolucionadas” a partir de las existentes.

El algoritmo utilizado es el resultado de la combinación de varios métodos heurísticos y metaheurísticos con el objetivo de obtener mejor resultados y de una manera más eficiente. Estos algoritmos se conocen bajo el nombre de Algoritmos Meméticos (MA) y son un tipo de metaheurística que combina la búsqueda global basada en población junto a mecanismos de búsqueda local. La componente de búsqueda global se encarga de realizar la exploración del espacio de búsqueda y las técnicas de búsqueda local se utilizan para la exploración posterior (explotación) de las regiones que se consideran prometedoras, lo que resulta útil para obtener soluciones más precisas (Molina et al., 2020b).

El estudio realizado utiliza un algoritmo memético basado en ACS como metaheurística poblacional y VND como búsqueda local. Este algoritmo precisa de un valor de partida que nuestro en caso se obtiene a partir de la elaboración de una heurística constructiva semiparalela.

## 4.2 Heurística constructiva semiparalela

Esta heurística es la utilizada en el paso inicial a la obtención de la solución. Es la encargada de obtener el valor de partida del algoritmo y sobre el que se realizará la optimización mediante la metaheurística.

El término semiparalelo hace referencia a la generación simultánea de posibles estructuras, una por máquina, sobre las que colocar las piezas a ser fabricadas, mientras que la programación de jobs para ser introducidas en las máquinas se genera de manera secuencial. En cada iteración del mecanismo de construcción se consideran todas las orientaciones disponibles, y todas las piezas sin asignar son candidatas para insertar en todas las máquinas. Siguiendo la metodología de Paraskevopoulos et al., (2008), las estructuras se inicializan con un “seed”, que hace referencia al criterio de introducción de piezas basándose en la pieza de mayor volumen disponible y que cumple con las especificaciones de la máquina. Las próximas piezas se introducirán en

función de los valores de la función “greedy”. Esta función está formada por cuatro métricas diferentes que miden la relativa importancia de introducir una pieza en una determinada orientación en una estructura. Las características analizadas en la inserción son: (1) la altura de la pieza, (2) la superficie ocupada por la pieza en la estructura, (3) el volumen ocupado por la pieza y (4) el incremento de coste generado en la estructura por la inserción de la pieza.

Inicialmente, para cada máquina, se selecciona una pieza en su mejor orientación del conjunto de piezas no asignadas utilizando la función “greedy”. A continuación, la pieza seleccionada se asigna temporalmente dentro del job correspondiente en la máquina. Este procedimiento se repite para cada máquina explícitamente, considerando todas las piezas no asignadas, incluidas las asignadas temporalmente a otras máquinas, hasta que todos los jobs se llenan de piezas. Al final de este procedimiento, se selecciona un job junto a su conjunto de piezas asignadas y se añade a la solución parcialmente construida. La selección del job se basa en un criterio que mide el coste de fabricación del job en la máquina correspondiente, seleccionando el de menor coste por unidad de volumen. Por último, las piezas asignadas a los jobs seleccionados se eliminan del conjunto de piezas no asignadas y se repite el procedimiento global con la selección de un nuevo job, hasta que todas las piezas queden asignadas a un job. Hay que tener en cuenta que durante la construcción de los jobs de las máquinas de manera semiparalela, se permite la duplicación de piezas; sin embargo, el hecho de que sólo se incluya un job de los generados de forma paralela en las distintas máquinas a la solución parcialmente construida en cada iteración, garantiza que cada pieza sea fabricada una única vez.

La métrica de selección del mejor job durante la construcción semiparalela se basa en el menor coste por unidad de volumen. Esta métrica nos facilita la comparación en función de la eficiencia de la fabricación y le daremos el nombre de ACUM por sus siglas en inglés (*Average Cost of Unit Manufactured*; Coste Medio por Unidad Fabricada). El indicador se expresa como la fracción entre los costes de fabricación del job (caracterizados por los valores de coste por volumen, operación y setup) y el volumen de material a fabricar total en el job. De esta manera, a menor valor de ACUM, mayor será la eficiencia del proceso. Tal y como se mencionaba anteriormente, el job seleccionado de la máquina correspondiente se añade a la solución parcialmente generada. Siguiendo este modo de selección, en cada iteración se selecciona el job producido en una máquina  $m$  que minimiza el coste por volumen. La métrica ACUM queda definida a continuación:

$$ACUM_m = \frac{HC * SET_m + (TC_m * VT_m + MC) * \sum V_i + HT * TC_m * H_{MAX}}{\sum V_i} \quad (9)$$

Tras haber definido el criterio de selección de jobs, se debe dar foco al criterio para la selección y posterior inserción de piezas para generar los jobs.

La metodología para insertar piezas con una orientación concreta en jobs se ha desarrollado mediante el cálculo de la función de “greedy” que tiene en consideración valores referentes a la superficie, la altura, el volumen y el coste de añadir cierta pieza en una orientación concreta.

Respecto a la superficie y el volumen, se pretende conseguir una mayor utilización de estas magnitudes en el job finalmente construido. Por lo tanto, las orientaciones que ocupen mayor superficie y las piezas con mayor volumen tendrán prioridad en el proceso de inserción consiguiendo maximizar el espacio utilizado. Respecto a la componente de altura, el objetivo es conseguir que la altura de la pieza más alta sea lo menor posible, reduciendo de esta manera los costes de operación por altura. Para ello se analiza la variación de altura en el job al añadir una pieza nueva, en una orientación concreta. De manera análoga, para los costes de material se tiene en cuenta la variación de coste al insertar una pieza nueva.

Siendo  $i$  la pieza candidata a insertar en la orientación  $k$  dentro del job  $j$ , estando este job parcialmente generado, la función “greedy” evalúa la importancia relativa de insertar la pieza  $i$  con orientación  $k$  en el job  $j$ . La mejor inserción vendrá dada por el valor más pequeño de la función “greedy”. Esta función combina los efectos de varias submétricas mediante una relación lineal ponderada de manera adimensional definida de la siguiente manera:

$$\Phi_{ik}^j = \alpha_1 * C_{jik}^1 + \alpha_2 * C_{jik}^2 + \alpha_3 * C_{jik}^3 + \alpha_4 * C_{jik}^4 \quad (10)$$

Los valores de  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  y  $\alpha_4$  son pesos no negativos cuya suma es la unidad.

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \quad (11)$$

La primera componente ( $C_{jik}^1$ ) expresada en la ecuación (12) hace referencia a la componente de superficie a cubrir por la pieza  $i$  a insertar en la orientación  $k$ .  $S_{max}$  corresponde a la superficie máxima del job  $j$ ,  $S_j$  indica la superficie ya cubierta por las piezas asignadas con anterioridad al job, y  $S_{ik}$  denota la superficie de la nueva pieza a ser insertada. Esta componente depende de la orientación considerada para la pieza y ofrece ventaja a la pieza que vaya a ocupar una mayor superficie en la bandeja del job.

$$C_{jik}^1 = \frac{S_{max} - S_j - S_{ik}}{S_{max}} \quad (12)$$

La segunda componente ( $C_{jik}^2$ ) evalúa la componente de volumen a utilizar por la pieza  $i$ . Este componente no presenta variación respecto a las orientaciones ya que es una constante de la pieza. De manera análoga a la componente primera ( $C_{jik}^1$ )  $V_{max}$  corresponde al volumen máximo a ser cubierta en el job  $j$ ,  $V_j$  indica el volumen ya cubierto por piezas anteriormente asignadas al job, y  $V_i$  denota el volumen de la nueva pieza a ser insertada.

$$C_{jik}^2 = \frac{V_{max} - V_j - V_i}{V_{max}} \quad (13)$$

La tercera componente ( $C_{jik}^3$ ) indica la variación de altura que sufre el espacio de fabricación de la máquina.  $H_{max}$  corresponde a la altura máxima que permite la máquina,  $H_j$  indica la altura máxima de las piezas anteriormente asignadas al job, y  $H_{ik}$  denota la altura de la nueva pieza a ser insertada en la orientación  $k$ .

$$C_{jik}^3 = \frac{H_{ik} - H_j}{H_{max}} \quad (14)$$

Esta diferencia de alturas puede llevar a dos resultados:

1. La diferencia de alturas sea menor o igual que cero. Este caso describe la situación para la que la nueva pieza tiene una altura menor a la de la pieza más alta. Al valor de la componente  $C_{jik}^3$  se le otorgará un valor de cero ya que no influye en la decisión de inserción. Siempre se busca insertar piezas en aquellas posiciones que no aumenten, o aumenten en la menor medida  $H_j$ .
2. La diferencia de alturas sea mayor que cero. En esta situación la altura de la pieza a insertar en una orientación determinada  $k$  es mayor a la altura de la pieza más alta, lo que genera un valor positivo de la componente  $C_{jik}^3$ . Si finalmente esa pieza es asignada al job, actualizaremos el valor de  $H_j$  con la



nueva altura  $H_{ik}$ .

$$H_j^* = H_{ik} \quad (15)$$

La cuarta componente ( $C_{jik}^4$ ) está relacionada con la variación de coste de fabricación de la máquina.  $C_{max}$  corresponde al coste máximo que permite la máquina en referencia a las componentes de costes relacionadas (costes por volumen de material, operación y setup en los casos extremos de saturación de la máquina). De manera análoga a la componente ( $C_{jik}^3$ ),  $C_j$  indica el coste actual de fabricación del job y  $C_{ik}$  denota el coste total del job tras la posible inserción de la pieza  $i$  en la orientación  $k$ .

$$C_{jik}^4 = \frac{C_{ik} - C_j}{C_{max}} \quad (16)$$

El coste de fabricación de una nueva pieza siempre va a acarrear un coste, por lo tanto, siempre que se introduzca una nueva pieza  $i$  en una orientación  $k$  se actualizará el coste del job ( $C_j$ ). La siguiente fórmula expresa la actualización de  $C_j$ , donde  $C_j^*$  hace referencia al nuevo valor.

$$C_j^* = C_{ik} \quad (17)$$

El proceso de introducción de piezas en jobs finalizará cuando se hayan insertado todas las piezas del problema en un job de una máquina.

Concretamente la función de “greedy” premiará a las orientaciones que presenten una superficie y volumen mayores, y una altura y costes de fabricación menores, y realizará una ponderación de los cuatro componentes en función de los valores de alfa introducidos. El estudio llevado a cabo realiza un ajuste de estos parámetros en el apartado de experimentación con el objetivo de encontrar el rango de alfas más adecuado para encontrar una solución de partida sobre el que ejecutar la metaheurística aplicada (Mem-ACS).

A continuación, se aporta el pseudocódigo de la heurística constructiva:

**Heurística de construcción semi-paralela ( $\alpha_1$  rango,  $\alpha_2$  rango,  $\alpha_3$  rango,  $\alpha_4$  rango)**

```

1 MS ← Mejor_Solucion();
2 For  $\alpha_1 \in \alpha_1\_rango$  do:
3   For  $\alpha_2 \in \alpha_2\_rango$  do:
4     For  $\alpha_3 \in \alpha_3\_rango$  do:
5        $\alpha_4 = 1 - \alpha_3 - \alpha_2 - \alpha_1$ 
6       If ( $\alpha_4 \geq 0$ ):
7         S ← SolucionInicial(), LPi ← ListaPiezas();
8         While (LPi ≠ 0) do:
9           For todas las máquinas m de Mm do:
10            jm ← InicializaJob(m), (semilla, orient) ← BuscaPiezaSemilla()
11            If (semilla ≠ 0):
12              jm ← InsertaPiezaSemilla(semilla, orient), done ← True
13              While (done = True) do:
14                done ← False
15                For todas las piezas i de LPi - {semilla} do:
16                  For todas las orientaciones k de i do:

```

```

17          $\phi_{i,j,k,m} \leftarrow \text{FuncionGreedy}(i, j, k, m, \alpha_1, \alpha_2, \alpha_3, \alpha_4),$ 
18         If ( $\phi_{i,j,k,m} > 0$ ):
19              $\Phi \leftarrow \text{AlmacenarMejor}(\phi_{i,j,k,m}), \text{done} \leftarrow \text{True}$ 
20         EndIf
21     EndFor
22 EndFor
23 If (done):
24      $jm \leftarrow \text{Insert}(i, k, \Phi)$ 
25 EndIf
26 EndWhile
27 EndIf
28 EndFor
29 For todos los jobs  $jm$  do:
30      $ACUMm \leftarrow \text{AverageCostUnitManufactured}(jm), x \leftarrow \text{AlmacenarMejor}(ACUMm)$ 
31 EndFor
32      $S \leftarrow \text{InsertarJob}(jx), LPi \leftarrow \text{EliminarPiezas}(jx);$ 
33 EndWhile
34 If  $f(S)$  improves  $f(MS)$  then
35      $MS \leftarrow S;$ 
36 EndIf
37 EndIf
38 EndFor
39 EndFor
40 EndFor

```

La siguiente secuencia de imágenes describe el proceso de construcción de job de forma paralela de una manera gráfica. Se ha tomado un ejemplo simple utilizando 8 piezas a ser fabricadas en 3 máquinas FA distintas. En el ejemplo mostrado no se han considerado las orientaciones de las piezas para simplificar el proceso y facilitar la representación de manera esquemática. Se muestran en azul los jobs que están siendo generados de forma paralela, en verde el job seleccionado y en naranja los jobs descartados. Horizontalmente se identifica el objeto esquemático y verticalmente se ilustra la inserción de piezas en el job de cada máquina. La diferente forma que presentan los jobs representa las distintas características que tiene cada máquina suponiendo restricciones a la hora de evaluar la inserción de piezas.

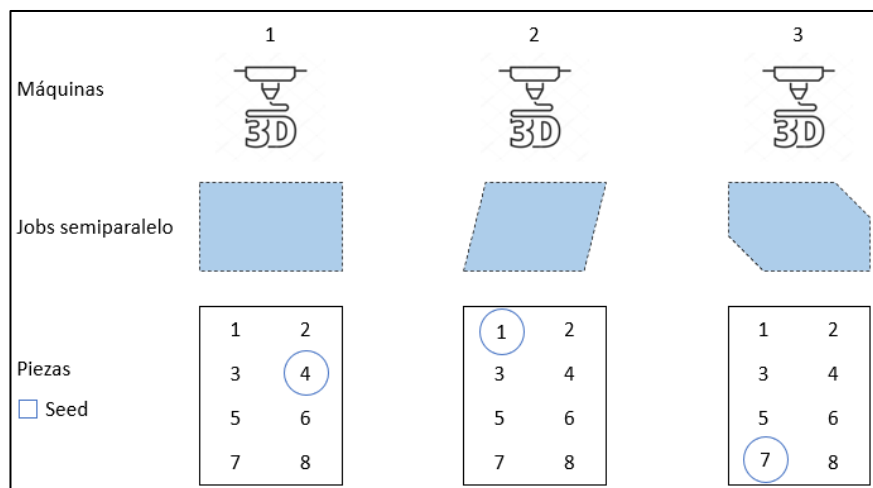


Ilustración 8. Formación de jobs 1

Esta primera imagen muestra la situación inicial de generación de jobs. Observamos que todas las piezas están disponibles para todas las máquinas, ya que es el comienzo del proceso. Cada máquina tiene una forma de job diferente que se ajusta a las características de la máquina. Las piezas señaladas de azul hacen referencia a las piezas "seed" a insertar inicialmente en cada job (pieza de mayor volumen admitida en el job). Esto puede variar en función de las características de las piezas y su adecuación a la máquina.

A continuación, se insertarán el resto de las piezas según los valores de la función “greedy” hasta completar el espacio de producción del job. En la siguiente imagen podemos ver las piezas insertadas en cada jobs y las piezas que han quedado sin ser utilizadas en la generación del primer job.

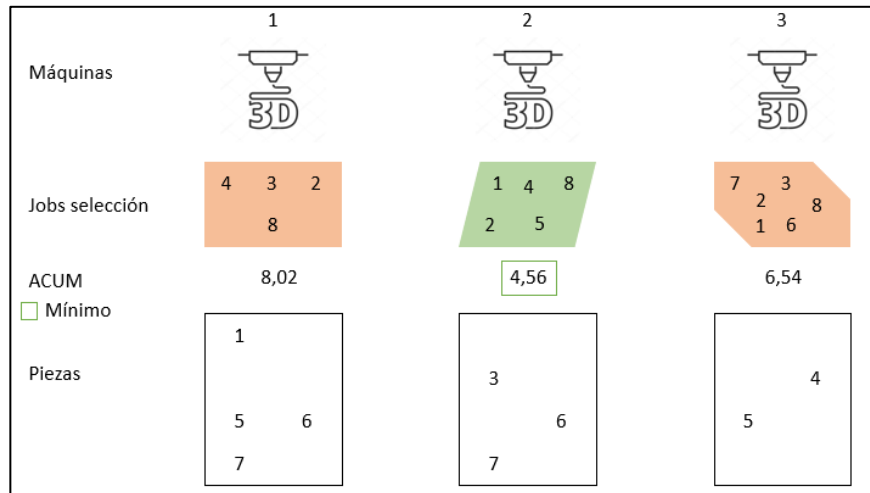


Ilustración 9. Formación de jobs 2

Tras haber completado los jobs, se evalúa el valor de la métrica ACUM para cada uno de ellos y se selecciona el de menor coste por unidad de volumen, marcado en verde en la imagen. Será el job señalado en verde el elegido a ser fabricado. El resto de jobs marcados en naranja son descartados. Se continúa con la generación de jobs a partir de las piezas que han quedado sin utilizar, siendo en este caso las piezas 3, 6 y 7. Las ya insertadas se eliminan del grupo de posibles piezas.

Se vuelven a generar los 3 jobs candidatos de forma paralela e inicia el proceso de inserción nuevamente comenzando por la inserción de la pieza “seed”.

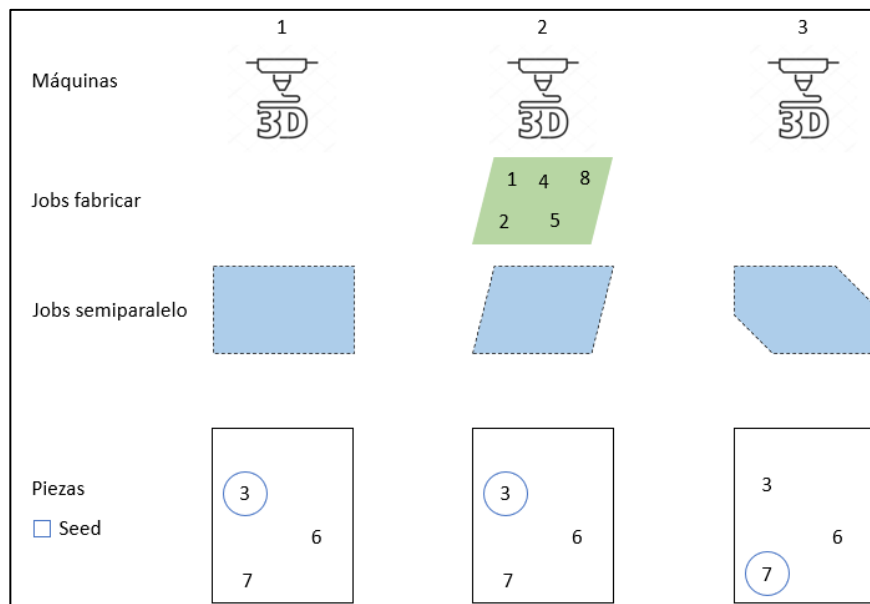


Ilustración 10. Formación de jobs 3

Siguiendo la inserción de la pieza “seed” se repite el proceso de inserción del resto de piezas que quedan por ser fabricadas. Se completan de nuevo los jobs y se evalúa el coste por unidad de volumen para seleccionar el job.

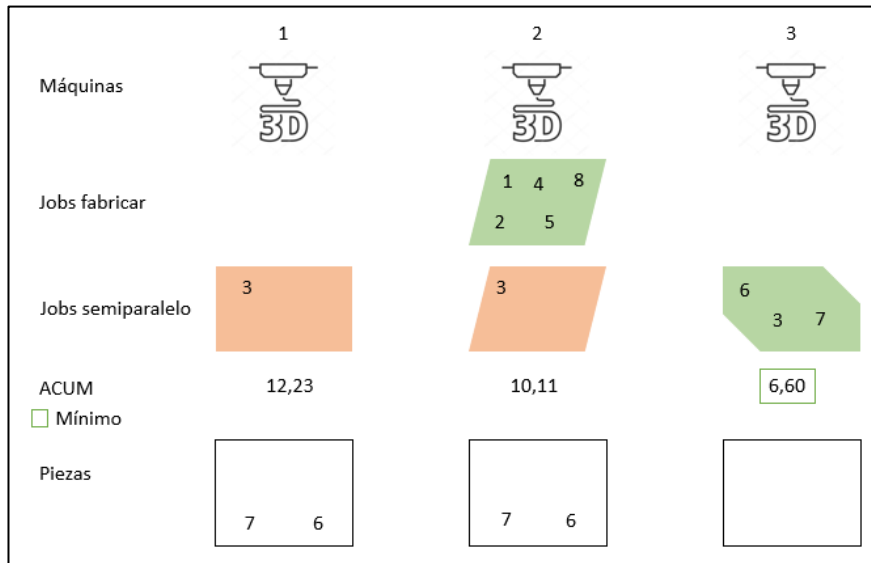


Ilustración 11. Formación de jobs 4

Se puede apreciar que el job referido a la máquina 3 ha sido el elegido completando así el proceso de generación de jobs.

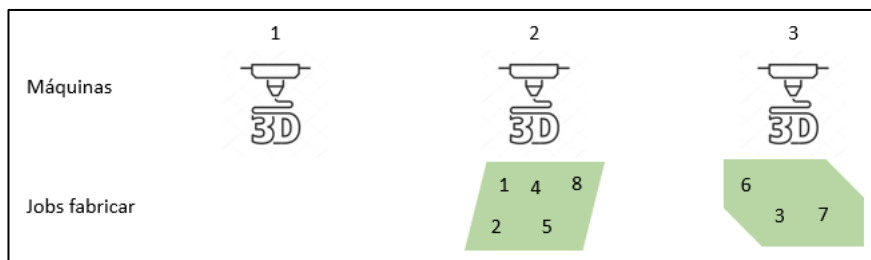


Ilustración 12. Formación de jobs 5

Finalmente, el ejemplo que utilizaba 8 posibles piezas a ser fabricadas en 3 máquinas queda completado con dos jobs, asignados a las máquinas dos y tres respectivamente.

### 4.3 Metaheurística (Mem-ACS)

Los algoritmos meméticos son aquellos que combinan varios procesos heurísticos y metaheurísticos. Suelen estar formados por dos componentes principales, una de búsqueda global basada en población y otra de búsqueda local. En el estudio llevado a cabo se ha adoptado un algoritmo memético que se nutre de un algoritmo de colonia de hormigas (ACS), y de una heurística de búsqueda descendente sobre vecindad variable (VND).

Este algoritmo memético fue presentado para la optimización del problema del rutado de vehículos (VRP) con ventanas temporales y vehículos heterogéneos (Molina et al., 2020a) y ha supuesto las bases para la ejecución del problema de FA. Las similitudes de generación de rutas por parte de las hormigas en el problema VRP puede asemejarse considerablemente al problema de planificación de la producción de piezas en jobs en FA, aunque con diferentes restricciones. Si para el VRP tenemos varios vehículos que recorren rutas pasando por los puntos de demanda necesarios, el problema de FA tendrá una serie de máquinas que albergan agrupaciones de piezas definidas como jobs. En FA los jobs serían equivalentes a las rutas en el VRP y las piezas harían referencia a los clientes. Además, en FA la posición u orden que ocupa una pieza dentro de un job no influye en el valor del coste, contrariamente a lo que ocurre con los clientes en las rutas generadas en el VRP. El cambio en la forma de visualizar el problema sería que en FA las piezas están asignadas a los jobs, y éstos pertenecen a las máquinas. De esta manera el ACS se aproxima al problema a tratar de FA generando en lugar de rutas, jobs de manera progresiva formados por piezas en orientaciones concretas para las máquinas.

El siguiente pseudocódigo describe el algoritmo memético desarrollado:

**Algoritmo memético del sistema de colonia de hormigas**

```

1 Initialization of parameters ( $\alpha1_{\text{rango}}, \alpha2_{\text{rango}}, \alpha3_{\text{rango}}, \alpha4_{\text{rango}}, \text{iter\_max}, M, \rho, \alpha, \beta, q0$ );
2  $S \leftarrow$  Semi-parallel Insertion Heuristic ( $\alpha1_{\text{rango}}, \alpha2_{\text{rango}}, \alpha3_{\text{rango}}, \alpha4_{\text{rango}}$ ),
3  $\tau_{ikpk'm} \leftarrow$  Pheromone_matrix();  $\tau_0 \leftarrow$  Calculate_pheromone_value (S);  $\Delta\tau_{bs} \leftarrow \tau_0$ ;
4  $\tau(N,N) \leftarrow$  Initialize_pheromone_matrix ( $\tau_0$ );
5 iterations=1;
6 While (iterations  $\leq$  iter_max) AND (CPU time consumed  $\leq \gamma$ ) do:
7   ( $\alpha1, \alpha2, \alpha3, \alpha4$ )  $\leftarrow$  Modifica_parametros(iterations)
8   ant=1;
9   While (ant  $\leq$  M) AND (CPU time consumed  $\leq \gamma$ ) do:
10     $S' \leftarrow$  Ant_Route_Construction ( $\alpha, \beta, q0, \alpha1, \alpha2, \alpha3, \alpha4$ );
11     $\tau_{ikpk'm} \leftarrow$  Local_Update_pheromone_matrix ( $S', \rho, \tau_0$ );
12     $S'' \leftarrow$  VND ( $S'$ );
13    If  $f(S'')$  improves  $f(S)$  then
14       $S \leftarrow S''$ ;  $\Delta\tau_{bs} \leftarrow$  Calculate_pheromone_value ( $S''$ );
15      ant=ant+1;
16    EndWhile
17     $\tau_{ikpk'm} \leftarrow$  Global_Update_pheromone_matrix (S,  $\rho, \Delta\tau_{bs}$ );
18    iterations=iterations+1;
19  EndWhile

```

**4.3.1 Búsqueda global - Sistema de colonia de hormigas (ACS)**

ACS se inspira directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria y fue presentado por primera vez por Dorigo & Gambardella, (1997). Se basan en una colonia de hormigas artificiales, esto es, agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromonas artificiales para identificar el camino óptimo a seguir. ACS define un número máximo de iteraciones ( $N$ ) y una población de hormigas ( $M$ ) que participa en las dos fases principales del algoritmo; la construcción de la solución de la hormiga y la actualización de la feromona.

**4.3.1.1 Construcción de jobs**

En el problema de FA una hormiga individual simula un conjunto de jobs, cada uno de ellos asociado a una máquina en particular, que se construirán secuencialmente. Se necesita únicamente una lista que albergue el conjunto de piezas no asignadas para construcción de job. La construcción de soluciones se realiza utilizando la heurística semiparalela explicada en el apartado 4.2 de este TFM. De esta forma en cada iteración, primeramente, se construirá un job para cada máquina FA, considerando que todas las piezas sin asignar son candidatas para insertar en todas las máquinas. Posteriormente, se escogerá como parte de la solución final el job con el valor más pequeño de ACUM.

En la construcción de jobs, la selección de la pieza y orientación a insertar en cada iteración dependerá del valor de la probabilidad  $P_{ijk}$  mostrada en la ecuación (18). Ésta representa la probabilidad de elegir la pieza  $i$  con orientación  $k$  para un job  $j$ . El proceso de elección de pieza y orientación sigue el siguiente método proporcional pseudoaleatorio. Se escoge un número  $q$  de manera aleatoria distribuido uniformemente entre  $[0,1]$ . En el caso de que  $q$  sea menor que  $q_0$  se utiliza la ecuación (18) para seleccionar la siguiente pieza y orientación a insertar (será aquella que presente el valor más alto de probabilidad). En caso contrario, la pieza y orientación a insertar se elige mediante un criterio de selección de “fitness” proporcional conocido como “selección de rueda de ruleta” (Lipowski & Lipowska, 2012), en función de la distribución de probabilidad dada por la ecuación (18). Por lo tanto, con probabilidad  $q_0$ , se realiza la mejor inserción descrita, mientras que con probabilidad  $(1 - q_0)$  se exploran diferentes alternativas.

La construcción de un job comienza con la elección de la pieza de mayor volumen (“seed”) que produzca una inserción factible en alguna orientación. La orientación seleccionada vendrá dada a través del método anteriormente descrito. A continuación, las piezas se irán insertando iterativamente en los jobs, hasta que ninguna pieza pueda insertarse en alguna orientación debido a las restricciones de superficie o altura máxima. A continuación, el job perteneciente a una máquina con menor ACUM se añade a la solución parcialmente construida y las piezas utilizadas en dicho job se eliminan de la lista de piezas no asignadas. De esta forma, el procedimiento general se repite hasta que todas las piezas son asignadas a un job de una máquina.

$$P_{ijk} = \begin{cases} \frac{(\tau_{ijk})^\alpha * (\eta_{ijk})^\beta}{\sum_B (\tau_{ijk})^\alpha * (\eta_{ijk})^\beta} & \text{si } (i, k) \in V(j) \\ 0 & \text{en caso contrario} \end{cases} \quad (18)$$

$P_{ijk}$  toma en consideración dos valores distintos denotados por  $(\tau_{ijk})$  que representa el nivel de feromona y  $(\eta_{ijk})$  que representa la atractividad de la inserción.  $V(j)$  representa el conjunto de inserciones factibles (pieza orientación) en el job teniendo en cuenta las características de la máquina.

Los parámetros alfa ( $\alpha$ ) y beta ( $\beta$ ) son parámetros de control de la influencia de la feromona y la atractividad respectivamente. Estos valores de feromona y atractividad se definen específicamente para el tipo de problema planteado siendo la atractividad expresada en la ecuación (19) y la feromona en el siguiente apartado.

$$\eta_{ijk} = \frac{1}{\Phi_{ik}^j} \quad (19)$$

Para la ecuación de la atractividad (19) el valor de  $\Phi$  hace referencia a la ecuación de greedy (10) mostrada en el apartado anterior. Esta función mide el coste de insertar una nueva pieza en el job.

Los parámetros alfa pertenecientes a la función de “greedy” irán variando a lo largo de las iteraciones del algoritmo. Esto permitirá variar la atractividad a medida que avanza el algoritmo generando distinta combinación de parámetros y contribuyendo a la diversificando las soluciones.

A continuación, se aporta el pseudocódigo de la heurística de generación de ruta de hormigas:

#### **Proceso de construcción de ruta de hormiga ( $\alpha, \beta, q0, \alpha1, \alpha2, \alpha3, \alpha4$ )**

```

1  S ← SolucionHormiga(), LPi ← ListaPiezas();
2  While (LPi ≠ 0) do:
3  For todas las máquinas m de Mm do:
4    jm ← InicializaJob(m), (semilla, orient) ← BuscaPiezaSemilla()
5    If (semilla ≠ 0):
6      jm ← InsertaPiezaSemilla(semilla, orient), done ← True
7      While (done = True) do:
8        done ← False
9        For todas las piezas i de LPi - {semilla} do:
10         For todas las orientaciones k de i do:
11           ηi,j,k,m ← FuncionGreedy(i, j, k, m, α1, α2, α3, α4),
12           ρi,j,k,m ← Calculate_Probabilities(ηi,j,k,m, τikpk'm),
13         EndFor
14       EndFor
15       If (ρi,j,k,m ≠ 0):
16         q ← ChooseRandomNumber();
17         If (q ≤ q0):
18           (pieza, orient) ← Maximum_Value(ρi,j,k,m);

```

```

19         Else:
20             (pieza,orient)←Roulette_Wheel_Selection ( $\rho_{i,j,k,m}$ );
21         EndIf
22          $j_m \leftarrow Insert(pieza,orient)$ ; done ← True;
23     EndIf
24 EndWhile
25 EndIf
26 EndFor
27 For todos los jobs  $j_m$  do:
28      $ACUM_m \leftarrow AverageCostUnitManufactured(j_m)$ ,  $x \leftarrow AlmacenarMejor(ACUM_m)$ 
29 EndFor
30  $S \leftarrow InsertarJob(j_x)$ ,  $LP_i \leftarrow EliminarPiezas(j_x)$ ;
31 EndWhile
    
```

#### 4.3.1.2 Definición, inicialización y actualización de feromonas

Durante el proceso de generación de la solución factible, el rastro de feromonas representa las experiencias pasadas de las hormigas que construyen las soluciones. La definición del rastro de feromonas es crucial, ya que una definición inadecuada del rastro de feromonas conducirá a soluciones de baja calidad.

El rastro de feromona  $\tau_{ijk}$  se define como la atractividad de agrupar una pieza  $i$  con orientación  $k$  en un job  $j$ . En este trabajo, esta relación es transformada a la relación que tiene la pieza a insertar en una determinada orientación con todas las piezas pertenecientes al job parcialmente construido. De esta forma,  $\tau_{ijk}$  se define como la feromona promedio calculada a partir de los valores de feromona de la pieza a insertar en una determinada orientación y de las piezas que ya están ubicadas en una orientación en el job actual. (20), donde  $\tau_{ikpk_2m}$  representa la feromona de agrupar la pieza  $i$  en orientación  $k$  con la pieza  $p$  en orientación  $k_2$  en la máquina  $m$  y  $|j|$  denota el número de piezas insertadas en el job  $j$ . Para poder abordar el caso en el que una pieza con una determinada orientación se inserta por primera vez en un job, se considera la existencia de una pieza vacía ( $p'$ ) con la que se mantendrán valores de feromona.

$$\tau_{ijk} = \frac{\sum_{(\forall p \in j+p')} \tau_{ikpk_2m}}{|j|} \quad (20)$$

Al comenzar el algoritmo se obtiene un valor inicial de feromonas tras la ejecución del algoritmo de la heurística semiparalela. El valor inicial, definido como  $\tau_0$ , se muestra en la ecuación (21), donde  $N$  representa el número de piezas a ser fabricadas, y  $L$  es el coste por unidad de volumen obtenido en la solución inicial.

$$\tau_0 = \frac{N}{L} \quad (21)$$

El método ACS utiliza dos tipos de actualizaciones de feromonas: local y global. Cada vez que una hormiga construye una solución, la actualización local se realiza modificando el nivel de feromona de la relación correspondientes entre piezas de la solución obtenida ( $S$ ). La actualización de feromonas debe realizarse en cada iteración para asegurar la calidad y diversidad de la solución encontrada (Jia et al., 2019). Esto se consigue mediante el uso del parámetro  $\rho$  para regular la reducción de feromona, mecanismo denominado evaporación. Por otro lado, una vez que todas las hormigas han calculado su solución, la actualización global sólo se realiza mediante la hormiga que ha generado la mejor solución ( $S^*$ ).

La ecuación (22) hace referencia al método de cálculo de la actualización local de feromona. Este valor se calcula cada vez que una hormiga encuentre una solución durante una misma iteración. El valor de  $\tau_0$  es el valor inicial de feromona, y  $\rho$  es un valor entre  $[0, 1]$  que controla la velocidad de evaporación de la feromona. Dicha actualización se realizará para cada pareja de piezas  $i_1, i_2$ , con orientaciones  $k_1$  y  $k_2$ , respectivamente, que pertenezcan a un mismo job de una máquina  $m$  en la solución construida.

$$\tau_{i_1 k_1 i_2 k_2 m} = (1 - \rho) * \tau_{i_1 k_1 i_2 k_2 m} + \rho * \tau_0 \quad (22)$$

Los valores de feromonas de cada solución se actualizan globalmente después de cada iteración. Para ello, se tiene en cuenta la hormiga que ha obtenido globalmente el mejor valor de solución y se utiliza para actualizar.

La regla de actualización global de la feromona se ofrece a continuación en la ecuación (23), donde  $L_{best}$  es el coste por unidad de volumen obtenido en la mejor solución encontrada hasta el momento y  $N$  el número de piezas. Análogamente al proceso de actualización local de feromonas, la actualización se llevará a cabo para cada pareja de piezas  $i_1, i_2$ , con orientaciones  $k_1$  y  $k_2$ , respectivamente, que pertenezcan a un mismo job de una máquina  $m$  en la mejor solución.

$$\tau_{i_1 k_1 i_2 k_2 m} = (1 - \rho) * \tau_{i_1 k_1 i_2 k_2 m} + \rho * \frac{N^2}{L_{best}} \quad (23)$$

### 4.3.2 Búsqueda local - Descenso por vecindad variables (VND)

El proceso de búsqueda local es conocido como el mecanismo de intensificación del algoritmo memético. Es el utilizado para refinar la solución encontrada en la metaheurística anterior. Para nuestro problema se ha utilizado el método de Búsqueda por Entornos Variables Descendentes (*Variable Neighborhood Descent*, VND) que explora el espacio de soluciones de una manera efectiva mediante una variación sistemática de vecindades en el proceso de búsqueda local. VND es una metaheurística comúnmente usada en el proceso de búsqueda local de otras metaheurísticas (Hansen et al., 2010; Mladenović & Hansen, 1997) presentando numerosos casos de éxito en la literatura científica (p. ej. Duarte et al. 2018; Molina et al. 2020a; Molina et al. 2020b).

VND es un método metaheurístico que está inspirado en VNS, metaheurística propuesta por Mladenović & Hansen, (1997). El VNS se centra en explorar un conjunto de vecindades de una solución actual encontrada, realiza una búsqueda local sobre ella y se mueve a otra solución únicamente en el caso de que se haya producido mejora. Dado que un óptimo local para un determinado tipo de movimiento (estructura de vecindad) no lo es necesariamente para otro, VNS cambia la estructura de la vecindad durante la búsqueda para escapar de los óptimos locales. El VND aparece cuando el proceso se realiza de una forma determinista, eliminando la componente aleatoria. Por tanto, el VND comienza definiendo un conjunto de estructuras de vecindad  $N_\lambda$  ( $\lambda=1, \dots, \lambda_{max}$ ). El proceso iterativo comienza desde una solución inicial  $s$  obtenida por una hormiga en el algoritmo ACS. A continuación, un proceso de búsqueda descendente es realizado para determinar una nueva solución  $s'$  en  $N_\lambda$ . Si  $f(s')$  mejora  $f(s)$ , se actualiza la mejor nueva solución y la búsqueda retorna a la primera vecindad definida ( $N_1$ ). En otro caso, la búsqueda explora la siguiente vecindad  $N_{\lambda+1}$ . Este proceso se repite hasta que todas las vecindades son examinadas sin mejora de la solución, es decir, hasta encontrar una solución que presente un mínimo con respecto a todas las estructuras de vecindad definidas (Hansen et al., 2010).

A continuación, se aporta el pseudocódigo del VND:

#### **Algoritmo de búsqueda descendente en entornos variables (VND (S))**

```

1 Definir un conjunto de estructuras de vecindad  $N_k, k=1, 2, \dots, k_{max}$ ;
2  $k \leftarrow 1$ ;
3 While ( $k \leq k_{max}$ ) do:
4    $s' \leftarrow Busqueda\_Local(S, N_k)$ ;
5   If  $f(s')$  mejora  $f(S)$  then
6      $S \leftarrow s'; k \leftarrow 1$ ;
7   Else
8      $k \leftarrow k+1$ ;
9   EndIf

```



10 EndWhile

El VND define, para este trabajo, 5 estructuras de vecindad ( $\lambda_{max}=5$ ) para la búsqueda de nuevas soluciones:

1. Cambio de orientación de las piezas dentro de un mismo job. Este operador consiste en modificar la orientación de una pieza de un job. Por tanto, considerando el conjunto de jobs construidos en la solución analizada, se modifica iterativamente la orientación de una única pieza de un job con el propósito de encontrar mejores soluciones factibles ya que el cambio de orientación de las piezas supone un cambio en la superficie proyectada sobre el job. La Ilustración 13 representa estas posibles variaciones de una manera simplificada mediante imágenes en 2D.



Ilustración 13. Cambio de orientación de las piezas dentro de un mismo job

2. Inserción de piezas en otro job de la misma máquina. De manera iterativa se inserta una pieza de un job en otro hasta completar todas las posibles inserciones. Se analizan todas las posibles orientaciones en la inserción. Consideramos la inserción de piezas entre jobs de la misma máquina para reducir el espacio de búsqueda y por la posibilidad de incompatibilidad de parámetros de la máquina. La Ilustración 14 muestra el mecanismo de inserción entre jobs.

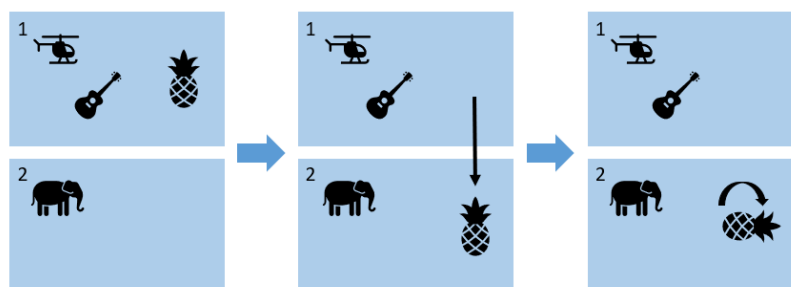


Ilustración 14. Inserción de piezas en otro job de la misma máquina

3. Intercambio de dos piezas entre jobs distintos pertenecientes a la misma máquina. Una pieza de un job se intercambia por una pieza de otro job de la misma máquina, analizando todas las posibles orientaciones de ambas piezas. Este mecanismo queda expresado mediante la Ilustración 15.

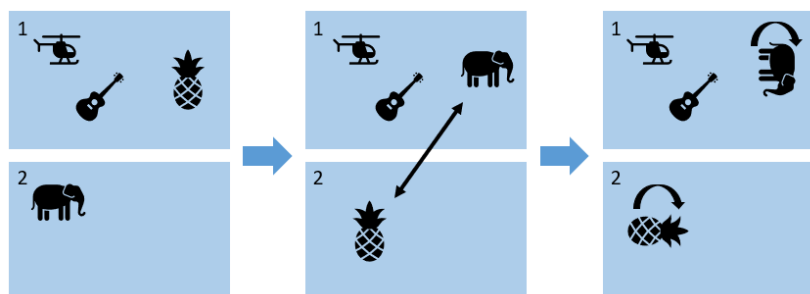


Ilustración 15. Intercambio de dos piezas entre jobs distintos pertenecientes a la misma máquina

4. Intercambio de una pieza de un job por dos piezas de otro job, ambos pertenecientes a la misma máquina y considerando las posibles orientaciones. Se muestra mediante la Ilustración 16 el intercambio de piezas.

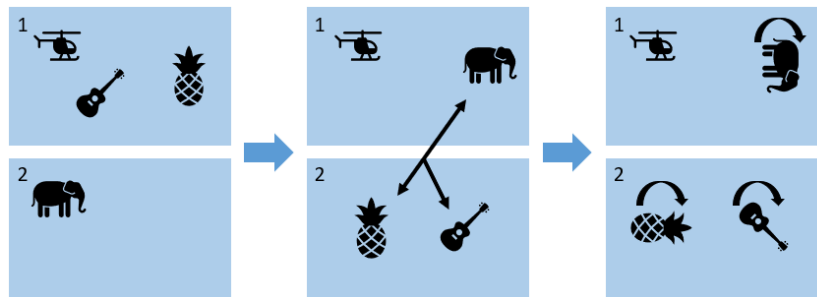


Ilustración 16. Intercambio de una pieza de un job por dos piezas de otro job

5. Intercambio de dos piezas de un job por otras dos de otro job. Intercambio 2-2 siempre entre jobs pertenecientes a la misma máquina con posibles cambios en la orientación. Se puede comprender este intercambio de piezas mediante la Ilustración 17.

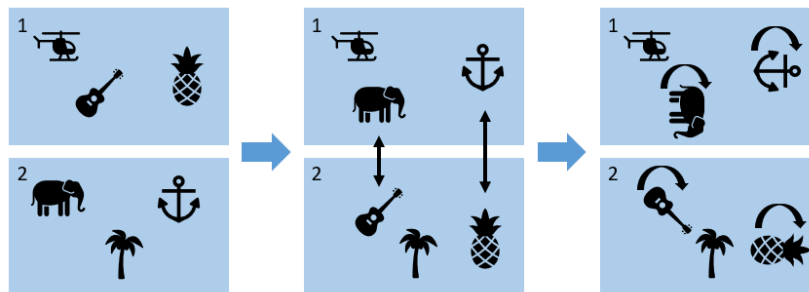


Ilustración 17. Intercambio de dos piezas de un job por otras dos de otro job

# 5 EXPERIMENTACIÓN

---

Esta sección describe los experimentos computacionales que se han llevado a cabo para demostrar la efectividad del algoritmo memético (Mem-ACS) presentado en la sección 4. Para ello, se ha comenzado generando una batería de problemas basados en los experimentos de Li et al., (2017), que resolvía los problemas para una orientación minimizando los costes de producción, introduciendo dos orientaciones alternativas más por pieza al problema.

A continuación, se ha realizado la variación de parámetros sobre un conjunto de instancias características para encontrar los mejores valores de los parámetros alfa de la heurística y de los parámetros de la metaheurística Mem-ACS. Finalmente, se ejecutaron las instancias con los parámetros seleccionados utilizando el algoritmo memético y se analizaron los resultados obtenidos. En la experimentación, se ha considerado únicamente el problema que minimiza los costes de fabricación aditiva y se han planteado tres conjuntos diferentes de problemas (1, 2, y 3 orientaciones alternativas en las piezas).

La sección 5.1 define el proceso de generación de nuevas instancias para el problema abordado, la sección 5.2 muestra el proceso de identificación de los mejores valores para los parámetros del algoritmo Mem-ACS y finalmente, en la sección 5.3 se presentan los resultados computacionales del algoritmo propuesto.

## 5.1 Generación de instancias

La generación de nuevas instancias para la experimentación del problema abordado tiene su origen en las instancias presentadas en Li et al., (2017) a las que se añaden orientaciones alternativas al conjunto de piezas. Los experimentos consisten en la ejecución de 42 problemas que van aumentando en dificultad computacional debido al aumento progresivo de piezas y máquinas. El número de piezas de los problemas varía desde 10 hasta 660, las máquinas utilizadas oscilan entre 2 y 6, y se fijan un número máximo de jobs para cada problema.

La incorporación de orientaciones alternativas en las piezas de las instancias sigue la siguiente metodología. La primera orientación proviene de los datos recogidos del artículo de Li et al., (2017). La segunda orientación se basa en reducir la altura y aumentar la superficie de cada pieza en unas proporciones determinadas. Para ello se recorre el conjunto de piezas de la instancia, según su numeración, considerando grupos de 5 piezas. Cada una de las piezas del grupo verá reducida su altura en un factor de  $1/2$ ,  $2/3$ ,  $3/4$ ,  $4/5$  y  $5/6$  respectivamente. A cambio se aumenta la superficie de dichas piezas multiplicando por el factor inverso, es decir,  $(2, 3/2, 4/3, 5/4$  y  $6/5)$  manteniendo la relación “altura\*superficie”. Este proceso se realiza iterativamente hasta recorrer todas las piezas del problema. Como se puede observar, el volumen de cada pieza se mantiene constante en todas

las orientaciones al mantener la relación “altura\*superficie”. La tercera orientación se obtiene aplicando el mismo proceso anterior, pero considerando incrementos de altura y reducciones de superficie en las piezas. La Tabla 6 presenta los coeficientes utilizados para la generación de nuevas orientaciones en las instancias. Además, en la Tabla 7 se muestra un ejemplo de cálculo para los datos del problema nº1 de Li et al., (2017).

Tabla 6. Valores para la generación de instancias

Pieza	Relación altura O2	Relación superficie O2	Relación altura O3	Relación superficie O3
1	1/2	2	2	1/2
2	2/3	3/2	3/2	2/3
3	3/4	4/3	4/3	3/4
4	4/5	5/4	5/4	4/5
5	5/6	6/5	6/5	5/6

Tabla 7. Ejemplo cálculo de nuevas orientaciones para el problema nº1 de Li et al., (2017)

Pieza	Altura O1	Superficie O1	Altura O2	Superficie O2	Altura O3	Superficie O3	Volumen
1	29,72	924,34	14,86	1.848,68	59,44	462,17	12.504,71
2	9,94	315,12	6,63	472,68	14,91	210,08	2.023,74
3	17,13	48,27	12,85	64,36	22,84	36,2	315
4	2,67	84,97	2,14	1.06,21	3,34	67,98	121,82
5	16,02	1.302,15	13,35	1.562,58	19,22	1.085,13	3.527,93
6	11,77	1.126,33	5,89	2.252,66	23,54	563,17	3.907,79
7	33,23	248,68	22,15	373,02	49,85	165,79	4.235,62
8	32,64	243,62	24,48	324,83	43,52	182,72	3.843,08
9	12,53	269,66	10,02	337,08	15,66	215,73	1.786,36
10	18,09	175,77	15,08	210,92	21,71	146,48	1.885

Los datos de cada problema se han recibido en formato Excel. Se han desarrollado ficheros de conversión de instancias para modificar de manera recurrente los ficheros utilizando la ventana programador de Excel y el lenguaje de programación de Excel VBA. Estos programas hacen referencia a la ejecución iterativa de actuadores para el preprocesado de datos y adecuación del formato de lectura de los ficheros.

Los ficheros capturados por el algoritmo de hormigas deben introducirse en formato simple de texto con extensión “.amo”, con una estructura predefinida para la correcta ejecución del código. La adaptación de los ficheros Excel a ficheros de texto con la correspondiente extensión se ha realizado mediante programación en Python. De esta manera, se ha conseguido reducir los esfuerzos de manipulación de datos y adaptación de instancias. Se aportan los códigos de manipulación de datos en el anexo 7.4.

Los ficheros de datos de entrada del problema contienen información de las orientaciones de cada pieza y de las características específicas de cada máquina. Los ficheros tendrán 4 partes claramente diferenciadas. La primera de ellas contiene la información del número de piezas, jobs máximo y máquinas. La segunda parte contendrá la información del volumen de las piezas. La tercera aportará los datos de las distintas orientaciones de las piezas. Y concluyendo, la cuarta sección con las características de las máquinas utilizadas.

A continuación, se aporta la estructura de fichero de entrada de datos con extensión “.amo” con la información de las piezas y las máquinas. Se muestran como ejemplo los datos del problema 1 con 3 orientaciones, que tendrá el nombre de “Set\_P1\_3O.amo”.

```
Número parts:
10
Número jobs máximo:
6
Número máquinas:
2
```

Ilustración 18. Fichero entrada sección inicial

```
*****
Part  # orient volumen
1     3     12504.71
2     3     2023.74
3     3     315.00
4     3     121.82
5     3     3527.93
6     3     3907.79
7     3     4235.62
8     3     3843.08
9     3     1786.36
10    3     1885.00
```

Ilustración 19. Fichero entrada sección volumen

```
*****
part.  orient  Altura  Sup
1      1      29.72   924.34
1      2      14.86   1848.68
1      3      59.44   462.17
2      1      9.94    315.12
2      2      6.63    472.68
2      3      14.91   210.08
3      1      17.13   48.27
3      2      12.85   64.36
3      3      22.84   36.20
4      1      2.67    84.97
4      2      2.14    106.21
4      3      3.34    67.98
5      1      16.02   1302.15
5      2      13.35   1562.58
5      3      19.22   1085.13
6      1      11.77   1126.33
6      2      5.89    2252.66
6      3      23.54   563.17
7      1      33.23   248.68
7      2      22.15   373.02
7      3      49.85   165.79
8      1      32.64   243.62
8      2      24.48   324.83
8      3      43.52   182.72
9      1      12.53   269.66
9      2      10.02   337.08
9      3      15.66   215.73
10     1      18.09   175.77
10     2      15.08   210.92
10     3      21.71   146.48
```

Ilustración 20. Fichero entrada sección orientaciones

mach.	TC	(1/VT)	MC	HT	SET	HC	MA	MH
1	60	32.4	2	0.7	2	20	625	32.5
2	80	32.4	2	0.7	1	20	1600	40

Ilustración 21. Fichero entrada sección máquinas

## 5.2 Ajuste de parámetros del algoritmo

### 5.2.1 Ajuste de parámetros de la función “Greedy”

La función “greedy” utilizada consta de cuatro parámetros alfas referentes a cada término en consideración. Los términos considerados son los incrementos en superficie, volumen, altura y coste al insertar una nueva pieza en un job. Los valores que puede tomar alfa varían de 0 a 100, consistiendo en una gran combinación de posibilidades manteniendo la relación de que la suma de todos los alfas debe ser igual a 100. La primera ejecución del código ha implementado únicamente la heurística de construcción semiparalela permitiendo un rango de alfas completo. Esta primera experimentación ha almacenado todos los resultados y sus valores de alfa correspondientes para la selección de instancias consideradas como referencia.

Con el objeto de reducir el tiempo de computación de la ejecución de la heurística semi-paralela en el algoritmo final, se pretende encontrar un rango de parámetros alfas que proporcione buenos resultados y que consecuentemente, permita reducir el número de combinaciones ejecutadas por la heurística. Para ello se han analizado las soluciones obtenidas en aquellas instancias que contienen el mayor número de piezas y ofrecen una visión significativa del espectro de soluciones. Las instancias seleccionadas han sido del 26-30, junto a las del 38-42; consistiendo en un total de 10 instancias, ejecutadas para una, dos y tres orientaciones. Esto asciende a 30 ejecuciones con tiempos de computación que ascendieron desde 35 minutos, hasta los 1.708 minutos (28.5 horas) para las instancias de mayor complejidad.

Analizando los valores obtenidos en la ejecución se selecciona aquella combinación de parámetros que obtiene el menor coste de la función objetivo para cada una de las ejecuciones. Para estos resultados se selecciona el mayor rango de variación de alfa posible. Se consigue mediante la obtención de los valores mínimo y máximo de cada alfa, en cada ejecución. Este método pretende conseguir un rango de valores para cada uno de los alfas, para ello, agruparemos los valores por instancia y volveremos a identificar los mínimos y máximos. De esta manera habremos conseguido un rango sobre el que variar los parámetros de manera que cubra el espectro de posibles soluciones con menor coste.

Los valores finales que utilizaremos en el algoritmo de hormigas corresponderán al cuartil más próximo al mínimo y máximo obtenido; pudiendo tomar los valores de 0, 25, 50, 75 o 100.

El método utilizado para obtener estos parámetros sigue una estructura de cálculo sencilla adecuada para la selección de estos valores, ya que busca reducir el espectro de combinaciones a recorrer por la metaheurística final utilizada. Debido a haber hecho uso de la heurística semiparalela en esta etapa, ejecutando todas las posibles combinaciones de alfas permitiendo su variación desde 0 hasta 100 y aumentando en 1 su valor, podemos acercarnos a un buen rango de valores a utilizar. Pero el verdadero motor de optimización de soluciones vendrá marcado por la ejecución del algoritmo de hormigas. En ejecuciones posteriores no solo se reducirá la variación de alfas al espectro marcado en este apartado, sino que también se considerará modificar dichos valores en razón de cuartiles (25), evitando una experimentación tediosa sobre el valor alfa y permitiendo al algoritmo memético mayor tiempo de procesamiento. En definitiva, el valor obtenido en la heurística semiparalela marcará el valor inicial de la función objetivo sobre el que el algoritmo de hormigas realizará la optimización.

A continuación, se aporta la tabla resumen de los rangos finales de alfas. El completo de los datos de la experimentación se muestran en el anexo 7.1.

Tabla 8. Resumen de rangos de alfa

Instancia	Alfa 1 min	Alfa 1 max	Alfa 2 min	Alfa 2 max	Alfa 3 min	Alfa 3 max	Alfa 4 min	Alfa 4 max
26	6	16	63	78	8	15	0	14
27	5	9	59	77	5	23	1	23
28	6	17	58	73	3	17	0	33
29	2	24	57	73	8	30	0	28
30	8	16	52	74	6	23	0	19
38	3	20	61	73	8	16	2	28
39	5	7	56	72	3	14	7	36
40	6	13	55	81	4	18	2	31
41	2	11	53	80	1	30	2	41
42	8	10	70	76	5	20	0	17
(Min-Max)	2	24	52	81	1	30	0	41

Observando los valores resultantes de la tabla anterior seleccionaremos los rangos de alfa de la siguiente manera.

- Alfa 1: variará de 0 a 25.
- Alfa 2: variará de 50 a 75.
- Alfa 3: variará de 0 a 25.
- Alfa 4: variará en función de los valores de los demás alfas, ya que la suma de todos ellos debe ser igual a 100.

El procesado de los valores de alfa se ha realizado extrayendo los datos del fichero de resultados .amo y exportándolos a Excel mediante un programa codificado en lenguaje Python. La gran cantidad de datos referentes a las distintas combinaciones hace necesario la creación de código de exportación de datos. El código referente a este fichero .py se añade en el anexo 7.5.

Como resultado de este ajuste la heurística semiparalela generará la solución inicial en un espectro de soluciones considerablemente bueno para comenzar con la metaheurística. Esto nos acerca a la mejor solución en el proceso previo al inicio de la metaheurística sin necesidad de dedicar mucho tiempo computacional.

### 5.2.2 Ajuste de parámetros del algoritmo memético

Se ha realizado la experimentación oportuna para realizar un análisis y ajuste de los valores que intervienen en el algoritmo memético. Anteriormente se han seleccionado los valores que tendrán los cuatro alfas involucrados en la heurística constructiva semiparalela y serán estos los utilizados durante esta fase. Es necesario seleccionar un rango de variación del parámetro alfa que denote la velocidad con que va cambiando el parámetro y restrinja el número de iteraciones a realizar. Debido a los rangos de alfa establecidos en el apartado anterior se ha seleccionado 25 como rango de variación permitiendo 7 combinaciones posibles entre los alfas.

Los parámetros que serán ajustados en este apartado son  $q0$  (variable aleatoria de decisión), beta (parámetro de control de influencia de la atractividad),  $N$  (número de iteraciones máxima) y  $M$  (población de hormigas).

La experimentación se ha estructurado en dos fases. La primera destinada a encontrar los valores de  $q0$  y beta, los parámetros relacionados con la elección de un camino. La segunda recibe el resultado del primer paso y realiza el estudio para los valores de  $N$  y  $M$ , que determinan la extensión de las iteraciones que realiza el experimento.

Los parámetros alfa referente a la feromona se ha mantenido en 1 y el valor de evaporación  $\rho$  (Rho) en 0.2 en todas las experimentaciones en base a ser el mejor valor para obtener soluciones en la literatura (Molina et al., 2020a).

La ejecución de estas fases se ha realizado sobre una selección de instancias. Aquellas instancias seleccionadas

para la experimentación de parámetros cumplen la condición de tener un número reducido de piezas, alrededor de 30 piezas, y que presenten multiorientación, 2 o 3 orientaciones. De esta manera las instancias utilizadas han sido: 14 y 16, en el caso de 3 orientaciones, y 15 y 17 en el caso de 2 orientaciones. Se han utilizado las mismas instancias en todas las etapas. Cada ejecución de parámetros se ha realizado 5 veces para cada instancia, debido a la posibilidad de obtener distintos resultados en cada ejecución. El cálculo de la desviación media ha sido el utilizado para la selección del mejor parámetro. A continuación, se describen las 2 fases y los valores alcanzados.

### 5.2.2.1 Primera fase de experimentación

Esta primera fase está relacionada con los parámetros  $q0$  y  $\beta$ . El valor  $q0$  hace referencia a la probabilidad realizar la mejor inserción de piezas en jobs. El valor de  $\beta$  incide en el exponente de la componente de atractividad.

Los valores a experimentar para  $q0$  son 75% y 85%, combinados con los de  $\beta$  que son 0.5, 0.7, 1, 1.5 y 2. Esto supone 200 ejecuciones en esta fase. Los valores referentes a esta experimentación pueden encontrarse en el anexo 7.2. En esta fase se han mantenido constantes los valores de  $N$  y  $M$  en 20%.

Para obtener los valores de esta fase comenzamos calculando la desviación del valor de las 5 ejecuciones de cada instancia respecto al valor mínimo encontrado en los resultados de cada instancia. Con esto obtenemos las desviaciones medias de cada problema que podemos ver en las siguientes tablas.

Tabla 9. Fase 1 | Instancia 14 | 3 orientaciones

$\beta \backslash q0$	75%	85%
0.5	0,00937%	0,00935%
0.7	0,00935%	0,00744%
1	0,00635%	0,01020%
1.5	0,00852%	0,01045%
2	0,01060%	0,00882%

Tabla 10. Fase 1 | Instancia 15 | 2 orientaciones

$\beta \backslash q0$	75%	85%
0.5	0,00%	0,00%
0.7	0,00%	0,00%
1	0,00%	0,00%
1.5	0,00%	0,00%
2	0,00%	0,00%

Tabla 11. Fase 1 | Instancia 16 | 3 orientaciones

$\beta \backslash q0$	75%	85%
0.5	0,00001%	0,00168%
0.7	0,00027%	0,00168%
1	0,00027%	0,00111%
1.5	0,00081%	0,00252%
2	0,00001%	0,00111%



Tabla 12. Fase 1 | Instancia 17 | 2 orientaciones

beta\q0	75%	85%
0.5	0,00156%	0,00156%
0.7	0,00156%	0,00198%
1	0,00198%	0,00158%
1.5	0,00232%	0,00156%
2	0,00198%	0,00125%

Haciendo uso de estos resultados se ha calculado la desviación media de las 4 instancias, obteniendo los resultados para la selección de los mejores parámetros en la siguiente tabla marcado en verde.

Tabla 13. Fase 1 | Desviación media

beta\q0	75%	85%
0.5	0,00274%	0,00315%
0.7	0,00280%	0,00277%
1	0,00215%	0,00322%
1.5	0,00291%	0,00363%
2	0,00315%	0,00279%

Podemos observar que el menor valor de desviación media se obtiene para  $q0 = 75\%$  y  $\beta = 1$ . Estos son los valores seleccionados y que serán fijados en las posteriores ejecuciones de código. Es interesante destacar que la desviación que sufren las distintas ejecuciones es considerablemente pequeña como para resaltar la baja influencia que presenta la variación de estos parámetros con la obtención de la función objetivo. El algoritmo memético es capaz de obtener un valor de coste eficaz para la mayoría de combinación de parámetros.

### 5.2.2.2 Segunda fase de experimentación

La segunda fase está destinada a seleccionar la mejor combinación de los parámetros relacionados con las iteraciones ( $N$ ) y la población de hormigas ( $M$ ). Estos parámetros hacen referencia a lo extensa que será la experimentación de cada ejecución siendo  $N$  el número máximo de iteraciones y  $M$  la población de hormigas a implementar en el algoritmo. En el estudio estos valores se han implementado como valores enteros expresados en porcentaje. Para calcular el dato de  $N$  y  $M$  se extrae el número de piezas característico de la instancia y se calcula el porcentaje correspondiente.

Los valores que se han experimentado en esta fase son 10%, 20% y 30% para ambos  $N$  y  $M$ , suponiendo 180 ejecuciones. Los valores referentes a esta experimentación pueden encontrarse en el anexo 7.3.

Se ha seguido la misma metodología de cálculo que en la primera fase, correspondiente a la obtención de la desviación media de las instancias. Las siguientes tablas ofrecen los valores porcentuales de las desviaciones de cada instancia.

Tabla 14. Fase 2 | Instancia 14 | 3 orientaciones

NM	10%	20%	30%
10%	0,00936%	0,01170%	0,01116%
20%	0,00936%	0,01046%	0,00691%
30%	0,01170%	0,00670%	0,00235%

Tabla 15. Fase 2 | Instancia 15 | 2 orientaciones

N\M	10%	20%	30%
10%	0,00000%	0,00000%	0,00000%
20%	0,00000%	0,00000%	0,00000%
30%	0,00000%	0,00000%	0,00000%

Tabla 16. Fase 2 | Instancia 16 | 3 orientaciones

N\M	10%	20%	30%
10%	0,00344%	0,00107%	0,00160%
20%	0,00110%	0,00251%	0,00027%
30%	0,00277%	0,00053%	0,00053%

Tabla 17. Fase 2 | Instancia 17 | 2 orientaciones

N\M	10%	20%	30%
10%	0,00077%	0,00092%	0,00051%
20%	0,00001%	0,00001%	0,00001%
30%	0,00001%	0,00001%	0,00001%

Haciendo uso de estos resultados se ha calculado la desviación media de las 4 instancias, obteniendo los resultados para la selección de los mejores parámetros en la siguiente tabla marcado en verde.

Tabla 18. Fase 2 | Desviación media

N\M	10%	20%	30%
10%	0,00339%	0,00342%	0,00332%
20%	0,00262%	0,00324%	0,00180%
30%	0,00362%	0,00181%	0,00072%

Podemos observar que el menor valor de desviación media se obtiene para  $N = 30\%$  y  $M = 30\%$ . Estos son los valores seleccionados y que serán fijados en las posteriores ejecuciones de código. De nuevo destaca el reducido valor que tienen las desviaciones. Del mismo modo que la fase anterior podemos concluir que la variación de estos parámetros no tiene una fuerte incidencia en la obtención de la mejor solución por parte del algoritmo memético.

### 5.3 Resultados computacionales del algoritmo memético ACS

Tras haber realizado los ajustes de parámetros ( $q0 = 75\%$ ,  $beta = 1$ ,  $N = 30\%$  y  $M = 30\%$ ) procedemos a ejecutar el algoritmo Mem-ACS en las diferentes instancias obteniendo los valores de función objetivo de costes y tiempo de ejecución (minutos). Los valores son mostrados según el tipo de instancia (consideración de 1, 2 o 3 orientaciones) y los resultados son comparados con los obtenidos por otros métodos de resolución. A continuación, se lleva a cabo el estudio de la eficiencia del uso de la búsqueda local en el Mem-ACS comparando los resultados obtenidos con los de un algoritmo evolutivo ACS. La siguiente Tabla 19 muestra las características de cada instancia y su indicador de dificultad. El indicador de dificultad ha sido calculado según “*número de piezas*\**número de máquinas*”. Este indicador será utilizado en el apartado 5.3.4 para la evaluación de la eficiencia del Mem-ACS.

Tabla 19. Características de las instancias

Instancia	Número de piezas	Número de máquinas	Volumen total de piezas	Indicador de dimensión del problema
1	10	2	34151,05	20
2	12	2	51277,84	24
3	14	2	37716,64	28
4	16	2	52972,41	32
5	18	2	51753,09	36
6	20	2	97587,83	40
7	15	3	57286,09	45
8	18	3	71312,58	54
9	21	3	57732,1	63
10	24	3	84383,74	72
11	27	3	95146,95	81
12	30	3	119275,1	90
13	20	4	81692,75	80
14	24	4	92724,76	96
15	28	4	98740,47	112
16	32	4	116572,13	128
17	36	4	144202,89	144
18	40	4	140852,58	160
19	25	5	164243,07	125
20	30	5	111738,66	150
21	35	5	149708,41	175
22	40	5	181401,63	200
23	60	5	230665,28	300
24	80	5	264204,19	400
25	100	5	323539,32	500
26	120	5	358871,69	600
27	140	5	497812,86	700
28	160	5	590828,26	800
29	180	5	741890,72	900
30	200	5	768641,54	1000
31	30	6	109733,88	180
32	60	6	193419,12	360
33	90	6	267478,42	540
34	120	6	422908,39	720
35	160	6	568003,38	960
36	200	6	749480,96	1200
37	250	6	1051179,1	1500
38	300	6	978095,92	1800
39	360	6	1317484,3	2160
40	420	6	1618653,1	2520
41	590	6	2317876,7	3540
42	660	6	2528683,5	3960

### 5.3.1 Comparación para problemas con 1 orientación

Este apartado presenta los resultados computaciones obtenidos del algoritmo Mem-ACS para los problemas de 1 orientación. La Tabla 21 muestra los resultados obtenidos de la heurística semiparalela y la Tabla 23 los obtenidos por nuestro algoritmo memético, comparando en ambos casos con las mejores soluciones encontradas en la literatura científica, en concreto en el artículo de Li et al., (2017). Este artículo muestra resultados de dos heurísticas denominadas BF ("best fit") y ABF ("adapted best fit"), junto con los resultados obtenidos de ejecutar mediante el software de optimización CPLEX. Los métodos de resolución son referenciados en la Tabla 21 y Tabla 23 usando las siguientes abreviaturas: CPLEX, BF y ABF. Los resultados obtenidos mediante CPLEX fueron obtenidos con un tiempo máximo de computación de 8.000 segundos.

La Tabla 21 y Tabla 23, muestran los valores de la FO alcanzado por cada uno de los métodos comparados. De igual manera ofrece la desviación o GAP obtenido (medido en %) por los valores de FO obtenidos por el la heurística semiparalela y Mem-ACS respecto a los demás métodos de optimización. Los valores negativos indican que el método bajo estudio (heurística semiparalela o Mem.ACS) ha alcanzado mejor solución que el método con el que se ha contrastado.

#### 5.3.1.1 Comparación para resultados de la heurística semiparalela en problemas con 1 orientación

Inicialmente se analizarán los resultados obtenidos por la heurística semiparalela respecto a CPLEX, BF y ABF.

Se puede ver en la Tabla 20 la cantidad de ocasiones en las que se ha igualado, mejorado y empeorado la solución de la heurística respecto a los demás métodos de optimización. La Tabla 21 presenta los datos comparativos por instancia de la heurística semiparalela con el resto de métodos, se indican en negrita los mejores valores de FO alcanzados.

El análisis respecto a CPLEX nos muestra que este método es el que consigue mejora en mayor medida los resultados la heurística en promedio (-2,056%). Esto contrasta con la cantidad de instancias para las que empeora los resultados (50,00%), siendo el método que mayor porcentaje de empeoramiento presenta. Esto se debe a que es en esta comparativa en la que se obtiene el valor máximo de mejora (-36.761%, obtenido en la instancia 36). También es para este método donde se obtiene el mayor empeoramiento (0,364%). Los valores identificados como N/A son debidos a que CPLEX no encontró solución.

Al analizar los resultados respecto a la heurística BF se observa que en promedio la heurística semiparalela consigue mejorar los resultados (-0,012%). En un 50% de ocasiones consigue mejorarlos. Además, este es el caso en el que se igualan los resultados en una mayor proporción (12,50%).

El caso de la comparación respecto al ABF podemos encontrar que es para el que se alcanza el menor valor de mejora promedio (-0.009%). En cambio, ofrece el mayor porcentaje de mejora en instancias individuales (60%).

Tabla 20. Análisis GAP Heurística Semiparalela | 1 orientación

	Contador GAP (CPLEX)	GAP % (CPLEX)	Contador GAP (BF)	GAP % (BF)	Contador GAP (ABF)	GAP % (ABF)
Igualado	3	7,50%	5	12,50%	0	0,00%
Mejora	7	17,50%	20	50,00%	24	60,00%
Empeora	21	52,50%	15	37,50%	16	40,00%
N/A	9	22,50%	0	0,00%	0	0,00%
GAP Promedio		-2,056%		-0,012%		-0,009%

Tabla 21. Comparación resultados Heurística Semiparalela | 1 orientación

Instancia	CPLEX	GAP (CPLEX)	BF	GAP (BF)	ABF	GAP (ABF)	Heurística semiparalela
1	<b>153.574,88</b>	0,071%	153.683,48	0,000%	<b>153.574,88</b>	0,071%	153.683,72
2*	233.773,11	-0,003%	233.882,84	-0,050%	233.773,11	-0,003%	<b>233.766,81</b>
3	<b>284.267,30</b>	0,000%	<b>284.267,30</b>	0,000%	284.426,84	-0,056%	<b>284.267,30</b>
4	<b>392.157,40</b>	0,000%	<b>392.157,40</b>	0,000%	392.456,69	-0,076%	<b>392.157,40</b>
5	<b>386.015,43</b>	0,019%	386.087,88	0,000%	386.148,44	-0,016%	386.088,34
6	<b>764.638,71</b>	0,047%	764.967,58	0,004%	765.692,66	-0,091%	764.994,63
7	257.311,93	0,032%	257.691,74	-0,115%	<b>257.291,31</b>	0,040%	257.394,11
8	<b>326.774,21</b>	0,254%	327.360,40	0,075%	327.627,82	-0,007%	327.605,31
9	<b>463.772,93</b>	0,364%	465.051,12	0,088%	465.246,83	0,046%	465.460,38
10	<b>618.890,60</b>	0,037%	<b>618.890,60</b>	0,037%	619.658,49	-0,087%	619.120,44
11	<b>705.307,21</b>	0,258%	705.758,21	0,194%	705.502,27	0,230%	707.124,69
12	851.207,94	0,026%	<b>851.070,78</b>	0,042%	851.453,65	-0,003%	851.429,25
13	<b>362.609,61</b>	0,000%	<b>362.609,61</b>	0,000%	362.621,05	-0,003%	<b>362.609,61</b>
14	<b>422.859,21</b>	0,194%	423.700,23	-0,005%	423.202,30	0,112%	423.678,06
15	<b>671.998,02</b>	0,231%	672.508,50	0,155%	672.406,80	0,170%	673.550,25
16	<b>815.294,99</b>	0,024%	815.736,79	-0,031%	815.494,32	-0,001%	815.487,25
17	<b>1.006.647,21</b>	0,118%	1.008.839,09	-0,099%	1.007.182,20	0,065%	1.007.840,00
18	<b>1.006.650,85</b>	0,075%	1.007.453,71	-0,005%	1.007.087,50	0,032%	1.007.407,69
19	<b>1.234.983,06</b>	0,030%	1.235.129,24	0,018%	1.235.153,87	0,016%	1.235.352,50
20	<b>486.363,75</b>	0,052%	486.948,14	-0,068%	486.863,22	-0,051%	486.615,03
21	<b>674.093,55</b>	0,066%	674.584,60	-0,007%	674.813,65	-0,041%	674.538,88
22	<b>802.417,41</b>	0,198%	803.268,19	0,092%	803.865,00	0,018%	804.007,38
23	<b>1.054.271,81</b>	0,190%	1.055.957,97	0,030%	1.055.256,75	0,096%	1.056.274,63
24	1.211.746,10	-0,080%	1.210.961,41	-0,016%	1.211.516,24	-0,061%	<b>1.210.773,13</b>
25	2.542.744,05	-1,717%	2.499.865,38	-0,031%	2.499.865,38	-0,031%	<b>2.499.085,75</b>
26*	1.651.631,59	2,468%	1.646.549,97	2,784%	<b>1.645.193,43</b>	2,869%	1.692.397,13
27	-	-	<b>2.281.570,92</b>	0,038%	2.282.317,64	0,005%	2.282.432,50
28	3.552.354,91	-23,957%	2.701.739,47	-0,016%	2.702.135,32	-0,031%	<b>2.701.303,50</b>
29	-	-	3.389.394,52	-0,034%	3.390.195,77	-0,057%	<b>3.388.253,75</b>
30	-	-	3.513.145,34	-0,073%	3.514.982,39	-0,125%	<b>3.510.577,50</b>
31	<b>502.008,36</b>	0,196%	502.352,92	0,127%	502.776,50	0,043%	502.990,53
32	<b>882.962,15</b>	0,215%	884.571,40	0,033%	884.395,39	0,053%	884.863,63
33	1.226.733,6	-0,192%	1.225.567,40	-0,097%	<b>1.223.815,41</b>	0,046%	1.224.373,50
34	1.997.616,2	-3,202%	1.935.211,88	-0,080%	1.935.055,40	-0,072%	<b>1.933.658,38</b>
35	2.613.610,7	-0,519%	2.600.103,63	-0,002%	2.600.768,20	-0,028%	<b>2.600.051,00</b>
36	5.414.460,3	-36,761%	3.424.296,06	-0,007%	<b>3.423.224,31</b>	0,024%	3.424.042,50
37	-	-	<b>4.798.043,93</b>	0,003%	4.798.937,43	-0,016%	4.798.164,00
38	-	-	<b>4.471.150,32</b>	0,004%	4.471.717,61	-0,008%	4.471.341,50
39	-	-	5.859.498,43	-0,217%	5.857.324,58	-0,180%	<b>5.846.757,50</b>
40	-	-	7.209.221,88	-0,164%	7.205.320,92	-0,110%	<b>7.197.391,50</b>

41	-	-	10.330.915,6	-0,161%	10.327.716,9	-0,130%	<b>10.314.315,0</b>
42	-	-	11.265.891,6	-0,182%	11.263.691,7	-0,163%	<b>11.245.334,0</b>
		-2,056%		-0,012%		-0,009%	

\*Son problemas que no se han tenido en cuenta para el análisis. El volumen total de sus piezas no coincide con el ofrecido en los resultados del artículo.

### 5.3.1.2 Comparación para resultados del Mem-ACS en problemas con 1 orientación

Se continúa contrastando los valores obtenidos por el Mem-ACS respecto a los métodos CPLEX, BF, ABF y Heurística semiparalela.

Se puede ver en la Tabla 22 y Tabla 23 la cantidad de ocasiones en las que se ha igualado, mejorado y empeorado la solución del Mem-ACS respecto a los demás métodos de optimización. La Tabla 21 presenta los datos comparativos por instancia del Mem-ACS con el resto de métodos.

Los valores de GAP analizados para el Mem-ACS indican una mejora promedio del -0,579% respecto a los 4 métodos contrastados. Este dato nos recomienda utilizar el método propuesto frente al resto de métodos. Además, la media del GAP para todos los métodos nos indica que el Mem-ACS obtiene mejores soluciones que los otros métodos (CPLEX -2,192%, BF -0,167%, ABF -0,165% y Heurística semiparalela -0,155%). Respecto a los valores medios del GAP podemos afirmar que el Mem-ACS supone una mejora en mayor medida respecto a los valores de CPLEX y se encuentra más próximo a los obtenidos mediante ABF.

La comparativa respecto a CPLEX nos ofrece un porcentaje de igualdad de soluciones del 35% suponiendo el mayor entre todos los métodos. También presenta una mejora del 35,00% que acumuladas ascienden a 72,50% de instancias igualadas o mejoradas. Es en este contraste donde se encuentra el mayor valor de mejora del Mem-ACS (-36,888%, instancia 36) y también el peor (0,114%, instancia 8).

Ambos métodos BF y ABF acumulan un porcentaje del 97,50% de instancias igualadas y mejoradas por el Mem-ACS pero es respecto al ABF frente al que se obtiene un mayor porcentaje de mejora (BF 87,50%, ABF 95%).

Respecto a la Heurística Semiparalela se consigue una mejora de 92,50% e igualación de 7,5% de resultados.

Todos los valores medios calculados indican que de forma general el Mem-ACS es capaz de obtener mejores soluciones únicamente empeorando en 4 instancias (2 frente a CPLEX, 1 frente a BF y 1 frente a ABF). Siendo este el mejor método para realizar optimización de problemas de FA considerando 1 orientación por pieza.

Tabla 22. Análisis GAP Mem-ACS | 1 orientación

	Contador GAP (CPLEX)	GAP % (CPLEX)	Contador GAP (BF)	GAP % (BF)	Contador GAP (ABF)	GAP % (ABF)	Contador GAP (Heu. Semi.)	GAP % (Heu. Semi.)
Igualado	15	37,50%	4	10,00%	1	2,50%	3	7,50%
Mejora	14	35,00%	35	87,50%	38	95,00%	37	92,50%
Empeora	2	5,00%	1	2,50%	1	2,50%	0	0,00%
N/A	9	22,50%	0	0,00%	0	0,00%	0	0,00%
GAP Promedio		-2,192%		-0,167%		-0,165%		-0,155%

Cabe destacar que el Mem-ACS ha obtenido nuevas mejores soluciones que el resto de instancias con menor número de jobs para las instancias 15, 16, 17, 22, 23 y 32.

Los datos mostrados en negrita en la Tabla 23 indican la mejor solución alcanzada para cada instancia.

Tabla 23. Comparación resultados Mem-ACS | 1 orientación

Instancia	CPLEX	GAP (CPLEX)	BF	GAP (BF)	ABF	GAP (ABF)	Heurística semiparalela	GAP (Semi-parallel)	Mem-ACS
1	<b>153.574,88</b>	0,000%	153.683,48	-0,071%	<b>153.574,88</b>	0,000%	153.683,72	-0,071%	<b>153.574,88</b>
2*	233.773,11	-0,003%	233.882,84	-0,050%	233.773,11	-0,003%	<b>233.766,81</b>	0,000%	233.766,81
3	<b>284.267,30</b>	0,000%	<b>284.267,30</b>	0,000%	284.426,84	-0,056%	<b>284.267,30</b>	0,000%	<b>284.267,30</b>
4	<b>392.157,40</b>	0,000%	<b>392.157,40</b>	0,000%	392.456,69	-0,076%	<b>392.157,40</b>	0,000%	<b>392.157,40</b>
5	<b>386.015,43</b>	0,000%	386.087,88	-0,019%	386.148,44	-0,034%	386.088,34	-0,019%	<b>386.015,43</b>
6	<b>764.638,71</b>	0,000%	764.967,58	-0,043%	765.692,66	-0,138%	764.994,63	-0,047%	<b>764.638,71</b>
7	257.311,93	0,000%	257.691,74	-0,147%	<b>257.291,31</b>	0,008%	257.394,11	-0,032%	257.311,93
8	<b>326.774,21</b>	0,114%	327.360,40	-0,065%	327.627,82	-0,146%	327.605,31	-0,140%	327.148,25
9	<b>463.772,93</b>	0,000%	465.051,12	-0,275%	465.246,83	-0,317%	465.460,38	-0,363%	<b>463.772,93</b>
10	<b>618.890,60</b>	0,000%	<b>618.890,60</b>	0,000%	619.658,49	-0,124%	619.120,44	-0,037%	<b>618.890,60</b>
11	<b>705.307,21</b>	0,000%	705.758,21	-0,064%	705.502,27	-0,028%	707.124,69	-0,257%	<b>705.307,21</b>
12	851.207,94	-0,014%	<b>851.070,78</b>	0,002%	851.453,65	-0,043%	851.429,25	-0,040%	851.091,56
13	362.609,61	-0,929%	362.609,61	-0,929%	362.621,05	-0,932%	362.609,61	-0,929%	<b>359.240,94</b>
14	<b>422.859,21</b>	0,000%	423.700,23	-0,198%	423.202,30	-0,081%	423.678,06	-0,193%	<b>422.859,21</b>
15	<b>671.998,02</b>	0,000%	672.508,50	-0,076%	672.406,80	-0,061%	673.550,25	-0,230%	<b>671.998,02</b>
16	815.294,99	-0,029%	815.736,79	-0,083%	815.494,32	-0,053%	815.487,25	-0,052%	<b>815.062,19</b>
17	1.006.647,21	-0,001%	1.008.839,09	-0,218%	1.007.182,20	-0,054%	1.007.840,00	-0,119%	<b>1.006.640,44</b>
18	1.006.650,85	-0,109%	1.007.453,71	-0,188%	1.007.087,50	-0,152%	1.007.407,69	-0,184%	<b>1.005.556,25</b>
19	<b>1.234.983,06</b>	0,000%	1.235.129,24	-0,012%	1.235.153,87	-0,014%	1.235.352,50	-0,030%	<b>1.234.983,06</b>
20	<b>486.363,75</b>	0,000%	486.948,14	-0,120%	486.863,22	-0,103%	486.615,03	-0,052%	<b>486.363,75</b>
21	<b>674.093,55</b>	0,019%	674.584,60	-0,054%	674.813,65	-0,088%	674.538,88	-0,047%	674.221,38
22	802.417,41	-0,012%	803.268,19	-0,118%	803.865,00	-0,192%	804.007,38	-0,210%	<b>802.319,38</b>
23	1.054.271,81	-0,063%	1.055.957,97	-0,223%	1.055.256,75	-0,156%	1.056.274,63	-0,252%	<b>1.053.607,88</b>
24	1.211.746,10	-0,315%	1.210.961,41	-0,250%	1.211.516,24	-0,296%	1.210.773,13	-0,234%	<b>1.207.934,88</b>

25	2.542.744,05	-1,942%	2.499.865,38	-0,260%	2.499.865,38	-0,260%	2.499.085,75	-0,229%	<b>2.493.372,00</b>
26*	1.651.631,59	2,210%	1.646.549,97	2,526%	<b>1.645.193,43</b>	2,610%	1.692.397,13	-0,252%	1.688.135,13
27	-	-	2.281.570,92	-0,209%	2.282.317,64	-0,242%	2.282.432,50	-0,247%	<b>2.276.799,00</b>
28	3.552.354,91	-24,114%	2.701.739,47	-0,222%	2.702.135,32	-0,236%	2.701.303,50	-0,206%	<b>2.695.745,25</b>
29	-	-	3.389.394,52	-0,192%	3.390.195,77	-0,215%	3.388.253,75	-0,158%	<b>3.382.891,25</b>
30	-	-	3.513.145,34	-0,198%	3.514.982,39	-0,250%	3.510.577,50	-0,125%	<b>3.506.194,00</b>
31	<b>502.008,36</b>	0,000%	502.352,92	-0,069%	502.776,50	-0,153%	502.990,53	-0,195%	<b>502.008,36</b>
32	882.962,15	-0,078%	884.571,40	-0,260%	884.395,39	-0,240%	884.863,63	-0,293%	<b>882.270,00</b>
33	1.226.733,6	-0,480%	1.225.567,40	-0,385%	1.223.815,41	-0,242%	1.224.373,50	-0,288%	<b>1.220.848,88</b>
34	1.997.616,2	-3,329%	1.935.211,88	-0,211%	1.935.055,40	-0,203%	1.933.658,38	-0,131%	<b>1.931.120,25</b>
35	2.613.610,7	-0,703%	2.600.103,63	-0,187%	2.600.768,20	-0,213%	2.600.051,00	-0,185%	<b>2.595.234,50</b>
36	5.414.460,3	-36,888%	3.424.296,06	-0,209%	3.423.224,31	-0,177%	3.424.042,50	-0,201%	<b>3.417.149,50</b>
37	-	-	4.798.043,93	-0,137%	4.798.937,43	-0,155%	4.798.164,00	-0,139%	<b>4.791.476,00</b>
38	-	-	4.471.150,32	-0,212%	4.471.717,61	-0,225%	4.471.341,50	-0,217%	<b>4.461.655,00</b>
39	-	-	5.859.498,43	-0,463%	5.857.324,58	-0,426%	5.846.757,50	-0,246%	<b>5.832.348,00</b>
40	-	-	7.209.221,88	-0,413%	7.205.320,92	-0,359%	7.197.391,50	-0,249%	<b>7.179.481,00</b>
41	-	-	10.330.915,6	-0,410%	10.327.716,9	-0,379%	10.314.315,0	-0,250%	<b>10.288.539,00</b>
42	-	-	11.265.891,6	-0,421%	11.263.691,7	-0,402%	11.245.334,0	-0,239%	<b>11.218.449,00</b>
Media		-2,192%		-0,167%		-0,165%		-0,155%	

\*Son problemas que no se han tenido en cuenta para el análisis. El volumen total de sus piezas no coincide con el ofrecido en los resultados del artículo.



### 5.3.2 Comparación para problemas con 2 orientaciones

Este apartado ofrece el análisis para los problemas de 2 orientaciones respecto a la FO medida en términos de costes de producción, con el objetivo de conocer el éxito del Mem-ACS respecto a los otros métodos de optimización. Se ha comparado con las soluciones ofrecidas por LINGO hasta la instancia 15 (ya que para el resto de instancias no ofreció resultados) y a la heurística semiparalela.

Se ofrecen en la Tabla 24 y Tabla 25 los datos y la diferencia porcentual existente entre el Mem-ACS y los demás métodos de optimización analizados.

Respecto a los valores medios del GAP podemos afirmar que el Mem-ACS supone una mejora respecto a los valores de los tres métodos comparados (-0,203%). La mayor diferencia media la obtiene respecto a la heurística semiparalela (-0,205%).

El Mem-ACS es capaz de obtener mejor resultado de función objetivo que LINGO (-0,197%). Únicamente se igualan los resultados para las instancias 1, 2, 6, 7 y 8. No se empeora el resultado en ningún caso. El valor máximo de mejora del Mem-ACS ha sido del -1,203%, conseguida para la instancia 11 respecto a LINGO.

El 100% de las instancias han obtenido mejoras (97,62%) o igualdad (2,38%) de resultados al utilizar el Mem-ACS respecto a la heurística semiparalela. En ningún caso se empeoraron los resultados.

Concluyendo, el Mem-ACS es el mejor método para realizar optimización de problemas de FA considerando 2 orientaciones por pieza.

Tabla 24. Análisis GAP Mem-ACS | 2 orientaciones

	Contador	GAP %	Contador	GAP %
	GAP (LINGO)	(LINGO)	GAP (Heu. Semi.)	(Heu. Semi.)
Igualado	5	11,90%	1	2,38%
Mejora	10	23,81%	41	97,62%
Empeora	0	0,00%	0	0,00%
N/A	27	64,29%	0	0,00%
GAP Promedio		-0,197%		-0,205%

Debemos resaltar que el Mem-ACS no está limitado por un número de jobs máximo ya que permite generar más jobs en el caso de que esto suponga un ahorro global de costes. Este es el caso de 26 (61,90%) instancias: 3 – 5, 10, 11, 13, 15, 19, 20, 25 – 30 y 32 – 42.

Tabla 25. Comparación resultados | 2 orientaciones

Instancia	LINGO	GAP (LINGO)	Heurística semiparalela	GAP (Semiparalela)	Mem-ACS
1	<b>149.341,64</b>	0,000%	<b>149.341,64</b>	0,000%	<b>149.341,64</b>
2	<b>233.543,39</b>	0,000%	233.964,50	-0,180%	<b>233.543,39</b>
3	282.071,20	-0,019%	283.034,28	-0,359%	<b>282.016,94*</b>
4	360.233,91	-0,573%	359.507,03	-0,371%	<b>358.171,53*</b>
5	384.081,43	-0,028%	384.549,03	-0,150%	<b>383.972,56*</b>
6	<b>698.117,11</b>	0,000%	699.895,94	-0,254%	<b>698.117,11</b>
7	<b>256.044,92</b>	0,000%	256.662,02	-0,240%	<b>256.044,92</b>
8	<b>326.235,36</b>	0,000%	326.841,84	-0,186%	<b>326.235,36</b>
9	443.650,78	-0,200%	443.909,53	-0,258%	<b>442.762,34</b>
10	595.195,98	-0,001%	595.741,94	-0,093%	<b>595.188,00*</b>
11	649.020,85	-1,203%	641.883,56	-0,104%	<b>641.215,63*</b>
12	835.955,48	-0,133%	835.097,00	-0,030%	<b>834.846,56</b>

13	351.503,42	-0,185%	351.438,72	-0,166%	<b>350.854,19*</b>
14	423.652,50	-0,266%	423.600,06	-0,253%	<b>422.527,19</b>
15	665.731,44	-0,346%	664.880,63	-0,218%	<b>663.430,00*</b>
16	-	-	795.175,56	-0,139%	<b>794.071,00</b>
17	-	-	949.282,81	-0,273%	<b>946.687,50</b>
18	-	-	968.398,75	-0,351%	<b>965.002,25</b>
19	-	-	1.216.488,75	-0,107%	<b>1.215.182,13*</b>
20	-	-	469.469,44	-0,104%	<b>468.983,50*</b>
21	-	-	669.024,25	-0,128%	<b>668.166,25</b>
22	-	-	794.911,81	-0,253%	<b>792.898,63</b>
23	-	-	1.054.798,63	-0,176%	<b>1.052.947,00</b>
24	-	-	1.210.153,25	-0,215%	<b>1.207.554,75</b>
25	-	-	2.450.462,00	-0,439%	<b>2.439.714,00*</b>
26	-	-	1.692.005,88	-0,242%	<b>1.687.903,38*</b>
27	-	-	2.282.221,25	-0,234%	<b>2.276.876,75*</b>
28	-	-	2.700.574,75	-0,170%	<b>2.695.978,00*</b>
29	-	-	3.389.147,75	-0,181%	<b>3.383.026,75*</b>
30	-	-	3.513.581,75	-0,206%	<b>3.506.340,75*</b>
31	-	-	502.122,06	-0,149%	<b>501.371,59</b>
32	-	-	883.887,75	-0,200%	<b>882.120,25*</b>
33	-	-	1.224.234,75	-0,293%	<b>1.220.644,88*</b>
34	-	-	1.935.337,38	-0,241%	<b>1.930.679,63*</b>
35	-	-	2.600.992,00	-0,225%	<b>2.595.139,00*</b>
36	-	-	3.427.675,50	-0,319%	<b>3.416.730,50*</b>
37	-	-	4.803.351,00	-0,252%	<b>4.791.254,00*</b>
38	-	-	4.472.456,00	-0,240%	<b>4.461.731,50*</b>
39	-	-	5.786.697,00	-0,204%	<b>5.774.919,00*</b>
40	-	-	7.093.875,00	-0,156%	<b>7.082.798,50*</b>
41	-	-	10.191.406,0	-0,142%	<b>10.176.942,0*</b>
42	-	-	11.111.496,0	-0,108%	<b>11.099.483,0*</b>
Media	-	-0,197%	-	-0,205%	-

\*Soluciones que utilizan mayor número de jobs que los definidos en las instancias de la literatura. El algoritmo memético permite utilizar más jobs en el caso de que esto suponga una ventaja en costes.

### 5.3.3 Comparación para problemas con 3 orientaciones

Este apartado ofrece el análisis para los problemas de 3 orientaciones respecto a la FO medida en términos de costes de producción, con el objetivo de conocer el éxito del Mem-ACS respecto a los otros métodos de optimización. Se ha comparado respecto a las soluciones ofrecidas por LINGO hasta la instancia 15 y a la heurística semiparalela.

Se ofrecen en la Tabla 26 y Tabla 27 los datos y la diferencia porcentual existente entre el Mem-ACS y los demás métodos de optimización analizados (LINGO y Heurística semiparalela).

Respecto a los valores medios del GAP podemos afirmar que el Mem-ACS supone una mejora respecto a los valores de los dos métodos comparados (-0,188%). La mayor diferencia en promedio la obtiene respecto a LINGO (-0,222%). Supone un -0,174% en el caso de la heurística.

Para ningún caso la mejora del Mem-ACS ha superado el -1,24%. Esta ha sido la mayor mejora alcanzada, en

la instancia 12 respecto a LINGO. Ha habido dos ocasiones en las que LINGO ha obtenido un valor de FO mejor que el Mem-ACS, siendo estos las instancias 3 (0,077%) y 8 (0,039%). También se han igualado los resultados en las instancias 1, 2, 4, 5 y 7 con LINGO (11,90%).

Las instancias 1, 3, 4, 10, 13 con la heurística semiparalela son aquellas para las que se ha igualado el resultado (11,90%). Para el resto de los casos el Mem-ACS ha obtenido resultados mejores (88,10%)

Estos resultados indican que la diferencia es no significativa en términos porcentuales, pero en la comparación de valores nominales de coste, el uso de Mem-ACS supondría un gran ahorro para las industrias que realicen trabajos de producción a gran escala relacionados con la FA.

Todos los valores medios calculados indican que el Mem-ACS es capaz de obtener mejores soluciones. Siendo éste el mejor método para realizar optimización de problemas de FA considerando 3 orientaciones por pieza.

Tabla 26. Análisis GAP Mem-ACS | 3 orientaciones

	Contador GAP (LINGO)	GAP % (LINGO)	Contador GAP (Heu. Semi.)	GAP % (Heu. Semi.)
Igualado	5	11,90%	5	11,90%
Mejora	8	19,05%	37	88,10%
Empeora	2	4,76%	0	0,00%
N/A	27	64,29%	0	0,00%
GAP Promedio	-	-0,222%	-	-0,174%

Debemos resaltar que el Mem-ACS no está limitado por un número de jobs máximo ya que permite generar más jobs en el caso de que esto suponga un ahorro global de costes. Este es el caso de 11 (26,29%) instancias: 6, 11, 19, 25, 29, 36, 37 y 39 – 42.

Tabla 27. Comparación resultados | 3 orientaciones

Instancia	LINGO	GAP (LINGO)	Heurística semiparalela	GAP (Semiparalela)	Mem-ACS
1	<b>147.279,00</b>	0,000%	147.307,97	-0,020%	<b>147.279,00</b>
2	<b>233.427,47</b>	0,000%	233.674,42	-0,106%	<b>233.427,47</b>
3	<b>280.353,78</b>	0,077%	280.948,53	-0,135%	280.569,63
4	<b>345.568,37</b>	0,000%	347.324,66	-0,506%	<b>345.568,37</b>
5	<b>379.076,31</b>	0,000%	379.925,38	-0,224%	<b>379.076,31</b>
6	660.072,57	-0,449%	658.902,38	-0,273%	<b>657.106,44*</b>
7	<b>247.718,74</b>	0,000%	248.111,70	-0,158%	<b>247.718,74</b>
8	<b>325.980,16</b>	0,039%	326.695,44	-0,180%	326.106,16
9	432.149,38	-0,176%	432.348,50	-0,222%	<b>431.388,63</b>
10	584.829,76	-0,043%	584.975,50	-0,068%	<b>584.577,94</b>
11	594.041,07	-0,804%	589.781,31	-0,087%	<b>589.265,44*</b>
12	839.229,82	-1,241%	829.426,56	-0,073%	<b>828.817,69</b>
13	350.247,35	-0,140%	350.167,53	-0,117%	<b>349.757,75</b>
14	423.521,22	-0,261%	423.252,47	-0,198%	<b>422.415,91</b>
15	652.629,00	-0,336%	652.219,00	-0,273%	<b>650.437,25</b>
16	-	-	758.444,06	-0,220%	<b>756.774,88</b>
17	-	-	888.885,31	-0,195%	<b>887.153,06</b>
18	-	-	931.014,94	-0,340%	<b>927.846,44</b>

19	-	-	1.137.236,25	-0,184%	<b>1.135.141,63*</b>
20	-	-	469.125,38	-0,135%	<b>468.494,31</b>
21	-	-	664.265,88	-0,192%	<b>662.991,69</b>
22	-	-	783.579,50	-0,296%	<b>781.258,50</b>
23	-	-	1.054.310,75	-0,153%	<b>1.052.699,25</b>
24	-	-	1.208.813,25	-0,149%	<b>1.207.013,13</b>
25	-	-	2.410.051,25	-0,294%	<b>2.402.955,25*</b>
26	-	-	1.690.115,88	-0,163%	<b>1.687.366,50</b>
27	-	-	2.280.148,50	-0,173%	<b>2.276.196,00</b>
28	-	-	2.699.007,75	-0,133%	<b>2.695.413,25</b>
29	-	-	3.387.177,75	-0,137%	<b>3.382.532,75*</b>
30	-	-	3.509.790,00	-0,116%	<b>3.505.720,25</b>
31	-	-	501.380,56	-0,077%	<b>500.992,38</b>
32	-	-	883.475,06	-0,186%	<b>881.835,06</b>
33	-	-	1.222.609,38	-0,179%	<b>1.220.415,63</b>
34	-	-	1.933.370,88	-0,157%	<b>1.930.342,00</b>
35	-	-	2.598.986,25	-0,173%	<b>2.594.496,00</b>
36	-	-	3.421.157,25	-0,148%	<b>3.416.085,75*</b>
37	-	-	4.796.919,00	-0,123%	<b>4.791.030,00*</b>
38	-	-	4.467.996,50	-0,151%	<b>4.461.231,50</b>
39	-	-	5.742.071,50	-0,176%	<b>5.731.958,50*</b>
40	-	-	7.022.285,00	-0,137%	<b>7.012.697,50*</b>
41	-	-	10.112.126,0	-0,148%	<b>10.097.143,0*</b>
42	-	-	11.011.307,0	-0,125%	<b>10.997.544,0*</b>
Media	-	-0,222%	-	-0,174%	-

\*Soluciones que utilizan mayor número de jobs que los definidos en las instancias de la literatura. El algoritmo memético permite utilizar más jobs en el caso de que esto suponga una ventaja en costes.

### 5.3.4 Eficiencia del algoritmo memético

A continuación, se analiza la eficiencia del Mem-ACS. Para ello se muestra en la Tabla 29 los resultados del Mem-ACS respecto a los del ACS para los 3 tipos de problemas según orientaciones posibles de las piezas y en la Tabla 28 la diferencia porcentual. Estos valores permiten que comparar los resultados que es capaz de encontrar el ACS sin necesidad de realizar la búsqueda local posterior. Se muestran los valores obtenidos para todas las instancias en la Tabla 29. Eficiencia Mem-ACS mostrando los valores de la función objetivo, tiempo de ejecución y GAP entre Mem-ACS y ACS. La ejecución del ACS está limitada a los tiempos (minutos) obtenidos en el Mem-ACS.

Se calcula la mejora en función del valor de costes final obtenido. Los valores negativos indican que el Mem-ACS consigue unos valores menores de FO en 107 instancias suponiendo un 84,92%. Esto ocurre en la mayoría de las instancias, exceptuando algunas en las que ambos mecanismos alcanzan el mismo resultado de FO (18 instancias, 14,29%), como son para 1 orientación ( 2 – 5, 7, 10 y 20 ), para 2 orientaciones ( 1, 3, 5, 7, 10 y 11) y para 3 orientaciones ( 1, 3, 4, 10 y 13). El único caso en el que el ACS en solitario obtiene mejor solución que le Mem-ACS es en la instancia 13 para el problema de 2 orientaciones. Esta instancia no es representativa debido a que supone el 0,79% de los casos analizados.

Los datos demuestran un decremento en la cantidad de instancias con FO igualadas (1O 16,67%; 3O 11,90%) y un aumento en el número de instancias mejoradas (1O 83,33%; 3O 88,10%) en función del incremento de orientaciones de las piezas.

Tabla 28. Análisis GAP Mem-ACS vs. ACS

	Contador GAP (1O)	GAP % (1O)	Contador GAP (2O)	GAP % (2O)	Contador GAP (3O)	GAP % (3O)	Total	Total %
Igualado	7	16,67%	6	14,29%	5	11,90%	18	14,29%
Mejora	35	83,33%	35	83,33%	37	88,10%	107	84,92%
Empeora	0	0,00%	1	2,38%	0	0,00%	1	0,79%
N/A	0	0,00%	0	0,00%	0	0,00%	0	0,00%

Podemos observar en la Tabla 29. Eficiencia Mem-ACS que los valores de función objetivo alcanzados por ambos métodos son considerablemente parecidos ya que la diferencia porcentual del coste total es menor al 1% en todas las instancias y en todos los grupos de orientaciones experimentados. A pesar de ello, la aplicación de este mecanismo de optimización puede ofrecer grandes ahorros económico, ya que, a pesar de ser reducida la diferencia porcentual, en valores absolutos supone una mejora considerable (ej.: 27 1O 6.055,25 u.m.; 37 3O 15.576,00 u.m.).

A continuación, podemos observar la representación de GAP existente entre estos dos tipos de métodos de optimización para los distintos grupos de instancias agrupadas por las orientaciones disponibles de cada pieza. Se representan las instancias en el eje de abscisas respecto al indicador de dificultad calculado en la Tabla 19.

Es destacable la tendencia decreciente que corresponde a la línea de regresión para los 3 casos. Esto nos demuestra que a medida que la dimensión del problema aumenta, el Mem-ACS es capaz de ofrecer mejores resultados.

También denotamos que el aumento de magnitud a partir de la instancia 25 (100 piezas y 5 máquinas; indicador 500) marca una diferencia en el rendimiento del Mem-ACS.

En 1 orientación el GAP será inferior a -0,163% (exceptuando la instancia 37 -0,159%) llegando al -0,413% en la instancia 38. Para 2O el GAP se colocará en valores inferiores al -0,1302% alcanzando -0,3220% en la instancia 38. Y en los problemas de 3O se situarán en GAPs menores del -0,115%, consiguiendo un mínimo en la instancia 38 (-0,348%), nuevamente. Con esto diremos que la búsqueda local VND supone una gran ventaja en los problemas de dimensión superior a 500 sin importar el número de orientaciones ofrecidas, ya que en las 3 gráficas puede verse esta mejora a partir de la misma instancia.

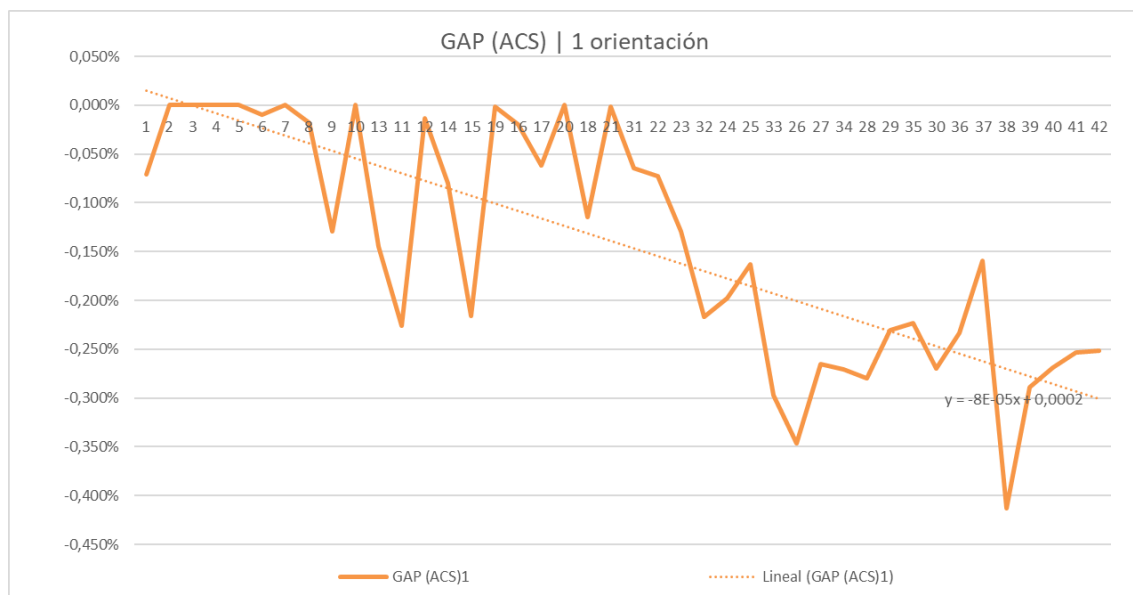


Ilustración 22. GAP Mem-ACS vs. ACS | 1 orientación

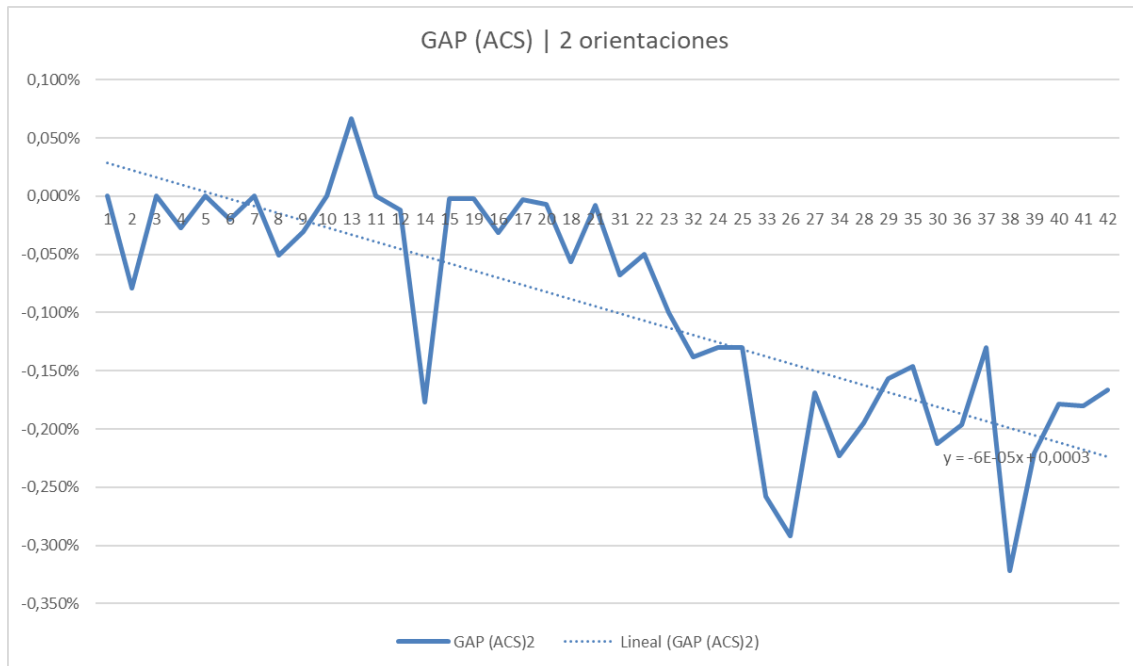


Ilustración 23. GAP Mem-ACS vs. ACS | 2 orientaciones

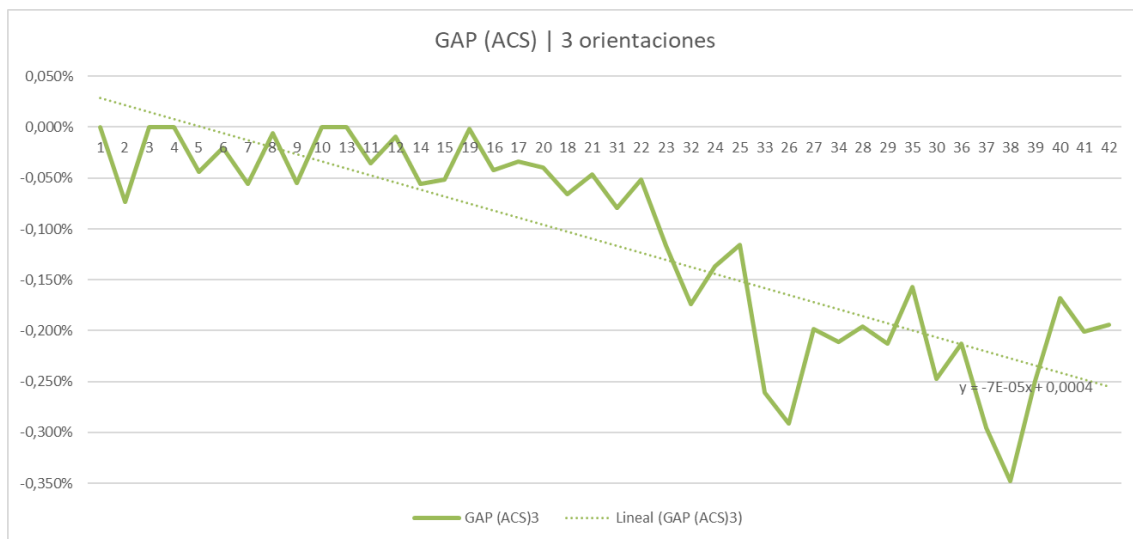


Ilustración 24. GAP Mem-ACS vs. ACS | 3 orientaciones

Tras observar los datos de GAP entre el Mem-ACS y el ACS sin VND se ha analizado la evolución de mejora de costes del Mem-ACS con el incremento de orientaciones de las piezas. El valor representativo para llevar a cabo esta comparación ha sido el de reducción de coste por unidad de volumen de los resultados. Estos valores se muestran en la Tabla 30.

Tabla 29. Eficiencia Mem-ACS

Instancia	1 Orientación				2 Orientaciones				3 Orientaciones			
	Mem-ACS	ACS	GAP	Tiempo CPU	Mem-ACS	ACS	GAP	Tiempo CPU	Mem-ACS	ACS	GAP	Tiempo CPU
1	153.574,88	153.683,70	-0,071%	0,000283	149.341,64	149.341,64	0,000%	0,001633	147.279,00	147.279,00	0,000%	0,002417
2	233.766,81	233.766,81	0,000%	0,000167	233.543,39	233.728,03	-0,0790%	0,003867	233.427,47	233.598,27	-0,073%	0,053417
3	284.267,30	284.267,30	0,000%	0,000167	282.016,94	282.016,94	0,0000%	0,004633	280.569,63	280.569,63	0,000%	0,0236
4	392.157,40	392.157,40	0,000%	0,00065	358.171,53	358.269,44	-0,0273%	0,01115	345.568,37	345.568,37	0,000%	0,015967
5	386.015,43	386.015,43	0,000%	0,000533	383.972,56	383.972,56	0,0000%	0,026833	379.076,31	379.242,63	-0,044%	0,06435
6	764.638,71	764.713,56	-0,010%	0,002183	698.117,11	698.253,63	-0,0196%	0,028383	657.106,44	657.237,94	-0,020%	0,049167
7	257.311,93	257.311,93	0,000%	0,001533	256.044,92	256.044,92	0,0000%	0,01055	247.718,74	247.856,92	-0,056%	0,030083
8	327.148,25	327.206,03	-0,018%	0,004133	326.235,36	326.400,56	-0,0508%	0,075967	326.106,16	326.126,22	-0,006%	0,393
9	463.772,93	464.374,53	-0,130%	0,005033	442.762,34	442.895,94	-0,0302%	0,039233	431.388,63	431.626,91	-0,055%	0,15945
10	618.890,60	618.890,60	0,000%	0,01565	595.188,00	595.188,00	0,0000%	0,1775	584.577,94	584.577,94	0,000%	0,7822
11	705.307,21	706.906,31	-0,226%	0,03305	641.215,63	641.215,63	0,0000%	0,094267	589.265,44	589.476,44	-0,036%	0,379417
12	851.091,56	851.207,06	-0,014%	0,042467	834.846,56	834.945,81	-0,0119%	0,40075	828.817,69	828.892,13	-0,009%	1,32635
13	362.609,61	359.762,31	0,791%	0,009267	350.854,19	350.620,66	0,0666%	0,08135	349.757,75	349.601,03	0,045%	0,202017
14	422.859,21	423.196,41	-0,080%	0,026817	422.527,19	423.276,28	-0,1770%	0,326883	422.415,91	422.651,00	-0,056%	2,600017
15	671.998,02	673.450,63	-0,216%	0,0324	663.430,00	663.442,25	-0,0018%	0,364683	650.437,25	650.771,31	-0,051%	1,161167
16	815.062,19	815.224,31	-0,020%	0,120367	794.071,00	794.317,19	-0,0310%	1,433	756.774,88	757.091,75	-0,042%	4,24575
17	1.006.640,44	1.007.265,19	-0,062%	0,171183	946.687,50	946.716,63	-0,0031%	1,3413	887.153,06	887.450,44	-0,034%	4,746883
18	1.005.556,25	1.006.709,75	-0,115%	0,166733	965.002,25	965.547,31	-0,0565%	2,404983	927.846,44	928.461,81	-0,066%	10,386133
19	1.234.983,06	1.235.003,25	-0,002%	0,025467	1.215.182,13	1.215.205,75	-0,0019%	0,293367	1.135.141,63	1.135.161,75	-0,002%	0,67465
20	486.363,75	486.363,75	0,000%	0,075317	468.983,50	469.014,69	-0,0066%	0,546517	468.494,31	468.680,06	-0,040%	2,292567
21	674.221,38	674.234,19	-0,002%	0,193433	668.166,25	668.220,25	-0,0081%	1,8119	662.991,69	663.300,00	-0,046%	6,268717
22	802.319,38	802.905,38	-0,073%	0,39155	792.898,63	793.295,63	-0,0500%	3,137683	781.258,50	781.663,19	-0,052%	12,422334
23	1.053.607,88	1.054.975,75	-0,130%	5,014117	1.052.947,00	1.054.000,25	-0,0999%	60,003132	1.052.699,25	1.053.924,88	-0,116%	60,004066
24	1.207.934,88	1.210.322,75	-0,197%	18,83185	1.207.554,75	1.209.121,63	-0,1296%	60,003567	1.207.013,13	1.208.665,13	-0,137%	60,010201
25	2.493.372,00	2.497.450,25	-0,163%	25,23975	2.439.714,00	2.442.894,75	-0,1302%	60,000816	2.402.955,25	2.405.730,75	-0,115%	60,000633

26	1.688.135,13	1.694.001,50	-0,346%	60,005402	1.687.903,38	1.692.841,00	-0,2917%	60,030182	1.687.366,50	1.692.293,38	-0,291%	60,178734
27	2.276.799,00	2.282.854,25	-0,265%	60,006451	2.276.876,75	2.280.720,50	-0,1685%	60,042233	2.276.196,00	2.280.721,00	-0,198%	60,036316
28	2.695.745,25	2.703.309,25	-0,280%	60,001465	2.695.978,00	2.701.236,75	-0,1947%	60,013966	2.695.413,25	2.700.711,75	-0,196%	60,494282
29	3.382.891,25	3.390.697,50	-0,230%	60,008568	3.383.026,75	3.388.338,25	-0,1568%	60,130733	3.382.532,75	3.389.750,00	-0,213%	60,097618
30	3.506.194,00	3.515.686,75	-0,270%	60,010368	3.506.340,75	3.513.802,25	-0,2123%	60,019516	3.505.720,25	3.514.419,75	-0,248%	60,1036
31	502.008,36	502.330,34	-0,064%	0,183917	501.371,59	501.710,03	-0,0675%	3,012367	500.992,38	501.391,72	-0,080%	17,923901
32	882.270,00	884.191,06	-0,217%	5,98945	882.120,25	883.337,63	-0,1378%	60,000568	881.835,06	883.373,06	-0,174%	60,003948
33	1.220.848,88	1.224.483,63	-0,297%	60,000198	1.220.644,88	1.223.794,63	-0,2574%	60,001301	1.220.415,63	1.223.607,38	-0,261%	60,099632
34	1.931.120,25	1.936.368,38	-0,271%	60,000965	1.930.679,63	1.934.998,75	-0,2232%	60,022518	1.930.342,00	1.934.424,25	-0,211%	60,011616
35	2.595.234,50	2.601.045,25	-0,223%	60,000465	2.595.139,00	2.598.932,75	-0,1460%	60,009083	2.594.496,00	2.598.585,25	-0,157%	60,000851
36	3.417.149,50	3.425.146,25	-0,233%	60,00745	3.416.730,50	3.423.442,75	-0,1961%	60,058182	3.416.085,75	3.423.381,25	-0,213%	60,411385
37	4.791.476,00	4.799.120,00	-0,159%	60,00005	4.791.254,00	4.797.489,00	-0,1300%	60,005032	4.791.030,00	4.805.229,50	-0,296%	60,20285
38	4.461.655,00	4.480.167,50	-0,413%	60,028084	4.461.731,50	4.476.146,50	-0,3220%	60,131466	4.461.231,50	4.476.807,50	-0,348%	62,564449
39	5.832.348,00	5.849.258,50	-0,289%	60,256851	5.774.919,00	5.787.679,00	-0,2205%	60,022316	5.731.958,50	5.746.250,50	-0,249%	60,53775
40	7.179.481,00	7.198.826,50	-0,269%	60,004051	7.082.798,50	7.095.475,50	-0,1787%	61,552952	7.012.697,50	7.024.470,50	-0,168%	60,494598
41	10.288.539,00	10.314.628,00	-0,253%	60,795166	10.176.942,00	10.195.283,00	-0,1799%	60,305733	10.097.143,00	10.117.482,00	-0,201%	60,12225
42	11.218.449,00	11.246.726,00	-0,251%	60,171249	11.099.483,00	11.117.941,00	-0,1660%	72,408798	10.997.544,00	11.018.964,00	-0,194%	81,261131
Media			-0,118%				-0,098%				-0,112%	



### 5.3.5 Efecto de la multiorientación de piezas

Los datos a continuación mostrados indican la mejora de costes por unidad de volumen del Mem-ACS a medido que se aumenta el número de orientaciones permitidas en las piezas.

De forma general el Mem-ACS obtiene mejora de costes con el aumento de orientaciones como puede verse en los valores medios de la Tabla 30. La relación entre 1 y 2 orientaciones ofrece una mejora de 0,1158 u.m./u.vol. y la relación entre 1 y 3 orientaciones 0,2095 u.m./u.vol.

Cabe destacar que el problema de 2 orientaciones no consiguió mejorar los costes por volumen del problema de 1 orientación para las instancias 27, 28, 29, 30 y 37 siendo los valores de aumento de coste -0,0002, -0,0004, -0,0002 y -0,0001, respectivamente (indicadas en naranja sobre la Tabla 30).

Tabla 30. Mejora de coste por unidad de volumen | Mem-ACS

Instancia	Volumen	1O vs 2O			1O vs 3O		
		u.m.	u.m./u.vol.	%	u.m.	u.m./u.vol.	%
1	34.151,05	4.233,24	0,1240	-0,0276	6.295,90	0,1844	-0,0410
2	51.277,84	223,47	0,0044	-0,0010	339,38	0,0066	-0,0015
3	37.716,64	2.250,36	0,0597	-0,0079	3.697,67	0,0980	-0,0130
4	52.972,41	33.985,88	0,6416	-0,0867	46.589,04	0,8795	-0,1188
5	51.753,09	2.042,87	0,0395	-0,0053	6.939,21	0,1341	-0,0180
6	97.587,83	66.521,65	0,6817	-0,0870	107.532,27	1,1019	-0,1406
7	57.286,09	1.267,60	0,0221	-0,0049	9.593,23	0,1675	-0,0373
8	71.312,58	913,47	0,0128	-0,0028	1.042,09	0,0146	-0,0032
9	57.732,10	21.010,59	0,3639	-0,0453	32.384,31	0,5609	-0,0698
10	84.383,74	23.702,60	0,2809	-0,0383	34.312,66	0,4066	-0,0554
11	95.146,95	64.091,58	0,6736	-0,0233	116.041,77	1,2196	-0,0268
12	119.275,10	16.245,00	0,1362	-0,0909	22.273,88	0,1867	-0,1645
13	81.692,75	8.386,75	0,1027	-0,0191	9.639,91	0,1180	-0,0262
14	92.724,76	332,02	0,0036	-0,0008	443,30	0,0048	-0,0010
15	98.740,47	8.568,02	0,0868	-0,0128	21.560,77	0,2184	-0,0321
16	116.572,13	20.991,19	0,1801	-0,0160	58.287,31	0,5000	-0,0808
17	144.202,89	59.952,94	0,4158	-0,0258	119.487,38	0,8286	-0,0715
18	140.852,58	40.554,00	0,2879	-0,0596	77.709,81	0,5517	-0,1187
19	164.243,07	19.800,88	0,1206	-0,0357	99.841,38	0,6079	-0,0367
20	111.738,66	17.380,25	0,1555	-0,0403	17.869,44	0,1599	-0,0773
21	149.708,41	6.055,13	0,0404	-0,0090	11.229,69	0,0750	-0,0167
22	181.401,63	9.420,75	0,0519	-0,0013	21.060,88	0,1161	-0,0020
23	230.665,28	660,88	0,0029	-0,0117	908,63	0,0039	-0,0262
24	264.204,19	380,13	0,0014	-0,0006	921,75	0,0035	-0,0009
25	323.539,32	53.658,00	0,1658	-0,0002	90.416,75	0,2795	-0,0005
26	358.871,69	231,75	0,0006	-0,0003	768,63	0,0021	-0,0008
27	497.812,86	-77,75	-0,0002	-0,0215	603,00	0,0012	-0,0363
28	590.828,26	-232,75	-0,0004	-0,0002	332,00	0,0006	-0,0004
29	741.890,72	-135,50	-0,0002	-0,0001	358,50	0,0005	-0,0005
30	768.641,54	-146,75	-0,0002	0,0000	473,75	0,0006	-0,0003

31	109.733,88	636,77	0,0058	-0,0002	1.015,98	0,0093	-0,0004
32	193.419,12	149,75	0,0008	0,0001	434,94	0,0022	-0,0001
33	267.478,42	204,00	0,0008	0,0000	433,25	0,0016	-0,0001
34	422.908,39	440,63	0,0010	-0,0000	778,25	0,0018	-0,0003
35	568.003,38	95,50	0,0002	0,0000	738,50	0,0013	-0,0001
36	749.480,96	419,00	0,0006	-0,0001	1.063,75	0,0014	-0,0003
37	1.051.179,10	222,00	0,0002	-0,0000	446,00	0,0004	-0,0001
38	978.095,92	-76,50	-0,0001	0,0000	423,50	0,0004	-0,0001
39	1.317.484,30	57.429,00	0,0436	-0,0098	100.389,50	0,0762	-0,0172
40	1.618.653,10	96.682,50	0,0597	-0,0135	166.783,50	0,1030	-0,0232
41	2.317.876,70	111.597,00	0,0481	-0,0108	191.396,00	0,0826	-0,0186
42	2.528.683,50	118.966,00	0,0470	-0,0106	220.905,00	0,0874	-0,0197
Media			0,1158	-0,0172		0,2095	-0,0309

Se muestran las gráficas de los datos anteriores para facilitar la comprensión y completar el análisis mostrando las instancias en el eje de abscisas de manera incremental en función de su dimensión.

Primero se ha analizado la representación del coste por unidad de volumen mediante la Ilustración 25. Es interesante observar la oscilación de esta mejora desde la instancia 1 a la 33, presentando un máximo en la instancia 11 con valor 1,2196 u.m./u.vol. respecto a la relación de 1 y 3 orientaciones. Entre las instancias 33 y 38 se observa un valle en la gráfica en el que los costes unitarios se sitúan entre los valores -0,0004 u.m./u.vol. (relación 1 y 2 orientaciones, instancia 28, -232,75 u.m.) y 0,0021 u.m./u.vol. (relación 1 y 3 orientaciones, instancia 26, 768,63 u.m.).

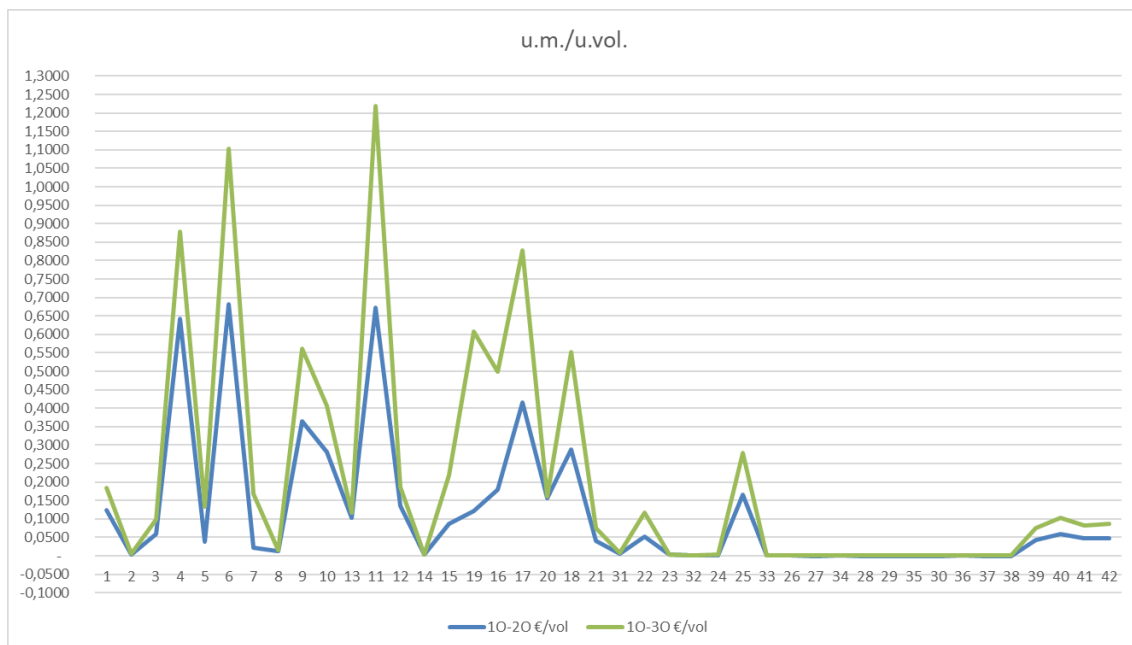


Ilustración 25. Representación u.m./u.vol. Mem-ACS

La Ilustración 26 ofrece otra perspectiva de mejora de costes que no considera el volumen de las piezas. Muestra los costes totales de mejora de cada instancia en función de la dimensión del problema. En esta gráfica también puede apreciarse la oscilación de mejora de costes desde la instancia 1 a la 33 y el valle entre las instancias 33 y 38. La experimentación con problemas de mayor dificultad se hace palpable al observar estos resultados que parecen intuir un comportamiento incremental a partir de problemas de dificultad 1800.

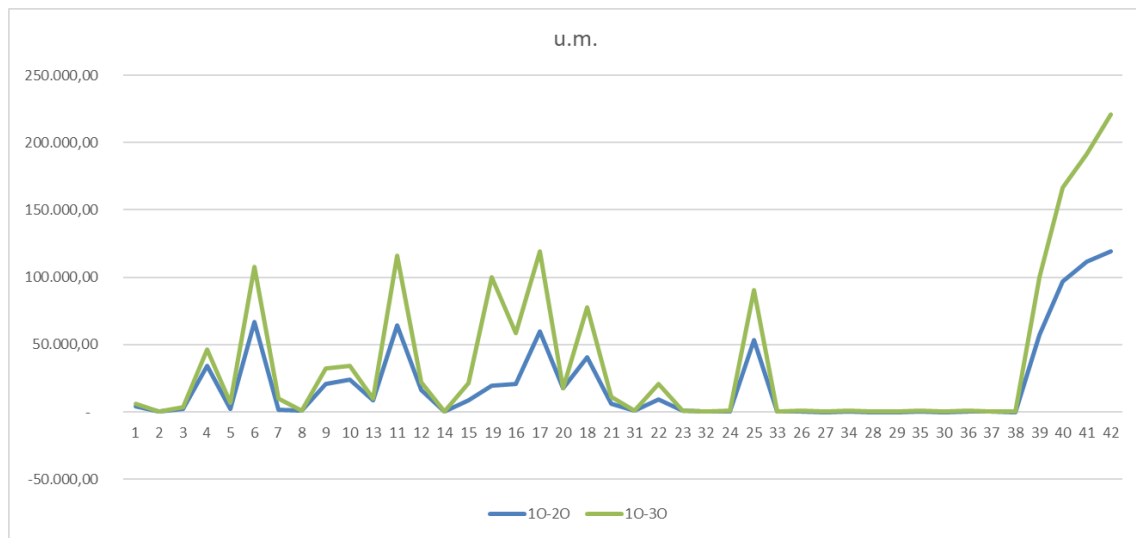


Ilustración 26. Representación u.m. Mem-ACS

A pesar de la tendencia decreciente de la evolución de costes por unidad de volumen de la Ilustración 25 se demuestra el aumento en el ahorro de costes mediante la Ilustración 26 para problemas que van aumentando su dimensión.

# 6 CONCLUSIONES

---

Este trabajo ha desarrollado el análisis de un algoritmo memético para la optimización en la planificación de la producción haciendo uso de máquinas de FA. Está basado en dos metaheurísticas muy extendidas como son el algoritmo de ACS y el VND. Este algoritmo no había sido utilizado en el campo bajo estudio anteriormente en la literatura y supone una contribución científica respecto a los procesos de optimización de costes en FA.

El objetivo principal del trabajo radica en la mejora de costes de los mecanismos de planificación de la producción basados en el problema de “nesting” para piezas con varias orientaciones. La ampliación de posibles orientaciones sobre problemas encontrados en la literatura aporta una complejidad añadida a las instancias y una aproximación real a los métodos utilizados en la industria. El algoritmo utilizado ha sido adecuado a las componentes específicas del problema como las características de las máquinas y las distintas orientaciones aportadas para cada pieza.

El algoritmo memético necesita de una solución inicial ofrecida por una heurística de generación de estructuras semiparalela inspirada en artículos de la literatura. Los parámetros principales de esta heurística han sido obtenidos mediante una experimentación previa para reducir los tiempos computacionales del Mem-ACS.

También, la comparación del Mem-ACS con otros métodos de optimización ha aportado la visión sobre la ventaja de su uso en este tipo de problemas. El análisis ha sido realizado por separado para cada conjunto de problemas en función de las orientaciones permitidas (1, 2 y 3). Se concluye ofreciendo el algoritmo Mem-ACS como el mejor método a la hora de evaluar problemas complejos de FA considerando orientaciones en las piezas. Además, el número de orientaciones permitidas influye directamente en la complejidad del problema y en la conveniencia de utilizar el Mem-ACS para su resolución ya que se consiguen ahorros de costes en un 84,92% de las instancias.

## 6.1 Líneas futuras de investigación

El presente estudio ha realizado la optimización del problema en términos de coste de producción. La componente de secuenciación es otro tema de gran importancia y desarrollado en otros trabajos de la literatura que no ha sido estudiado en este trabajo. Combinar ambas componentes aportaría una visión más completa del proceso de programación de la producción.

# 7 ANEXOS

## 7.1 Ajuste de parámetros alfa

Instancia	Orientaciones	Alfa 1 Min	Alfa 1 Max	Alfa 2 Min	Alfa 2 Max	Alfa 3 Min	Alfa 3 Max	Alfa 4 Min	Alfa 4 Max	Valor alcanzado
26	1	6	8	70	78	10	14	0	14	1.692.397,13
27	1	6	7	64	66	19	23	4	11	2.282.432,50
28	1	11	12	69	71	16	17	0	4	2.701.303,50
29	1	2	5	57	65	13	30	0	28	3.388.253,75
30	1	9	9	65	65	22	22	4	4	3.510.577,50
38	1	3	6	61	73	8	16	5	28	4.471.341,50
39	1	7	7	72	72	14	14	7	7	5.846.757,50
40	1	6	13	55	61	8	18	8	31	7.197.391,50
41	1	8	8	53	53	30	30	9	9	10.314.315,00
42	1	8	8	72	72	20	20	0	0	11.245.334,00
26	2	15	15	63	63	8	8	14	14	1.692.005,88
27	2	7	7	77	77	5	5	11	11	2.282.221,25
28	2	6	17	58	73	3	9	1	33	2.700.574,75
29	2	12	12	73	73	8	8	7	7	3.389.147,75
30	2	10	16	65	74	6	10	0	19	3.513.581,75
38	2	19	19	70	70	9	9	2	2	4.472.456,00
39	2	5	5	56	56	3	3	36	36	5.786.697,00
40	2	12	12	78	78	8	8	2	2	7.093.875,00
41	2	2	2	56	56	1	1	41	41	10.191.406,00
42	2	8	8	70	70	5	5	17	17	11.111.496,00
26	3	16	16	64	64	15	15	5	5	1.690.115,88
27	3	5	9	59	68	13	23	1	23	2.280.148,50
28	3	7	10	63	68	4	6	16	26	2.699.007,75
29	3	19	24	58	60	10	13	3	13	3.387.177,75
30	3	8	8	52	52	23	23	17	17	3.509.790,00
38	3	20	20	63	63	15	15	2	2	4.467.996,50
39	3	6	6	65	65	4	4	25	25	5.742.071,50
40	3	6	6	81	81	4	4	9	9	7.022.285,00
41	3	11	11	80	80	7	7	2	2	10.112.126,00
42	3	10	10	76	76	7	7	7	7	11.011.307,00

## 7.2 Ajuste de parámetros primera fase (q0 y beta)

q0	Beta	14_3O	15_2O	16_3O	17_2O
75	0,5	422415,87500	663430,00000	756774,87500	946687,50000
75	0,5	422407,37500	663430,00000	756774,87500	946687,50000
75	0,5	422415,87500	663430,00000	756774,87500	946687,50000
75	0,5	422415,87500	663430,00000	756774,87500	946687,50000
75	0,5	422375,46875	663430,00000	756774,87500	946687,56250
75	0,7	422415,87500	663430,00000	756774,87500	946687,50000
75	0,7	422415,87500	663430,00000	756774,87500	946687,62500
75	0,7	422415,87500	663430,00000	756774,87500	946687,50000
75	0,7	422366,50000	663430,00000	756784,93750	946687,56250
75	0,7	422415,87500	663430,00000	756774,87500	946687,50000
75	1,0	422415,87500	663430,00000	756774,87500	946687,56250
75	1,0	422415,87500	663430,00000	756774,87500	946687,56250
75	1,0	422366,50000	663430,00000	756774,87500	946687,50000
75	1,0	422392,84375	663430,00000	756784,93750	946687,50000
75	1,0	422375,43750	663430,00000	756774,87500	946707,25000
75	1,5	422405,68750	663430,00000	756774,87500	946687,62500
75	1,5	422415,87500	663430,00000	756784,93750	946687,50000
75	1,5	422415,87500	663430,00000	756785,00000	946687,56250
75	1,5	422366,50000	663430,00000	756784,93750	946723,50000
75	1,5	422408,46875	663430,00000	756774,87500	946687,56250
75	2,0	422415,87500	663430,00000	756774,87500	946707,25000
75	2,0	422415,87500	663430,00000	756774,87500	946687,62500
75	2,0	422415,87500	663430,00000	756774,87500	946687,50000
75	2,0	422392,84375	663430,00000	756774,87500	946687,50000
75	2,0	422415,87500	663430,00000	756774,87500	946687,50000
85	0,5	422415,90625	663430,00000	756806,50000	946687,50000
85	0,5	422415,87500	663430,00000	756774,87500	946687,50000
85	0,5	422415,87500	663430,00000	756774,87500	946687,50000
85	0,5	422366,50000	663430,00000	756774,81250	946687,56250
85	0,5	422415,87500	663430,00000	756806,50000	946687,50000
85	0,7	422371,00000	663430,00000	756774,87500	946687,50000
85	0,7	422415,87500	663430,00000	756806,50000	946707,25000
85	0,7	422415,87500	663430,00000	756774,87500	946687,50000
85	0,7	422415,87500	663430,00000	756774,87500	946687,50000
85	0,7	422370,96875	663430,00000	756806,50000	946687,50000
85	1,0	422397,31250	663430,00000	756774,87500	946688,56250
85	1,0	422415,87500	663430,00000	756806,43750	946687,50000
85	1,0	422415,87500	663430,00000	756784,93750	946687,50000
85	1,0	422415,87500	663430,00000	756774,87500	946687,50000
85	1,0	422402,93750	663430,00000	756774,87500	946687,56250
85	1,5	422413,53125	663430,00000	756806,50000	946687,50000

85	1,5	422405,68750	663430,00000	756806,50000	946687,50000
85	1,5	422413,53125	663430,00000	756806,50000	946687,50000
85	1,5	422415,87500	663430,00000	756774,87500	946687,50000
85	1,5	422404,56250	663430,00000	756774,87500	946687,50000
85	2,0	422415,87500	663430,00000	756806,43750	946687,50000
85	2,0	422366,50000	663430,00000	756774,87500	946672,75000
85	2,0	422415,90625	663430,00000	756774,87500	946687,50000
85	2,0	422415,87500	663430,00000	756784,93750	946687,56250
85	2,0	422404,56250	663430,00000	756774,81250	946687,50000
Media		422366,50000	663430,00000	756774,81250	946672,75000

### 7.3 Ajuste de parámetros segunda fase (N y M)

N	M	14_30	15_20	16_30	17_20
10	10	422415,87500	663430,00000	756774,87500	946687,56250
10	10	422415,87500	663430,00000	756774,87500	946723,43750
10	10	422415,87500	663430,00000	756822,81250	946687,50000
10	10	422415,90625	663430,00000	756808,93750	946687,56250
10	10	422366,46875	663430,00000	756822,87500	946687,62500
10	20	422415,87500	663430,00000	756785,00000	946711,12500
10	20	422415,90625	663430,00000	756785,00000	946707,25000
10	20	422415,87500	663430,00000	756785,00000	946687,56250
10	20	422415,87500	663430,00000	756785,00000	946687,50000
10	20	422415,87500	663430,00000	756774,93750	946687,50000
10	30	422415,87500	663430,00000	756815,31250	946687,56250
10	30	422415,87500	663430,00000	756784,93750	946687,50000
10	30	422404,56250	663430,00000	756785,00000	946711,12500
10	30	422415,87500	663430,00000	756774,87500	946687,50000
10	30	422415,87500	663430,00000	756774,87500	946687,50000
20	10	422415,90625	663430,00000	756774,87500	946687,43750
20	10	422366,50000	663430,00000	756774,87500	946687,50000
20	10	422415,87500	663430,00000	756784,93750	946687,50000
20	10	422415,87500	663430,00000	756806,50000	946687,50000
20	10	422415,87500	663430,00000	756774,87500	946687,56250
20	20	422413,53125	663430,00000	756806,50000	946687,50000
20	20	422405,68750	663430,00000	756806,50000	946687,50000
20	20	422413,53125	663430,00000	756806,50000	946687,50000
20	20	422415,87500	663430,00000	756774,87500	946687,50000
20	20	422404,56250	663430,00000	756774,87500	946687,50000
20	30	422370,96875	663430,00000	756774,87500	946687,50000
20	30	422415,90625	663430,00000	756784,93750	946687,50000
20	30	422370,93750	663430,00000	756774,87500	946687,50000
20	30	422415,87500	663430,00000	756774,87500	946687,50000

20	30	422404,56250	663430,00000	756774,87500	946687,50000
30	10	422415,87500	663430,00000	756806,43750	946687,50000
30	10	422415,87500	663430,00000	756806,50000	946687,50000
30	10	422415,87500	663430,00000	756784,93750	946687,50000
30	10	422415,87500	663430,00000	756774,87500	946687,50000
30	10	422415,87500	663430,00000	756806,43750	946687,50000
30	20	422366,50000	663430,00000	756774,87500	946687,50000
30	20	422409,03125	663430,00000	756774,87500	946687,50000
30	20	422415,87500	663430,00000	756784,93750	946687,50000
30	20	422415,87500	663430,00000	756774,87500	946687,56250
30	20	422366,50000	663430,00000	756785,00000	946687,50000
30	30	422366,50000	663430,00000	756774,87500	946687,50000
30	30	422366,50000	663430,00000	756774,87500	946687,50000
30	30	422366,50000	663430,00000	756784,93750	946687,50000
30	30	422415,87500	663430,00000	756774,87500	946687,50000
30	30	422366,50000	663430,00000	756784,93750	946687,50000
Media		422366,46875	663430,00000	756774,87500	946687,43750

## 7.4 Código de generación de instancias

Este fichero .py sirve para convertir los datos de entrada Excel al formato .amo necesario para la ejecución del código Visual Studio.

El dato de número máximo de jobs debe extraerse a mano del artículo: Li et al., (2017)

Instrucciones de uso del fichero:

El fichero .py debe estar en el mismo directorio que los siguientes ficheros:

- IOfunctionsExcel2.py
- Los ficheros Excel con los datos de entrada. Tendrán un formato para el título P##.

```
import IOfunctionsExcel2 as ex
```

```
import openpyxl
```

```
o = 3 ### Escribir 1, 2 o 3 para leer el tipo de archivo adecuado.
```

```
r= 3 #redondeo
```

```
njobmax=[6,5,8,11,11,10,10,7,10,12,14,20,12,10,15,18,23,23,16,16,20,21,19,27,49,33,42,49,57,60,10,17,24,35,47,63,74,84,170,190,274,305]
```

```
problemas = {}
```

```
for t in range(1,43): ##Seleccionar el rango de ficheros para lo que trabajar
```

```
    datos_parts={}
```

```
    datos_machines={}
```

```
    try:
```



```

excel_document = openpyxl.load_workbook('P%s.xlsx'%(t), read_only=True)
datos_parts.update( ex.Read_Excel_to_NesteDic(excel_document['PartsData'],'a1','h1000'))
datos_machines.update( ex.Read_Excel_to_NesteDic(excel_document['MachinesData'],'a1','i1000'))
excel_document.close()
n='P%s'%(t)
problemas.update({n: {'parts': datos_parts}})
problemas[n].update({'machines': datos_machines})
f= open('Set_P%s_%sO.amo'%(t,o), 'w')

f.write('Número parts:\n%s\n'%(len(problemas[n]['parts'])))
f.write('Número jobs máximo:\n%s\n'%(njobmax[t-1]))
f.write('Número máquinas:\n%s\n'%(len(problemas[n]['machines'])))

f.write('*****\n')
f.write('Part\t# orient volumen\n')
count=1
for keys, values in problemas[n]['parts'].items():
    f.write ('%\s\t%\s\t %s\n'%(count,o,round(values['Volume (cm3)'],r)))
    count+=1

f.write('*****\n')
f.write('part.\torient\tAltura\tSup\n')
count=1
for keys, values in problemas[n]['parts'].items():
    f.write ('%\s\t1\t%\s\t%\s\n'%(count,values['Height (cm)'],values['Area (cm2)']))
    if o >= 2:
        f.write ('%\s\t2\t%\s\t%\s\n'%(count,values['2Height (cm)'],values['2Area (cm2)']))
    if o >= 3:
        f.write ('%\s\t3\t%\s\t%\s\n'%(count,values['3Height (cm)'],values['3Area (cm2)']))
    count+=1

f.write('*****\n')
f.write('mach.\tTC\t(1/VT)\tMC\tHT\tSET\tHC\tMA\tMH\n')
count=1
for keys, values in problemas[n]['machines'].items():
    f.write
('%\s\t%\s\t%\s\t%\s\t%\s\t%\s\t%\s\t%\s\n'%(count,values['TCm(GBP/hour)'],1/values['VTm(hour/cm3)'],valu
es['MCm(GBP/cm3)'],values['HTm(hour/cm)'],values['STm(hour)'],values['HCm(GBP/hour)'],values['Am(cm
2)'],values['Hm(cm)']))
    count+=1

```

```
f.write('*****\n')
    f.close()
    print('ejecutado%s'%(t))
except:
    print('fail %s'%(t))
print('FIN')
```

## 7.5 Código de extracción de alfas para el ajuste de parámetros

Este fichero .py se utiliza para extraer los valores de alpha de los ficheros .amo con las soluciones y los guarda en un Excel.

Instrucciones:

- Los ficheros a leer .amo y el .py deben estar en el mismo directorio.
- Los ficheros .amo deben tener el nombre estándar definido.
- Seleccionar las orientaciones para las que estamos haciendo la lectura de datos. Variable "o".
- Se generará un Excel con los datos de alpha para los problemas analizados.

```
import pandas as pd

## Seleccion de orientaciones
o = 3
datos={}
a=[26,27,28,29,30]
b=[38,39,40,41,42]
p=a+b
writer = pd.ExcelWriter('Alfas comparacion %sO.xlsx'%(o), engine='xlsxwriter')
for i in p:
    try:
        strucxl={}
        file1 = open("Set_P%s_%sO.amo_soluciones_heuristica_alfas.sol"%(i,o),"r")
        print("Alfas de P%s para %s orientaciones"%(i,o))
        count=0
        strucxl={}
        for line in file1:
            try:
                strucxl[count]={}
                a=line.split()
                datos[count]=a
```

```

    strucxl[count]['Alfa 1']=datos[count][2]
    strucxl[count]['Alfa 2']=datos[count][5]
    strucxl[count]['Alfa 3']=datos[count][8]
    strucxl[count]['Alfa 4']=datos[count][11]
    strucxl[count]['Resultado']=datos[count][13]
    count+=1
except:
    print('fin fichero P%s'%(i))
    pass
file1.close()
df = pd.DataFrame.from_dict(strucxl,orient='index') # convert dict to dataframe
df.to_excel(writer, sheet_name='Alfas P%s'%(i)) # write dataframe to file
except:
    print('fail %s'%(i))
    pass
writer.close()
print('Fin ejecución')

```

## 7.6 Código de extracción de costes totales

Este fichero .py se utiliza para extraer el valor de coste total de los ficheros .amo con las soluciones y los guarda en un Excel para su posterior copia al Excel de comparación de resultados.

Instrucciones:

- Los ficheros a leer .amo y el .py deben estar en el mismo directorio.
- Los ficheros .amo tienen que tener el nombre estándar definido.
- Seleccionar las orientaciones para las que estamos haciendo la lectura de datos. Variable "o".
- Se generará un Excel con los datos de coste total de los problemas analizados.

```

import openpyxl
import pandas as pd

## Seleccion de orientaciones
o = 3
costtotal={}
for i in range(1,43):
    costtotal[i]={}
    try:
        file1 = open("Set_P%s_%sO.amo_solucion_ACS_unido_construccion.sol"%(i,o),"r")
        print("Problema %s para %s orientaciones"%(i,o))

```

```
count=1
for line in file1:
    if count == 2:
        c=line.split()
        d=c[8]
        d=float(d)
        print(d)
        costtotal[i]['njobmax']=d
    if count == 3:
        a=line.split()
        b=a[2]
        b=float(b)
        print(b)
        costtotal[i]['coste']=b
    if count == 4:
        a=line.split()
        b=a[3]
        b=float(b)
        print(b)
        costtotal[i]['CPU']=b
    count+=1
file1.close()
except:
    print('fallo')
    pass
writer = pd.ExcelWriter('Costes totales %sO_MemACS.xlsx'%(o), engine='xlsxwriter')
df = pd.DataFrame.from_dict(costtotal,orient='index') # convert dict to dataframe
df.to_excel(writer, sheet_name='Costes') # write dataframe to file
writer.close()
```

# REFERENCIAS

- 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 and Operations Research*, *131*. <https://doi.org/10.1016/j.cor.2021.105272>
- Che, Y., Hu, K., Zhang, Z., & Lim, A. (2021). Machine scheduling with orientation selection and two-dimensional packing for additive manufacturing. *Computers & Operations Research*, *130*, 105245. <https://doi.org/10.1016/J.COR.2021.105245>
- Chergui, A., Hadj-Hamou, K., & Vignat, F. (2018). Production scheduling and nesting in additive manufacturing. *Computers & Industrial Engineering*, *126*, 292–301. <https://doi.org/10.1016/J.CIE.2018.09.048>
- Costabile, G., Fera, M., Fruggiero, F., Lambiase, A., & Pham, D. (2016). Cost models of additive manufacturing: A literature review. *International Journal of Industrial Engineering Computations*, *8*(2). <https://doi.org/10.5267/j.ijiec.2016.9.001>
- Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1). <https://doi.org/10.1109/4235.585892>
- Fera, M., Fruggiero, F., Lambiase, A., & Macchiaroli, R. (2016). State of the art of additive manufacturing: Review for tolerances, mechanical resistance and production costs. *Cogent Engineering*, *3*(1). <https://doi.org/10.1080/23311916.2016.1261503>
- Fera, M., Fruggiero, F., Lambiase, A., Macchiaroli, R., & Todisco, V. (2018). A modified genetic algorithm for time and cost optimization of an additive manufacturing single-machine scheduling. *International Journal of Industrial Engineering Computations*, *9*(4). <https://doi.org/10.5267/j.ijiec.2018.1.001>
- Fera, M., Macchiaroli, R., Fruggiero, F., & Lambiase, A. (2018). A new perspective for production process analysis using additive manufacturing—complexity vs production volume. *International Journal of Advanced Manufacturing Technology*, *95*(1–4). <https://doi.org/10.1007/s00170-017-1221-1>
- Fera, M., Macchiaroli, R., Fruggiero, F., & Lambiase, A. (2020). A modified tabu search algorithm for the single-machine scheduling problem using additive manufacturing technology. *International Journal of Industrial Engineering Computations*, *11*(3). <https://doi.org/10.5267/j.ijiec.2020.1.001>
- Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, *175*(1). <https://doi.org/10.1007/s10479-009-0657-6>
- Jia, Z. hong, Wang, Y., Wu, C., Yang, Y., Zhang, X. yi, & Chen, H. ping. (2019). Multi-objective energy-aware batch scheduling using ant colony optimization algorithm. *Computers and Industrial Engineering*, *131*. <https://doi.org/10.1016/j.cie.2019.03.033>
- Kucukkoc, I. (2019). MILP models to minimise makespan in additive manufacturing machine scheduling problems. *Computers and Operations Research*, *105*. <https://doi.org/10.1016/j.cor.2019.01.006>
- Li, Q., Kucukkoc, I., & Zhang, D. Z. (2017). Production planning in additive manufacturing and 3D printing. *Computers and Operations Research*, *83*. <https://doi.org/10.1016/j.cor.2017.01.013>
- Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and Its Applications*, *391*(6). <https://doi.org/10.1016/j.physa.2011.12.004>
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, *24*(11). [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)
- Molina, J. C., Salmeron, J. L., & Eguia, I. (2020). An ACS-based memetic algorithm for the heterogeneous vehicle routing problem with time windows. *Expert Systems with Applications*, *157*. <https://doi.org/10.1016/j.eswa.2020.113379>

- Molina, J. C., Salmeron, J. L., Eguia, I., & Racero, J. (2020). The heterogeneous vehicle routing problem with time windows and a limited number of resources. *Engineering Applications of Artificial Intelligence*, 94. <https://doi.org/10.1016/j.engappai.2020.103745>
- Murr, L. E., & Johnson, W. L. (2017). 3D metal droplet printing development and advanced materials additive manufacturing. *Journal of Materials Research and Technology*, 6(1), 77–89. <https://doi.org/10.1016/J.JMRT.2016.11.002>
- Oh, Y., Witherell, P., Lu, Y., & Sprock, T. (2020). Nesting and scheduling problems for additive manufacturing: A taxonomy and review. In *Additive Manufacturing* (Vol. 36). <https://doi.org/10.1016/j.addma.2020.101492>
- Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14(5). <https://doi.org/10.1007/s10732-007-9045-z>
- Ransikarbum, K., Ha, S., Ma, J., & Kim, N. (2017). Multi-objective optimization analysis for part-to-Printer assignment in a network of 3D fused deposition modeling. *Journal of Manufacturing Systems*, 43. <https://doi.org/10.1016/j.jmsy.2017.02.012>
- Sreerag, C., Gajjela, R., Manupati, V., & Machado, J. (2021). Additive manufacturing: a thrive for industries. *Additive Manufacturing*, 1–16. <https://doi.org/10.1016/B978-0-12-822056-6.00011-4>
- Wohlers, T., & Gornet, T. (2014). History of Additive Manufacturing 2014. *Wohlers Report 2014 - 3D Printing and Additive Manufacturing State of the Industry*.
- Zhang, J., Yao, X., & Li, Y. (2020). Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *International Journal of Production Research*, 58(8). <https://doi.org/10.1080/00207543.2019.1617447>
- Zhang, X., & Liou, F. (2021). Introduction to additive manufacturing. *Additive Manufacturing*, 1–31. <https://doi.org/10.1016/B978-0-12-818411-0.00009-4>
- Zhang, Y., Bernard, A., Harik, R., & Karunakaran, K. P. (2017). Build orientation optimization for multi-part production in additive manufacturing. *Journal of Intelligent Manufacturing*, 28(6). <https://doi.org/10.1007/s10845-015-1057-1>
- Zhang, Y., Gupta, R. K., & Bernard, A. (2016). Two-dimensional placement optimization for multi-parts production in additive manufacturing. *Robotics and Computer-Integrated Manufacturing*, 38. <https://doi.org/10.1016/j.rcim.2015.11.003>

