

**ANÁLISIS DEL DESEMPEÑO DE DOS METODOLOGÍAS META HEURÍSTICAS
EN EL PROBLEMA DE FORMACIÓN DE CELDAS DE MANUFACTURA**

CINDY JOHANNA RODRÍGUEZ LEÓN

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA**

2011

**ANÁLISIS DEL DESEMPEÑO DE DOS METODOLOGÍAS META HEURÍSTICAS
EN EL PROBLEMA DE FORMACIÓN DE CELDAS DE MANUFACTURA**

CINDY JOHANNA RODRÍGUEZ LEÓN

Proyecto de grado para optar por el título de Ingeniera Industrial

Director

JABID EDUARDO QUIROGA MÉNDEZ

Ingeniero Mecánico

Codirector

NÉSTOR RAÚL ORTIZ PIMIENTO

Ingeniero Industrial

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA**

2011

DEDICATORIA

A mi Dios que todo le debo

A mis padres, por toda la paciencia, los cuidados, las enseñanzas, las preocupaciones y sobre todo por todo el amor que no se cansan de brindarme; porque por ellos estoy en donde estoy y soy la buena persona que la gente ve en mí; por estar siempre ahí, por ser incondicionales y por ser las personas que más amo y valoro en el mundo.

A mis dos hermanitos, Kevin y Evelyn, por llenar el hogar de alegría y risas, por ser los niños más hermosos y ocurrentes que conozco y porque sin ellos la vida no sería la misma. A mi hermana mayor Laura, porque todos sus logros y su empuje me han inspirado y me han alentado a culminar con éxito los proyectos que me propongo; porque la veo como un ejemplo a seguir. Los amo a los tres con todo mi corazón.

AGRADECIMIENTOS

El autor expresa sus más sinceros agradecimientos a:

Al profe Jabid porque hizo de mi grado una realidad al permitirme desarrollar la idea de este proyecto, concebida por él. Por toda su colaboración y su disponibilidad en todo momento en el que necesité de su ayuda y sus conocimientos.

Al profesor Néstor Raúl por estar pendiente de los avances del trabajo y de hacer las correcciones necesarias, por guiarme en la elaboración del documento, por su disponibilidad de tiempo para realizar las reuniones y por la amabilidad que siempre mostró en cada una de las mismas.

A Guillermo Celis, por ayudarme a conseguir el instalador de Matlab, fundamental en el desarrollo de la presente investigación. Por ser un amigo incondicional y estar dispuesto a ayudarme siempre que lo necesité.

A Rafa Ramírez y Jaime Prieto por ayudarme a comprender el lenguaje de programación Matlab, porque aún cuando tenían sus propias obligaciones tuvieron la disponibilidad de ayudarme siempre que lo sentí necesario. Porque en ese momento fue que arrancó mi proyecto de grado. Muchas gracias por su amabilidad y por las enseñanzas y conocimientos compartidos.

A Henry Lamos, Edwin Garavito, Myriam Niño y todos los profesores que durante mi transcurso por la Universidad me enseñaron todo lo que ahora sé, por compartir sus conocimientos y guiarme con tanta paciencia, no solo a nivel académico, sino también a nivel personal, ético y profesional a través de sus experiencias compartidas.

TABLA DE CONTENIDO

INTRODUCCIÓN	17
TABLA DE CUMPLIMIENTO DE OBJETIVOS	19
1. GENERALIDADES DEL TRABAJO DE GRADO.....	20
1.1. DESCRIPCIÓN DEL TEMA DE INVESTIGACIÓN	20
1.2. OBJETIVOS DEL TRABAJO DE GRADO	23
1.3. ALCANCE DEL TRABAJO DE GRADO	24
1.4. INDICACIONES GENERALES SOBRE EL DOCUMENTO	24
2. MARCO TEÓRICO CONCEPTUAL	27
2.1. TECNOLOGÍA DE GRUPOS.....	27
2.2. MANUFACTURA CELULAR	29
2.3. METODOLOGÍAS PARA LA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA	33
2.4. ALGORITMO PSO	48
3.1. SUPUESTOS	59
3.2. DATOS DE ENTRADA.....	60
3.3. DECISIONES DE DISEÑO.....	61
3.4. OBJETIVOS DE DISEÑO.....	61
3.5. RESTRICCIONES	61
3.6. NOTACIONES.....	62
3.7. FUNCIÓN OBJETIVO	63
4. PROGRAMACIÓN DEL ALGORITMO PSO PARA EL DISEÑO DE SISTEMAS DE MANUFACTURA CELULAR.....	65
4.1. VARIABLES Y PARÁMETROS DE ENTRADA	65

4.2. SALIDAS	67
4.3. FUNCIONES DE PROGRAMACIÓN DEL ALGORITMO PSO.....	68
5. AJUSTE DE PARÁMETROS DEL ALGORITMO PSO.....	76
5.1. COEFICIENTE DE ACELERACIÓN C1	77
5.2. COEFICIENTE DE ACELERACIÓN C2	87
5.3. FACTOR DE INERCIA W	92
5.4. TAMAÑO DE LA POBLACIÓN.....	97
5.5. TAMAÑO DE POBLACIÓN CON BASE EN EL NÚMERO DE ITERACION	102
6. RESULTADOS DETALLADOS DEL ALGORITMO PSO AJUSTADO.....	108
7.1. RESULTADOS EN TÉRMINOS DEL COSTO.....	114
7.2. RESULTADOS EN TÉRMINOS DEL TIEMPO DE EJECUCIÓN	116
7.3. RESULTADOS DETALLADOS POR PROBLEMA	117
CONCLUSIONES	122
RECOMENDACIONES.....	124
BIBLIOGRAFÍA.....	125

LISTA DE FIGURAS

Figura 1: Componentes de una partícula PSO	50
Figura 2: Pseudo-código PSO clásico	52
Figura 3: Función principal Toolbox PSOt	56
Figura 4: Configuración de cada partícula	66
Figura 5: Diagrama de flujo algoritmo PSO	75
Figura 6: Gráfico de medias para la variable COSTO del factor C1	81
Figura 7: Gráfico de medias variable TIEMPO Factor C1	82
Figura 8: Gráfica de probabilidad normal de residuos factor C1	84
Figura 9: Gráfica de dispersión de valores predichos contra residuos del factor C1	85
Figura 10: Gráfica de tiempo vs residuos para el factor C1	86
Figura 11: Gráfico de medias variable COSTO Factor C2	90
Figura 12: Gráfico de medias variable TIEMPO Factor C2	91
Figura 13: Gráfico de medias variable COSTO Factor peso W	94
Figura 14: Gráfico de medias variable TIEMPO Factor peso W	95
Figura 15: Gráfico de medias para la variable COSTO factor tamaño de población	99
Figura 16: Gráfico de medias variable TIEMPO factor Tamaño de población	100
Figura 17: Gráfico de medias variable COSTO para tamaño de población con iteraciones	105
Figura 18: Gráfico de medias variable TIEMPO Tamaño de población con iteraciones	106

Figura 19: Valor fitness por iteración por réplica.....	109
Figura 20: Mejores resultados en cuanto al costo por metodología.....	115
Figura 21: Tiempos de ejecución empleados por metodología por problema.....	116
Figura 22: Costo por réplica por iteración, problema 1, algoritmo PSO	119
Figura 23: Costo por réplica por generación, problema 1, Algoritmo genético	119

LISTA DE TABLAS

Tabla 1: Clasificación de las metodologías Meta-heurísticas	39
Tabla 2: Niveles del parámetro C1	78
Tabla 3: Resultados consolidados del diseño experimental aplicado al factor Coeficiente C1	79
Tabla 4: Análisis de Varianza variable COSTO Factor C1	80
Tabla 5: Análisis de Varianza variable TIEMPO Factor C1	81
Tabla 6: Niveles del factor C2	88
Tabla 7: Resultados consolidados del diseño experimental aplicado al factor Coeficiente C2	88
Tabla 8: Análisis de Varianza variable COSTO Factor C2	89
Tabla 9: Análisis de Varianza variable TIEMPO Factor C2	90
Tabla 10: Niveles del factor W	92
Tabla 11: Resultados consolidados del diseño experimental aplicado al factor peso inercial W	93
Tabla 12: Análisis de Varianza variable COSTO Factor C2	94
Tabla 13: Análisis de Varianza variable TIEMPO Factor C2	95
Tabla 14: Niveles factor tamaño de población	97
Tabla 15: Resultados consolidados del diseño experimental aplicado al factor tamaño de población	98
Tabla 16: Análisis de Varianza variable COSTO Factor tamaño de población	98
Tabla 17: Análisis de Varianza variable TIEMPO factor Tamaño de población	99
Tabla 18: Tratamientos para el segundo ajuste de tamaño de población	103

Tabla 19: Resultados consolidados segunda etapa del diseño aplicado al factor tamaño de población (con iteraciones)	104
Tabla 20: Análisis de Varianza variable COSTO para Tamaño de población con iteraciones.....	104
Tabla 21: Análisis de Varianza variable TIEMPO Tamaño de población con iteraciones	106
Tabla 22: Valores de los parámetros ajustados	108
Tabla 23: Resultados por réplica algoritmo PSO ajustado.....	108
Tabla 24: Mejor configuración de celdas encontrada por el algoritmo ajustado ..	110
Tabla 25: Asignación de máquinas mejor solución encontrada	110
Tabla 26: Distribución de probabilidad para el número de operaciones	112
Tabla 27: Configuración de los seis problemas resueltos.....	113
Tabla 28: Mejores resultados por algoritmo.....	114
Tabla 29: Comparación tiempo de ejecución consumido por las metodologías ..	117
Tabla 30: Resultados detallados Problema 1	118
Tabla 31: Configuración de la mejor solución encontrada al problema 1.....	120
Tabla 32: Necesidades de máquina de la mejor solución encontrada para el problema 1	121

LISTA DE ANEXOS

ANEXO A: CÓDIGO DE PROGRAMACIÓN ALGORITMO PSO FICHERO 'PARAMETROSPSO.M'	132
ANEXO B: CÓDIGO DE PROGRAMACIÓN ALGORITMO PSO FICHERO 'PSO_TRELEA_VECTORIZED.M'	133
ANEXO C: CÓDIGO DE PROGRAMACIÓN FUNCIÓN OBJETIVO FICHERO 'OBJETIVO_INDIVIDUO.M'	139
ANEXO D: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO C1	141
ANEXO E: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO C2	142
ANEXO F: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO W	143
ANEXO G: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO TAMAÑO DE POBLACIÓN.....	144
ANEXO H: RESULTADOS POR RÉPLICA AJUSTE DEL FACTOR TAMAÑO DE POBLACIÓN CON ITERACIONES	145
ANEXO I: DATOS NECESARIOS PARA VERIFICAR SUPUESTOS DE LOS DISEÑOS DE EXPERIMENTOS APLICADOS EN EL AJUSTE DE PARÁMETROS.....	146
ANEXO J: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR C2.....	151
ANEXO K: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR W	154
ANEXO L: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR TAMAÑO DE POBLACIÓN	157
ANEXO M: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR TAMAÑO DE POBLACIÓN CON ITERACIONES.....	160

ANEXO N: CONJUNTO DE DATOS PROBLEMA DE AJUSTE EXPERIMENTAL DEL ALGORITMO PSO	163
ANEXO O: CONJUNTO DE DATOS PROBLEMA 1.....	166
ANEXO P: CONJUNTO DE DATOS PROBLEMA 2	169
ANEXO Q: CONJUNTO DE DATOS PROBLEMA 3.....	172
ANEXO R: CONJUNTO DE DATOS PROBLEMA 4.....	175
ANEXO S: CONJUNTO DE DATOS PROBLEMA 5	178
ANEXO T: CONJUNTO DE DATOS PROBLEMA 6	181
ANEXO U: RESULTADOS COMPLETOS EXPERIENCIA COMPUTACIONAL.....	184
ANEXO V: RESULTADOS ALGORITMO PSO PROBLEMA 2	186
ANEXO W: RESULTADOS ALGORITMO PSO PROBLEMA 3.....	188
ANEXO X: RESULTADOS ALGORITMO PSO PROBLEMA 4	190
ANEXO Y: RESULTADOS ALGORITMO PSO PROBLEMA 5	192
ANEXO Z: RESULTADOS ALGORITMO PSO PROBLEMA 6	194
ANEXO. 1: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 2	196
ANEXO. 2: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 3	198
ANEXO. 3: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 4	200
ANEXO. 4: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 5	202
ANEXO. 5: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 6	204

RESUMEN

TÍTULO: ANÁLISIS DEL DESEMPEÑO DE DOS METODOLOGÍAS META HEURÍSTICAS EN EL PROBLEMA DE FORMACIÓN DE CELDAS DE MANUFACTURA *

AUTOR: Cindy Johanna Rodríguez León **

PALABRAS CLAVES: Celdas de fabricación, tecnología de grupos, manufactura celular, metodologías meta-heurísticas, algoritmo PSO, algoritmos genéticos, transferencias intercelulares.

DESCRIPCIÓN:

En este trabajo de investigación se presenta el análisis de la metodología meta-heurística Particle Swarm Optimizatio (PSO) y su adaptación al problema de formación de celdas de manufactura, bajo un esquema de agrupamiento simultáneo de productos y máquinas. Para ello se llevó a cabo una amplia revisión bibliográfica acerca del problema de optimización a desarrollar y se construyó el estado del arte correspondiente.

Esta revisión bibliográfica sirvió como base para la formulación matemática del problema de formación de celdas de manufactura en el lenguaje propio del algoritmo implementado. En este punto fue necesario realizar un ajuste de parámetros a través de un diseño de experimentos completamente al azar en dos etapas: en la primera etapa se ajustaron cuatro de los parámetros del algoritmo, los más relevantes según la revisión bibliográfica realizada y el criterio del autor con base en el problema de optimización estudiado. En la segunda etapa se tuvieron en cuenta únicamente los parámetros que efectivamente resultaron más influyentes en el desempeño del algoritmo, para lo cual fue necesario incluir un quinto factor en el ajuste de parámetros.

Posteriormente se actualizó y se le hicieron los ajustes necesarios a un algoritmo genético previamente desarrollado. Los dos algoritmos fueron implementados en seis problemas de aplicación diferentes, en el lenguaje de programación Matlab, y las respuestas obtenidas fueron comparadas en términos del costo de la mejor solución encontrada y del tiempo de ejecución consumido para llegar a dicha solución.

* Proyecto de grado en la modalidad de Investigación

** Facultad de Ingenierías Físico-Mecánicas; Escuela de Estudios Industriales y Empresariales; Programa de Ingeniería Industrial; Ing. Jabid Quiroga Méndez, director del proyecto; Ing. Néstor Raúl Ortiz Pimiento, codirector del proyecto

ABSTRACT

TITLE: PERFORMANCE ANALYSIS OF TWO META-HEURISTIC METHODOLOGIES IN MANUFACTURING CELL FORMATION DESIGN*

AUTHOR: Cindy Johanna Rodríguez León**

KEY WORDS: Manufacturing cells, group technology, cellular manufacturing, meta-heuristic methodologies, PSO algorithm, genetic algorithms, intercellular transfers.

DESCRIPTION:

This research dissertation shows the Particle Swarm Optimization (PSO) meta-heuristic analysis and its adaptation to the cell formation problem under a simultaneous clustering of parts and machines approach. In order to carry out the dissertation it was conducted an extensive literature review on the optimization problem to be developed which served to construct the corresponding state of the art.

This literature review served as the basis for the mathematical formulation of the manufacturing cell formation problem in the proper language of the used algorithm. At this point it was necessary to make an adjustment of parameters through a completely randomized experiment design in two stages: in the first stage there were adjusted four of all the algorithm's parameters, the four most important according to the literature review done and the author's judgment based on the optimization problem under study. In the second stage it was taken into account only the parameters that were actually more influential in the performance of the algorithm, for which it was necessary to include a fifth factor in the adjustment of parameters.

Subsequently, updated and made him the necessary adjustments to a previously genetic algorithm developed. Both algorithms, PSO and genetic algorithm, were implemented in six different application problems, in the Matlab programming language, and the results obtained were compared in terms of cost of the best solution found and run time consumed to reach that solution.

* Degree dissertation in research project modality

** Physical-Mechanical Engineering Faculty; Industrial and Managerial Studies School; Industrial Engineering Program; Eng. Jabid Quiroga Méndez, director; Eng. Néstor Raúl Ortiz Pimiento, codirector.

INTRODUCCIÓN

La importancia de responder a la creciente necesidad de variar frecuentemente los sistemas productivos a fin de fabricar nuevos productos, como consecuencia de la rápida obsolescencia de los mismos, de las inesperadas fluctuaciones de demanda, de la cada vez más marcada personalización de los productos entre otras razones, ha dado lugar a nuevos conceptos de sistemas de fabricación. Dentro de estos nuevos conceptos se destaca ampliamente el conocido como “Tecnología de Grupos”, el cual es una práctica de manufactura por la que se agrupan las piezas/productos a fabricar en familias con características similares, bien sea a nivel de geometría (tamaño y/o forma) o de procesamiento (pasos requeridos para su manufactura). Esta práctica de manufactura se deriva de la filosofía Justo a Tiempo, en la cual en vez de contar con departamentos especializados en una operación se busca trabajar con todas las operaciones en un solo lugar, formando celdas o subsistemas completos y controlables.

Dentro de esta filosofía surge el concepto de “Manufactura Celular”, el cual no es más que una aplicación de la Tecnología de Grupos en la que, además de determinar la familia de productos a fabricar, se diseña un taller (llamado célula o celda) formado por la agrupación de máquinas requeridas para la fabricación de cada una de estas familias. Cada celda de producción debe ser capaz de producir cualquier miembro de la familia, omitiendo las operaciones innecesarias, y debe ser diseñada para permitir variaciones en el tamaño dentro de la familia con el mínimo costo de operación del sistema posible. Este problema de formación de celdas de manufactura hace parte de los denominados problemas de optimización combinatoria NP-hard (término científico utilizado en el contexto de la complejidad algorítmica), y la mejor manera de resolverlo es mediante la implementación de métodos meta-heurísticos o aproximados, los cuales proporcionan una buena

solución, no necesariamente óptima, con el empleo de tiempos computacionales razonables.

Dado que tanto el algoritmo PSO como el Algoritmo Genético son modelos meta heurísticos que evocan comportamientos biológicos observados en la naturaleza (Computación Evolutiva), que los dos hacen parte de los llamados algoritmos basados en poblaciones y que constan de elementos similares, ¿cuáles son las diferencias que pueden surgir en la solución final encontrada por cada algoritmo al resolver el mismo problema de optimización? Existen muchas opciones a la hora de implementar modelos meta-heurísticos en la resolución de ese tipo de problemas, y un aspecto importante de la investigación consiste en comparar estos métodos entre sí y llegar a una conclusión acerca de cuál es el modelo más apropiado a implementar en el tipo de problema estudiado, basado en el desempeño mostrado por el modelo en el transcurso de la investigación. El algoritmo PSO es de las últimas técnicas dentro del grupo de computación evolutiva basada en poblaciones que se han explotado y desarrollado hasta el momento, y sin embargo dentro de sus aplicaciones conocidas existen pocas referencias a la solución del problema de formación de celdas de manufactura.

TABLA DE CUMPLIMIENTO DE OBJETIVOS

OBJETIVO	CUMPLIMIENTO
<p>1. Programar el algoritmo PSO, mediante el uso del entorno de programación Matlab, para resolver el problema de formación de celdas de manufactura mediante la adaptación de la formulación matemática general del mismo.</p>	<p>Capítulos 3 y 4</p>
<p>2. Realizar un ajuste experimental para determinar los valores de los parámetros del algoritmo que permitan obtener respuestas más adecuadas a la formulación planteada, a través de la resolución de un problema de aplicación.</p>	<p>Capítulos 5 y 6</p>
<p>3. Resolver seis problemas diferentes de formación de celdas, una vez ajustados los parámetros del algoritmo, implementando tanto PSO como algoritmo genético.</p>	<p>Capítulo 7</p>
<p>4. Comparar los resultados obtenidos por ambos algoritmos en términos del costo de la mejor solución encontrada así como del tiempo de corrida de los mismos, valiéndose de los conceptos y herramientas proporcionadas por la estadística inferencial.</p>	<p>Capítulo 7</p>

1. GENERALIDADES DEL TRABAJO DE GRADO

1.1. DESCRIPCIÓN DEL TEMA DE INVESTIGACIÓN

Como ya es sabido, el sistema de producción Justo a Tiempo está basado en los principios de la filosofía “Lean Manufacturing”, la cual se vale de una serie de estrategias, metodologías y técnicas que le permiten a las empresas mejorar su capacidad de producción, de tal forma que puedan responder de manera eficiente, económica y eficaz a los cambios de su entorno. Dentro de esas estrategias, se encuentra la denominada “Tecnología de Grupos”. Esta estrategia puede ser aplicada en producción de diferentes maneras¹:

- Programación informal y enrutamiento de piezas similares a través de máquinas seleccionadas: este enfoque logra configurar ventajas, pero no se definen familias de piezas formales, y no se realiza un reordenado físico del equipamiento.
- Celda de máquina virtual: este enfoque implica la creación de familias de piezas y la dedicación del equipamiento de la fábrica a esas familias de piezas, pero sin el reordenamiento físico de las máquinas en celdas formales. Las máquinas en la celda virtual permanecen en su ubicación original en la fábrica. El uso de celdas virtuales parece facilitar la distribución de máquinas con otras celdas virtuales de producción de otras familias de piezas.
- Celdas de máquina formal: este es el enfoque convencional ‘Tecnología de Grupos’ en el cual un grupo de máquinas distintas son reubicadas físicamente en una celda que está dedicada a la producción de una o un limitado lote de familias de piezas. Las máquinas en una celda de máquina formal están situadas en las proximidades para reducir al mínimo la manipulación de las partes, tiempo de rendimiento, tiempo de instalación y trabajo en proceso.

¹ http://dmi.uib.es/~burguera/download/sif/tecnologia_grups_20809.pdf

Como puede verse, no todas las empresas manufactureras implementan la Tecnología de Grupos por medio de celdas de fabricación (es importante aclarar que el sistema de producción es intermitente). El interés de este trabajo de grado se centra en la aplicación de la Tecnología de Grupos a partir de celdas de fabricación. Las ventajas de las Celdas de fabricación son ampliamente conocidas por los expertos en el tema, entre ellas se pueden mencionar: la reducción del lead time, la reducción del trabajo en proceso, el mejoramiento de la calidad, la reducción en el tiempo de respuesta a los clientes, la reducción de distancias de movimiento de las partes y productos, el incremento en la flexibilidad de fabricación, la reducción de costos por unidad, la simplificación del programa de producción y el involucramiento de los empleados en el proceso de producción.

Para conformar celdas de fabricación, se parte de una situación inicial, en donde se dispone de máquinas de diferente tipo que son utilizadas por varias referencias de producto; posteriormente debe definirse el número de celdas a constituir y finalmente se procede a agrupar las máquinas en celdas de fabricación. En principio, cada celda de fabricación debería dedicarse a procesar una familia de productos, sin embargo esta exclusividad no es fácilmente alcanzable, razón por la cual algunas referencias pertenecientes a una familia de productos deberán compartir máquinas con alguna otra familia de productos. Las posibilidades de agrupamiento de máquinas son elevadas, sin embargo no todas las soluciones son factibles debido a las múltiples restricciones que deben cumplirse. Buscar la mejor alternativa de agrupamiento es un problema de optimización combinatoria ampliamente abordado por los investigadores, y el problema es conocido como "Cell Formation Problem" o bien "Problema de Formación de Celdas de Manufactura".

La literatura relacionada con la implementación del algoritmo PSO en la resolución del problema de formación de celdas de manufactura es realmente escasa. De hecho, las investigaciones de este algoritmo en el problema de formación de

celdas se inician apenas en el año 2006 con la publicación de (Andres & Lozano, 2006) en la cual se desarrolla el problema pero no en términos de costo sino de número de transferencias de lotes de productos entre las celdas. Esta respuesta es apenas un dato necesario para calcular el costo de incurrir en tales transferencias, que a su vez representa solo una de las dos variables necesarias para el cálculo del costo total que implica mantener determinada configuración de celdas. Incluso, en esta publicación, no se tiene en cuenta como información de entrada el tiempo de operación requerido para la fabricación de cada producto ni restricciones acerca del tamaño mínimo o máximo de celdas a conformar. Así mismo, dentro de la revisión bibliográfica desarrollada solo se encontró un trabajo de investigación en el que se compara el desempeño de un Algoritmo Genético con un algoritmo PSO en la resolución del problema de formación de celdas². Sin embargo, el algoritmo PSO desarrollado en el trabajo corresponde a una variación demasiado compleja del mismo, un algoritmo híbrido que involucra conceptos propios de la metodología de búsqueda difusa y que en todo caso no toma en consideración importantes elementos de entrada que sí son tenidos en cuenta en la formulación del problema realizada por (Gómez Gómez, 2007), para un modelado del problema mucho más aproximado a la realidad. Es por ello que para el presente trabajo de grado se propone la formulación y programación de un algoritmo PSO tradicional que resuelva el problema de formación de celdas de manufactura en términos de minimización de costos³ y bajo cierto número de restricciones, y la evaluación de su desempeño en diversos problemas propuestos a través de la comparación con los resultados obtenidos con la implementación de Algoritmo Genético. La comparación de estas dos metodologías, teniendo en cuenta información de entrada que no se había incluido en los problemas modelados de la bibliografía consultada es nueva dentro de la línea de

² (Caprihan, Slomp, Gusaran, & Agarwal, 2009)

³ Teniendo en cuenta además importantes datos de entrada como demanda requerida por tipo de producto, tiempos de operación y demás elementos descritos en el apartado 3.2. Esto con el fin de obtener un modelado del problema mucho más ajustado a la realidad.

investigación y desarrollo que han seguido los investigadores del problema de formación de celdas.

1.2. OBJETIVOS DEL TRABAJO DE GRADO

1.2.1. Objetivo General

Comparar el desempeño de dos modelos meta-heurísticos (PSO y Algoritmo Genético) mediante su implementación en la solución del problema de formación de celdas de manufactura.

1.2.2. Objetivos Específicos

- ✓ Programar el algoritmo PSO, mediante el uso del entorno de programación Matlab, para resolver el problema de formación de celdas de manufactura mediante la adaptación de la formulación matemática general del mismo.
- ✓ Realizar un ajuste experimental para determinar los valores de los parámetros del algoritmo que permitan obtener respuestas más adecuadas al tipo de problema y formulación planteada. Para ello, se resolverá un problema específico consistente en la fabricación de 30 tipos de productos con el empleo de 10 tipos de máquinas a ser agrupadas en 4 celdas de manufactura⁴. Los valores de los demás datos de entrada necesarios para la resolución del problema serán generados de manera aleatoria.
- ✓ Resolver seis problemas diferentes de formación de celdas, una vez ajustados los parámetros del algoritmo, implementando tanto PSO como algoritmo

⁴ Esta configuración específica está basada en un problema previamente definido en el trabajo de grado (Gómez Gómez, 2007).

genético⁵. Igualmente, la configuración específica de estos seis problemas estará dada por la generación aleatoria de valores para cada uno de los datos de entrada necesarios para la resolución de los mismos.

- ✓ Comparar los resultados obtenidos por ambos algoritmos en términos del costo de la mejor solución encontrada así como del tiempo de corrida de los mismos, valiéndose de los conceptos y herramientas proporcionadas por la estadística inferencial.

1.3. ALCANCE DEL TRABAJO DE GRADO

El presente trabajo abarca todas las etapas necesarias para la programación del algoritmo meta-heurístico de Optimización del Enjambre de Partículas, PSO, con miras a la resolución del problema de formación de celdas de manufactura y se extenderá hasta la evaluación del desempeño de dicho método a través de la comparación con los resultados obtenidos por el algoritmo genético, en términos de la calidad de respuesta obtenida y de los tiempos computacionales empleados, en los casos de estudio propuestos.

Para la consecución del objetivo general trazado para este trabajo de grado se resolverán seis problemas de formación de celdas de manufactura, en una primera etapa implementando algoritmo PSO y posteriormente implementando algoritmo genético en los mismos seis problemas resueltos por PSO.

1.4. INDICACIONES GENERALES SOBRE EL DOCUMENTO

⁵ Los valores de los parámetros relacionados con el algoritmo genético ya están ajustados puesto que se han tomado del trabajo de grado realizado en (Gómez Gómez, 2007). La configuración de cada uno de los seis problemas a desarrollar corresponden a una adaptación de los problemas desarrollados en el mismo trabajo de grado. La mayoría de datos de entrada de los diferentes problemas se generan de manera aleatoria, luego cabe aclarar que los problemas a resolver en este trabajo de investigación no son exactamente iguales a los resueltos por el autor del trabajo de investigación de referencia.

Con base en los objetivos específicos trazados y en el alcance planteado para el presente trabajo de investigación, el documento se ha organizado de la siguiente manera:

En el capítulo 2 se presenta el marco teórico conceptual que proporciona los conceptos básicos y los fundamentos de los temas trabajados durante la investigación. Comienza con la introducción a la tecnología de grupos para luego pasar a la manufactura celular, que es una aplicación de la anterior. A continuación se aborda de manera general el problema de formación de celdas de manufactura, señalando por qué hace parte de los denominados problemas de optimización combinatoria. En ese momento se introducen las metodologías para la solución de este tipo de problemas, haciendo un recorrido por la clasificación de los diferentes modelos y centrándose especialmente en las metodologías meta-heurísticas que se han aplicado al problema de formación de celdas. Finalmente se llega a la descripción del algoritmo PSO como una de las meta-heurísticas empleadas en la resolución de problemas de optimización. Se hace especial énfasis en la herramienta computacional general utilizada para la implementación del algoritmo.

En el capítulo 3 se entra en más detalle sobre el problema de formación de celdas de manufactura, presentando la formulación matemática del mismo y describiendo de manera precisa la información, datos de entrada, restricciones, variables de decisión y demás elementos necesarios para el planteamiento matemático y posterior obtención de resultados del problema desarrollado.

El capítulo 4 corresponde a la programación del algoritmo PSO. En este capítulo, y teniendo como referencia la información relacionada con el algoritmo presentada en el marco teórico conceptual, se describe de manera detallada el esquema de funcionamiento del algoritmo específicamente para el problema de formación de

celdas de manufactura. Corresponde básicamente a la descripción de los códigos de programación empleados.

En el capítulo 5 se desarrolla el ajuste de parámetros del algoritmo, a través de la aplicación de diseños de experimentos con análisis de varianza. El resultado es la selección de los valores de los parámetros que mejor rendimiento le generan al algoritmo y con los que se realizará la experiencia computacional para la posterior comparación de resultados con los arrojados por un algoritmo genético.

En el capítulo 6 se presentan los resultados detallados del algoritmo PSO para el problema que se utilizó como referencia para el ajuste del mismo.

Por último, en el capítulo 7 se desarrolla la experiencia computacional propiamente dicha. Para ello se resuelven seis problemas de diferente configuración, tanto por algoritmo genético como por algoritmo PSO, y una vez obtenidos los resultados se hace la comparación en términos de costo de la mejor solución encontrada y del tiempo de ejecución consumido por cada meta-heurística.

En los anexos se pueden encontrar los códigos de programación del algoritmo PSO, la verificación de los supuestos correspondientes al diseño de experimentos aplicado al ajuste de cada factor considerado, los resultados completos del ajuste realizado, la configuración y demás datos de entrada de cada uno de los problemas resueltos durante la experiencia computacional, los resultados completos obtenidos durante esa misma etapa y la mejor solución encontrada por cada meta-heurística en cada uno de los problemas de aplicación resueltos.

2. MARCO TEÓRICO CONCEPTUAL

2.1. TECNOLOGÍA DE GRUPOS

Es una práctica de manufactura por la que se agrupan las piezas/productos a fabricar en familias con características similares, bien sea a nivel de geometría (tamaño y/o forma) o de procesamiento (pasos requeridos para su manufactura).

Para la clasificación y conformación de las diferentes familias de productos se puede utilizar cualquiera de las tres técnicas que se describen a continuación (Sampol Aznar & Carmona Gómez, 2009):

- Inspección visual directa: se van clasificando las piezas a partir del examen de los planos y según sus procesos de fabricación en clases, subclases, grupos, subgrupos, etc. Pueden utilizarse las dimensiones necesarias hasta la formación de familias con el grado de semejanza requerido. El problema de esta técnica radica en la definición de los criterios de clasificación pues dependen en gran medida de la persona encargada de la inspección, es decir, se torna en un procedimiento subjetivo. Este problema es tanto más grave cuanto mayor el número de piezas y menor su semejanza. Para tratar de solventar este inconveniente, un estudio estadístico de la distribución de las piezas permite establecer los criterios básicos de clasificación. Esta es la técnica menos costosa pero también la menos precisa para la formación de familias de productos.
- Clasificación y codificación: es la técnica que consume mayor cantidad de tiempo. Las similitudes⁶ de todas las piezas son identificadas y reflejadas en un

⁶ Estas similitudes pueden ser de dos tipos: i) Similitudes de diseño: la forma de las piezas es parecida. Ayudan en el diseño de una pieza cuando ya se tiene otra similar previamente diseñada, permitiendo ahorrar mucho tiempo de diseño. ii) Similitudes de proceso: similitudes en la secuencia

código. Normalmente estos dos tipos de similitudes no están del todo relacionadas, lo cual permite definir tres tipos distintos de sistemas: i) los basados en similitudes de diseño, ii) los basados en similitudes de proceso, iii) los basados en ambos tipos de similitudes. Esto es así ya que para un proceso puede ser muy eficiente la codificación mediante similitudes de diseño y para otro ser completamente ineficiente.

- Análisis del flujo de producción: es un método de agrupación de las máquinas empleadas en los productos de fabricación, teniendo en cuenta qué máquina necesita cada una de las partes que se fabrican. Es un método basado en la hoja de ruta de cada producto a fabricar. Para ello, utiliza una matriz en la que las columnas representan las máquinas, y las filas representan las partes. A cada máquina se le asigna un valor numérico, y a cada parte una letra. Cuando una parte requiere una máquina en particular, el número de secuencia de funcionamiento se encuentra en la intersección de la hoja de cálculo⁷.

En esta última técnica de clasificación y formación de familias de productos se aprecia que no solo se tiene en cuenta el producto y sus características físicas o de procesamiento sino que además se considera importante el o los tipos de máquinas requeridas por los diferentes productos para conformar las familias. Cuando esto sucede ya no se está hablando de manera general del concepto de Tecnología de Grupos; se está haciendo referencia específica a una de sus aplicaciones conocida como “Manufactura Celular”. Además, en la técnica de Análisis del flujo de producción se introduce el concepto de “matriz de incidencia parte-máquina”, el cual es un elemento de información de entrada fundamental para la resolución del problema de formación de celdas de manufactura.

de producción de la pieza. Ayudan a la hora de diseñar las celdas y la posibilidad de automatizar el control de proceso, usando el código como el mismo control de proceso.

⁷ Para mayor detalle consultar en (Sampol Aznar & Carmona Gómez, 2009)

2.2. MANUFACTURA CELULAR

La Tecnología de Grupos es una práctica de manufactura que se deriva de la filosofía Justo a Tiempo, en la cual en vez de contar con departamentos especializados en una operación se busca trabajar con todas las operaciones en un solo lugar, formando celdas o subsistemas completos y controlables.

Dentro de esta filosofía surge el concepto de “Manufactura Celular”, el cual no es más que una aplicación de la Tecnología de Grupos en la que, además de determinar la familia de productos a fabricar, se diseña un taller (llamado célula) formado por la agrupación de máquinas requeridas para la fabricación de cada una de estas familias. Cada celda de producción debe ser capaz de producir cualquier miembro de la familia, omitiendo las operaciones innecesarias, y debe ser diseñada para permitir variaciones en el tamaño dentro de la familia con el mínimo costo de operación del sistema posible. Como lo plantea (Gómez Gómez, 2007) en su tesis de grado, lo anterior constituye un problema de agrupamiento conocido como el problema de “Formación de celdas de manufactura” cuyo objetivo es obtener economías de escala en talleres de trabajo que equivalen a los obtenidos mediante la producción en masa.

Concretamente, dado un conjunto de tipos de partes, requerimientos de procesamiento, demanda por tipo de producto y recursos disponibles para su fabricación, un diseño general de manufactura celular se compone de los siguientes enfoques: las partes similares son normalmente agrupadas en familias de productos de acuerdo con sus requerimientos de procesamiento; luego, las máquinas de diferente tipo son agrupadas para formar celdas de manufactura y consecuentemente las familias de productos son asignadas a las celdas. Dependiendo de los procedimientos involucrados en el problema, tres metodologías de solución son propuestas por (Selim, Askin, & Vakharia, 1998):

- Las familias de partes se forman como primera instancia y por lo tanto las máquinas son agrupadas en celdas de acuerdo a los requerimientos de procesamiento de las familias formadas. Esto se conoce como “identificación de familia de partes”.
- Las celdas de fabricación (agrupación de máquinas heterogéneas) se generan primero basado en uniformidades en las rutas de procesamiento de las partes y luego las familias de partes son asignadas a las celdas. Esto se conoce como “identificación de grupos de máquinas”.
- Las familias de partes y las celdas de manufactura son formadas de manera simultánea, lo cual se conoce como “agrupación de familia de partes/máquina”.

En este sentido, lo que se quiere lograr con la implementación de diseños de manufactura celular es hacer extensivas a las pequeñas series las ventajas obtenidas en la fabricación de grandes series⁸. De esta manera las empresas que deciden adoptar esta filosofía logran importantes beneficios en sus procesos productivos, dentro de los cuales se pueden mencionar (Mestre Genovart & Parets Fiol, 2009):

- Acortar el tiempo de fabricación: a través de cambios en la configuración, transporte de piezas, tiempo de espera y tamaño de lotes.
- Reducir el inventario en proceso: con tamaños de lote más pequeños y tiempos de fabricación más cortos.
- Mejora de la calidad: permitiendo que cada celda se especialice en un pequeño número de piezas diferentes. Esto reduce variaciones en el proceso.
- Simplificación en el programa de producción: la similitud entre las partes en la familia reduce la complejidad en el programa de producción. En lugar de la programación de las piezas a través de una secuencia de máquinas en un proceso de diseño tipo, las piezas están programadas a través de la celda.

⁸ Para profundizar sobre el tema consultar en (Díaz Ortega)

- Reducir el tiempo de instalación: mediante el uso de un grupo de herramientas (de corte, sujeción y accesorios) que han sido diseñadas para la familia de piezas, en lugar de herramientas genéricas, las cuales han sido diseñadas para una pieza individual. Esto reduce el número de herramientas individuales requeridas así como el tiempo para cambiar las herramientas entre piezas.
- Reducir el tiempo de respuesta de los pedidos.
- Reducir las distancias de movimiento de piezas.
- Incrementar la flexibilidad de fabricación.
- Reducir los costos por unidad.
- Facilitar la participación del empleado: al reducirse la variabilidad de tareas, los períodos de entrenamiento para los trabajadores son más cortos y pueden ser entrenados más específicamente para las partes incrementando por consiguiente el control de calidad.
- Reducir el inventario de productos terminados.

De esta manera, las ventajas generadas a raíz de la adopción e implementación de sistemas de manufactura celular son sorprendentemente efectivas y sumamente importantes y necesarias si se quiere crecer y mantener el éxito en un mercado tan competitivo como en el que se desenvuelven las empresas de hoy en día. Sin embargo, así como se pueden obtener grandes beneficios a partir de su implementación, el problema de formación de celdas de manufactura no es tarea fácil y requiere de un profundo y cuidadoso análisis del sistema productivo de cada organización, en cuanto a productos que se fabrican, recursos disponibles para producir, operaciones necesarias para transformar materiales y materias primas, tiempos consumidos por cada operación necesaria por producto, costos de mantener el sistema operando (materiales, mano de obra, máquinas y equipos, costos indirectos de fabricación), espacio disponible para la ubicación de los recursos de producción, flujo de producción, etc. Solo así se tiene la posibilidad de contar con información objetiva, imprescindible para la búsqueda de la

configuración más adecuada que permita obtener economías de escala al menor costo posible.

Como se puede advertir en el párrafo anterior, el problema de formación de celdas de manufactura es un problema de búsqueda de la mejor configuración, esto es, hace parte de los denominados “problemas de optimización combinatoria”. Más aún, el problema de formación de celdas de manufactura es un problema de optimización combinatoria *NP-hard*⁹.

En (Martí, 2003) se encuentra que *“el objetivo de este tipo de problemas es encontrar el máximo (o el mínimo) de una determinada función sobre un conjunto finito de soluciones, denotadas por S. Cabe señalar que dada la finitud de S, las variables han de ser discretas, restringiendo su dominio a una serie finita de valores. De forma más precisa, estos problemas se pueden expresar como encontrar el valor de unas variables de decisión para los que una determinada función objetivo alcanza su valor máximo o mínimo. El valor de dichas variables en ocasiones está sujeto a unas restricciones. Habitualmente, el número de elementos de S es muy elevado, haciendo impracticable la evaluación de todas sus soluciones para determinar el óptimo”*. De allí que se haga necesario el uso de métodos que estén basados en alguna forma de exploración del conjunto de soluciones factibles S.

Bajo tales consideraciones, la mejor manera de resolver el problema de formación de celdas de manufactura es mediante la implementación de métodos heurísticos o aproximados, los cuales proporcionan una buena solución, no necesariamente óptima, con el empleo de tiempos computacionales razonables, en contraposición

⁹ Término científico utilizado en el contexto de la complejidad algorítmica. Corresponde al acrónimo en inglés de *nondeterministic polynomial time* (tiempo polinomial no determinístico). Pretende reflejar la idea intuitiva de problema ‘difícil de resolver’, es decir, aquel problema para el cual no se puede garantizar el encontrar la solución óptima en un tiempo razonable.

a los métodos exactos. Sin embargo, “en los últimos años han aparecido una serie de métodos bajo el nombre de Meta-Heurísticos con el propósito de obtener mejores resultados que los alcanzados por los heurísticos tradicionales”¹⁰. En este momento existe un gran desarrollo y crecimiento de estos métodos, concretamente de los basados en poblaciones e inspirados en la evolución biológica como lo son los Algoritmos Genéticos y la Optimización del Enjambre de Partículas (PSO por sus siglas en inglés *Particle Swarm Optimization*).

2.3. METODOLOGÍAS PARA LA RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA

Dependiendo de la complejidad del problema a desarrollar, existen diferentes clasificaciones de las metodologías aplicadas a la resolución del problema de celdas de manufactura: (Martí, 2003) clasifica estas metodologías en tres grupos: métodos exactos, metodologías heurísticas y metodologías meta-heurísticas. Por su parte, (Grammatoula & Wilson, 2009) hace una clasificación en la que los métodos de resolución quedan agrupados bajo las siguientes categorías: análisis de agrupamiento, enfoques de gráficos de partición, métodos de programación matemática, algoritmos heurísticos y meta-heurísticos y finalmente, metodologías de inteligencia artificial. Y así se pueden encontrar muchas otras clasificaciones de estas metodologías, otros ejemplos pueden encontrarse en (Yasuda & Yin, 2001) y también en (Chan, Lam, & Lee, 1999). De todas ellas, se ha optado por la siguiente clasificación: métodos exactos, modelos de programación matemática, metodologías heurísticas y metodologías meta-heurísticas.

¹⁰ (Martí, 2003)

2.3.1. Métodos exactos

Proporcionan una solución óptima del problema pero generalmente a costa de elevados tiempos y costos computacionales. Tienen aplicación en problemas relativamente fáciles de resolver, es decir, aquellos en los que tanto la función objetivo como las restricciones son expresiones lineales. De hecho, dentro de los métodos más utilizados se encuentran la programación lineal y la programación entera como primera medida, así como la programación dinámica, la programación no lineal y la programación estocástica, entre otras.

Para la resolución de problemas de programación lineal, la técnica de mayor uso debido a su probada eficiencia es el denominado “Método simplex”, un algoritmo algebraico iterativo desarrollado por George Dantzing en 1947.

Dentro de la metodología de programación entera existen tres enfoques principales: las técnicas de ramificación y acotamiento, las técnicas con generación de planos cortantes y las técnicas de ramificación y cortadura, que no son más que un enfoque algorítmico combinado de las dos primeras, que provee resultados mucho más eficientes en problemas de gran escala. En (Castillo, Conejo, Pedregal, García, & Alguacil, 2002) se presenta una revisión amplia y detallada de las técnicas utilizadas en la resolución de problemas de programación lineal entera.

2.3.2. Modelos de Programación Matemática

Constituyen un soporte para el análisis cuantitativo en los procesos de toma de decisiones o para validar la construcción de algoritmos de búsqueda aleatoria y pseudo-aleatoria. Los beneficios que reporta como técnica de resolución del problema de formación de celdas de manufactura radican en que permite la inclusión de secuencias ordenadas de operaciones, planes alternativos de

procesamiento, incertidumbre en los niveles de demanda, operaciones no consecutivas de partes de la misma máquina y tiempos de preparación del proceso (Gómez Gómez, 2007).

Dentro de esta clasificación, el método más conocido y de más amplio uso por los investigadores es el denominado “método p-mediana”, el cual cuando era aplicado a problemas reales, en sus inicios, era gravemente restrictivo debido a dos inconvenientes principales: el tamaño del problema y el tipo de software utilizado. Debido a esto, en (Won & Lee, 2004) se propone un enfoque p-mediana modificado para la formación eficiente de celdas de manufactura con el objetivo de maximizar la suma de similitudes entre máquinas en la misma celda. De esta manera el modelo logra obtener dos grandes ventajas, en contraste con los dos inconvenientes que presentaba en su versión original: velocidad en la implementación y una gran capacidad de formación de celdas aún con el uso de software de propósitos educativos.

Adicional al modelo p-mediana, en (Foulds, French, & Wilson, 2006) se desarrolla un modelo de programación matemática entero mixto en el que se introduce por primera vez la modificación de máquinas. En el problema de formación de celdas es a menudo importante ser capaces de reasignar las partes a máquinas adicionales, con el objetivo de crear una mejor configuración del sistema de celdas y también evitar la duplicación de máquinas lo cual puede resultar bastante costoso. Por lo tanto, se introdujo la modificación de la máquina para permitir la transformación posterior de una parte en la misma celda para reducir los viajes intercelulares y se argumenta que el costo de tales modificaciones podría ser compensado por la consiguiente reducción de los gastos de viaje entre celdas.

2.3.3. Metodologías Heurísticas

“La existencia de una gran cantidad y variedad de problemas difíciles, que aparecen en la práctica y que necesitan ser resueltos de forma eficiente, impulsó el desarrollo de procedimientos eficientes para encontrar buenas soluciones aunque no fueran óptimas, con el empleo de tiempos computacionales razonables. Estos métodos, en los que la rapidez del proceso es tan importante como la calidad de la solución obtenida, se denominan heurísticos o aproximados” (Martí, 2003).

Gran parte de las metodologías heurísticas son procedimientos iterativos mejorados que se mueven de una solución dada a una solución en su vecindario que es mejor en términos del valor de la función objetivo. Así, en cada iteración el procedimiento de búsqueda local identifica y evalúa soluciones en el vecindario de la solución actual, selecciona la mejor de acuerdo con un criterio dado e implementa las transformaciones requeridas para establecer la solución seleccionada como solución actual. Las iteraciones continúan hasta que no es posible el mejoramiento de la solución actual encontrada (Crainic & Toulouse).

Varios autores han desarrollado el tema de las heurísticas aplicadas a la resolución del problema de formación de celdas de manufactura. En (Mukattash, Adil, & Tahboub, 2002) se proponen tres procedimientos heurísticos: dada una solución de formación de celdas, las heurísticas fueron diseñadas para asignar partes a las células en presencia de planes de procesamiento alternativos, múltiples máquinas alternativas y tiempos de procesamiento. Estas consideraciones llevaron a la eliminación de elementos excepcionales¹¹, los cuales podían ser luego agregados a las máquinas cuello de botella con el consiguiente

¹¹ Si una parte es asignada a una celda de manufactura dada, cualquier operación incluida en su hoja de ruta que tenga que ser procesada por fuera de la celda asignada constituye un elemento excepcional. Es un término que hace referencia a los movimientos intercelulares que normalmente deben ocurrir cuando se desarrollan problemas de formación de celdas de manufactura.

incremento de la utilización de la máquina. Por su parte, en (Chan, Lam, & Lee, 1999) se desarrolla un algoritmo heurístico que aborda problemas de asignación de máquinas sólo cuando el flujo intracelular de materiales es tenido en cuenta. Este algoritmo superó a otros enfoques en la medida en que proporcionaba soluciones más cercanas a las soluciones óptimas. En (Kim, Baek, & Baek, 2004) se considera el problema de formación de celdas teniendo en cuenta una formulación de máquinas multi-objetivos. La ruta de las familias de productos y las celdas de máquinas requerían ser determinadas de manera tal que se lograra minimizar la suma total de movimientos intercelulares de las partes y el desbalance de la carga de trabajo de las máquinas.

Existen métodos heurísticos de naturaleza muy diferente, por lo que es complicado dar una clasificación completa de los mismos. Pueden encontrarse categorías como métodos de descomposición, métodos inductivos, métodos de reducción¹². Sin embargo, existen dos sub-clasificaciones importantes y muy utilizadas por los investigadores dedicados al tema. Estas sub-clasificaciones de los métodos heurísticos son:

- Métodos constructivos: *“construyen paso a paso una solución del problema. Usualmente son métodos deterministas y suelen estar basados en la mejor elección en cada iteración. Estos métodos han sido muy utilizados en problemas clásicos como el del agente viajero”*¹³ (Martí, 2003).
- Métodos de búsqueda local: *“comienzan con una solución del problema y la mejoran progresivamente. El procedimiento realiza en cada paso un movimiento de una solución a otra con mejor valor. El método finaliza cuando,*

¹² Para una mejor comprensión de este tipo de metodologías consultar en (Martí, 2003)

¹³ Uno de los problemas más famosos en el campo de la optimización combinatoria computacional. Consiste en un agente de ventas que tiene que visitar n ciudades comenzando y terminando en la misma ciudad, visitando solo una vez cada ciudad y haciendo el recorrido de costo mínimo que bien puede estar expresado en términos de tiempo o de distancia.

para una solución hallada, no existe ninguna solución accesible que la mejore”
(Martí, 2003).

Estas dos sub-clases de métodos heurísticos, especialmente la última, constituyen la base de los procedimientos o metodologías meta-heurísticas.

2.3.4. Metodologías Meta-Heurísticas

Los procedimientos meta-heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos, puesto que este tipo de problemas no tienen un algoritmo o heurística específica que dé una solución satisfactoria; o bien no es posible implementar ese método óptimo. Los meta-heurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos (Osman & Kelly, 1996). Aunque cada meta-heurística tiene su propio comportamiento y características, todas comparten una serie de componentes fundamentales y ejecución de operaciones similares que hacen que sean clasificadas como tales. Estos elementos en común corresponden a: inicialización, vecindarios, criterio de selección del vecindario, selección de candidatos, criterio de aceptación y criterio de parada¹⁴.

Al igual que con los métodos de programación matemática y los métodos heurísticos de optimización, existe variedad de criterios diferentes para la clasificación de las técnicas pertenecientes al grupo de metodologías meta-heurísticas. Por ejemplo, en (Martí, 2003) se hace una clasificación de los modelos meta-heurísticos en tres grupos básicos: Búsqueda Tabú, Recocido Simulado y Métodos Evolutivos, que incluye Algoritmos Genéticos y Búsqueda Dispersa. En (Tamal, Sourav, Manojit, & Pranab, 2010) se hace una clasificación, la más

¹⁴ Definición y mayores detalles de estos conceptos en (Crainic & Toulouse)

reciente de todos los trabajos de investigación relacionados, mucho más completa y detallada de estas metodologías, clasificación que se encuentra representada en la tabla 1.

Por su parte, en (Grammatoula & Wilson, 2009) se hace una clasificación muy similar a la anterior, menos compleja aunque igualmente completa y válida, lo cual proporciona una mejor comprensión del esquema bajo el cual se mueven las técnicas existentes asociadas a las meta-heurísticas. Los autores dividen las metodologías meta-heurísticas en dos grandes grupos: las basadas en trayectorias y las basadas en poblaciones. Dentro de las basadas en trayectorias se incluyen el Recocido Simulado y la Búsqueda Tabú mientras que en las basados en poblaciones las más importantes corresponden a Algoritmos Genéticos, algoritmo PSO y algoritmo de Optimización de la Colonia de Hormigas.

Los modelos meta-heurísticos basados en trayectorias se caracterizan por partir de una única solución. La actualización de esta solución se realiza examinando el vecindario, de esta manera se genera una trayectoria.

Tabla 1: Clasificación de las metodologías Meta-heurísticas

Modelos Meta-heurísticos	Probabilísticos	Métodos Basados en Poblaciones	Algoritmos Evolucionarios	Algoritmos Genéticos
				Programación Genética
				Evolución Diferencial
				Búsqueda Dispersa
				Programación Evolucionaria
				Optimización Evolucionaria de

				Gránulos
				Algoritmo Memético
				Optimización de la Colonia de Hormigas
				Optimización del Enjambre de Partículas (PSO)
				Algoritmo de Abejas
				Algoritmo del Flujo de Agua
				Métodos Basados en Solución Simple
Determinísticos	Búsqueda Tabú			

Fuente: Autor

La mayoría de estos algoritmos surgen como extensiones de los métodos heurísticos de búsqueda local a los que se les añade algún mecanismo para escapar de los mínimos locales, debido a que su rendimiento es poco satisfactorio. Como criterios de parada se utilizan por lo general: un tiempo máximo de procesamiento, un número máximo de iteraciones, encontrar una solución S de determinada calidad o detectar que no se produjeron mejoras en las últimas iteraciones (Salto, 2009).

Entre tanto, los modelos meta-heurísticos basados en poblaciones trabajan con un conjunto de soluciones comúnmente denominado ‘población’ en cada iteración. De esta manera, los modelos basados en poblaciones proveen una forma natural e intrínseca de exploración del espacio de búsqueda.

A continuación se describen los avances e investigaciones que se han hecho relacionados con el problema de formación de celdas, de las principales meta-

heurísticas probabilísticas, que corresponden a los métodos de resolución más avanzados que existen actualmente

- Recocido Simulado: procedimiento propuesto en (Kirkpatrick, Gelatt, Vecchi, & Vecchi, 1983). Este procedimiento se basa en una analogía del proceso de enfriamiento físico de metales al someterlos a un baño de agua caliente. Es una técnica que ha sido probada con éxito en numerosos problemas de optimización, mostrando gran habilidad para evitar quedar atrapado en óptimos locales.

En (Liu & Wu, 1993) se introduce una forma general de esta técnica para el problema de formación de celdas de manufactura considerando la penalización de los costos en la función objetivo. En (Chen & Srivastava, 1994) se propone un modelo de programación cuadrática para maximizar la suma de las similitudes de las máquinas dentro de las celdas, sujeto a la limitación del tamaño de las mismas. En (Zolfaghari & Liang, Machine cell/part family formation considering processing times and machine capacities: a simulated annealing approach, 2003) se toma en cuenta el tiempo de procesamiento y la capacidad y duplicación de las máquinas. En (Su & Hsu, 1998) los autores presentan una técnica de Recocido Simulado paralela con elementos de Algoritmos Genéticos por medio de la cual se minimizan los costos totales, el desbalance de la carga de las máquinas, además de tomar en consideración la secuencia de operaciones, el tiempo de preparación y de operación y los costos de los movimientos intracelulares e intercelulares de las partes fabricadas.

- Búsqueda Tabú: metodología introducida en (Glover, 1989). Es una técnica basada en la interacción entre la memoria a corto plazo y la memoria a largo

plazo¹⁵. Ambos tipos de memoria llevan asociadas sus propias estrategias y atributos, y actúan en ámbitos diferentes. En especial, la memoria a largo plazo tiene dos estrategias asociadas: intensificar (regresar a regiones ya exploradas para estudiarlas más a fondo) y diversificar (visitar nuevas áreas no exploradas del espacio de soluciones) la búsqueda.

En (Onwubolu & Songore, 2000) se formula un problema de formación de celdas de manufactura con tres funciones objetivo: minimizar los movimientos intercelulares, minimizar la variación en la carga de las celdas y la combinación de los dos objetivos anteriores. Partiendo de estos objetivos se diseña un método que ofrece libertad para considerar el tamaño máximo de las celdas y del número de máquinas por cada celda. En (Spiliopoulos & Sofianopoulo, Designing manufacturing cells: a staged approach and a Tabu search algorithm, 2003) se desarrolla un enfoque de diseño de celdas con escenarios múltiples en donde la parte primaria es implementada por un algoritmo de búsqueda tabú integrado con estructuras adecuadas de memoria a corto y largo plazo. La estrategia de búsqueda global representa el beneficio de la memoria de adaptación y la exploración de respuesta. Además se implementa un diseño del experimento para el ajuste de los parámetros de entrada con el fin de detectar las soluciones casi óptimas, de manera más eficiente.

- Optimización de la Colonia de Hormigas: es de las técnicas de optimización de más reciente aparición, propuesta en (Dorigo & Stützle, 2004). Se vale de estrategias inspiradas en el comportamiento de las colonias de hormigas para descubrir fuentes de alimentación: una hormiga va depositando feromonas a lo largo del camino las cuales ayudarán a encontrar la comida al resto de las

¹⁵ La memoria a corto plazo suele almacenar atributos de soluciones recientemente visitadas, y su objetivo es explorar a fondo una región dada del espacio de soluciones. En ocasiones se utilizan estrategias de listas de candidatos para restringir el número de soluciones examinadas en una iteración dada o para evitar que la búsqueda quede atrapada en óptimos locales. La memoria a largo plazo almacena las frecuencias u ocurrencias de atributos en las soluciones visitadas tratando de identificar o diferenciar regiones.

hormigas. Como resultado de la exploración y cooperación entre los individuos de la colonia se establece, en general, el camino más corto entre dicha fuente de alimentación y el hormiguero. En (Salto, 2009) se encuentra que la técnica se basa en dos pasos principales: la construcción de una solución basada en el comportamiento de una hormiga y la actualización de los rastros de feromonas artificiales.

Este algoritmo ha sido exitosamente implementado en la resolución del problema de formación de celdas de manufactura. En (Prabhakaran, Muruganandam, Asokan, & Girish, 2005) se propone un enfoque del algoritmo para minimizar la variación total de la carga en las celdas y los movimientos intercelulares considerando las demandas de las partes, las secuencias de enrutamiento, los tiempos de procesamiento, la capacidad de las máquinas y el estado de carga de las mismas. En (Giri & Srinivas, 2007) se forma una combinación entre un método heurístico y el algoritmo de la colonia de hormigas para formar agrupaciones óptimas de parte-máquina sobre la base de la secuencia total de la máquina que conduce a configuraciones con un mínimo número de movimientos entre las celdas. El algoritmo es usado entonces para obtener secuencias óptimas de máquinas para un flujo máximo de volumen de partes. En (Spiliopoulos & Sofianopoulo, An efficient ant colony optimization system for the manufacturing cells formation problem, 2008) se desarrolla un nuevo algoritmo de colonia de hormigas, el cual es aplicado al problema de diseño de celdas teniendo en cuenta el tamaño máximo de celdas y la secuencia de procesamiento de las partes, y evitando las dificultades asociadas con el uso tradicional de la matriz de incidencia parte-máquina. En (Solimanpur, Saeedi, & Mahdavi, 2010) se propone un algoritmo que toma en consideración la secuencia de operaciones y el volumen de producción, con el objetivo de reducir los movimientos intercelulares y el número de vacíos.

- Algoritmos Genéticos: introducidos en (Holland, 1975). Están inspirados en el proceso observado en la evolución natural de los seres vivos. Establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, codificando la información de cada solución en un vector usualmente binario a modo de cromosoma. Se introduce una función de evaluación de los cromosomas, conocida como función de adaptación o “*fitness*” la cual está basada en la función objetivo del problema. Igualmente se introduce un mecanismo de selección de manera que los cromosomas con mejor evaluación sean escogidos para “reproducirse” más a menudo que los que tienen valores menos deseados.

En (Gómez Gómez, 2007) se puede encontrar una completa revisión bibliográfica¹⁶ de las investigaciones y trabajos llevados a cabo sobre la implementación de algoritmos genéticos en la resolución del problema de formación de celdas de manufactura. En cuanto a las investigaciones que se han adelantado desde entonces, en (Wu, Chu, & Wang, 2002) se propone un algoritmo genético con un nuevo método de selección dinámico para tratar con decisiones concurrentes que envuelven objetivos altamente correlacionados y también se desarrolla un nuevo grupo de operador de mutación¹⁷ que incrementa la probabilidad de mutación, para resolver simultáneamente tanto el problema de formación de celdas como el de decisión de disposición de las máquinas. En (Zolfaghari & Liang, A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes, 2003) se toman en cuenta los tiempos de procesamiento, el tamaño de lote y la capacidad de las máquinas para el problema general de agrupación parte-máquina. También se propone un algoritmo genético en el que los parámetros de entrada son

¹⁶ Esta revisión abarca toda la década de los 90's pues es a partir de entonces que empiezan a aparecer publicaciones relacionando algoritmos genéticos con problemas de optimización combinatoria, específicamente el referido al problema de formación de celdas de manufactura.

¹⁷ Los elementos y en general toda la terminología relacionada con el esquema de operación de los algoritmos genéticos pueden ser igualmente consultados en el trabajo de grado (Gómez Gómez, 2007)

cuidadosamente ajustados usando el diseño de experimentos y la prueba ANOVA con múltiples factores. Como resultado se reportan mejoras significativas y se da a conocer la importancia de los parámetros de selección. En (Chan, Lau, & Chan, 2004) se propone un modelo matemático de múltiples objetivos para el problema de formación de celdas con enrutamiento alternativo, inclusión y eliminación de máquinas. En (Rajagopalan & Fonseca, 2005) se propone por primera vez un modelo de sensibilidad al volumen basado en un límite de volumen de producción por cada componente en lugar de usar la mezcla de productos y se implementa un modelo de algoritmo genético para mostrar que cuando el movimiento de máquinas no es viable entonces un límite de volumen puede mejorar la elección de enrutamiento óptimo de componentes. En (Filho & Tiberti, 2006) se introduce un algoritmo genético de agrupación con nuevos operadores de cruce y de mutación, esquema de corrección y un nuevo esquema de codificación de cromosomas basado en los grupos de máquinas en lugar de máquinas individuales y la metodología efectivamente parece converger más rápidamente. En (Tavakkoli-Moghaddam, Aryanezhad, Safaei, & Azaron, 2007) se asume demanda dinámica de los componentes e incertidumbre en ambientes difusos y se desarrolla un completo código de algoritmo genético para trabajar cualquier tamaño de problema propuesto. En (Gómez Gómez, 2007) se desarrolla un modelo de algoritmo genético cuya función de adaptación consiste en la reducción simultánea de los costos de mantener máquinas operando en el sistema y los costos asociados a las transferencias intercelulares de la configuración adoptada. Se realiza además un diseño de experimentos factorial 2^k con el fin de ajustar los parámetros del algoritmo con los valores que mejor rendimiento le reporta y termina haciendo una comparación de los resultados obtenidos por este método con las soluciones óptimas encontradas a través de la implementación de un método de programación lineal entera. En (Mahapatra & Sudhakara Pandian, 2008) se estudia el tiempo y la secuencia de operación de las partes para minimizar la variación de la carga celular y de los elementos

excepcionales. En (Tunnukij & Hicks, 2009) se presenta una variación del algoritmo en la cual no están pre-establecidos ni el número de celdas de manufactura ni el número de partes o máquinas por celda. El método emplea una estrategia de clasificación basada en ruleta-elitismo como una nueva estrategia para la creación sucesiva de generaciones. En (Neto & Filho, 2010) se diseña un modelo de optimización de múltiples objetivos usando en el que la evaluación de la función de adaptación es realizada a través de la simulación de sistemas celulares en los que se incorporaron efectos de congestión y se usaron políticas dinámicas de enrutamiento. Los resultados computacionales muestran mejoras en los niveles de trabajo en proceso y movimientos intercelulares mediante la reducción de inversiones en maquinaria. Finalmente, en (Deljoo, Al-e-hashem, Deljoo, & Aryanezhad, 2010) se propone un trabajo en el que se considera que las condiciones de producción dinámicas tales como la mezcla de productos, la demanda de las partes durante algún período, los movimientos de máquinas y la adición de nuevos equipos son factores que afectan los problemas de formación de celdas proporcionando flexibilidad en manufactura celular.

- Optimización del enjambre de partículas (PSO): fue introducido en (Kennedy & Eberhart, Particle Swarm Optimization, 1995). Es una técnica inspirada en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. Se fundamenta en los factores que influyen en la toma de decisión de un agente que forma parte de un conjunto de agentes similares. La toma de decisión por parte de cada agente se realiza conforme a un componente social y un componente individual, mediante los que se determina el movimiento (dirección) de este agente para alcanzar una nueva posición en el espacio de soluciones (García Nieto, 2006).

En (Andres & Lozano, 2006) el autor es el primero en adaptar el algoritmo al problema de formación de celdas de manufactura. En este trabajo la función

objetivo consiste en minimizar el número de elementos excepcionales que surgen debido a la naturaleza misma del problema. El enfoque indica que PSO tiene una gran capacidad para encontrar soluciones óptimas en cantidades de tiempo razonables. En (Ming & Ponnambalama, 2008) se propone un enfoque híbrido entre PSO y algoritmo genético, cuya metodología considera generaciones iniciales aleatorias de partículas y velocidades y fue exitosamente aplicado en la minimización de las variaciones totales de carga en las celdas y de los movimientos totales de los componentes. En (Durán, Rodríguez, & Consalter, 2008) se propone un algoritmo PSO discreto y se aplica en la computación del vector velocidad como en el algoritmo PSO tradicional. El algoritmo propuesto trabaja con una nueva noción del componente velocidad y está basado en un enfoque de probabilidades proporcionales el cual es usado en los problemas de explotación de datos. La función objetivo del problema está expresada en términos de minimización de movimientos intercelulares, es decir, el problema no toma en consideración costos ni tiempos de operación de las partes a fabricar. Se realizan una serie de experimentos que demuestran que el algoritmo es estable y presenta baja variabilidad. En (Mehdizadeh & Tavakkoli-Moghaddam, 2009) se propone una técnica PSO difusa en el contexto de la agrupación parte-máquina en donde cada partícula corresponde a un vector central de agrupación y el enjambre representa un número de agrupación de candidatos para el actual vector de datos. En (Caprihan, Slomp, Gusaran, & Agarwal, 2009) se establece un método cuántico de PSO y se diseña un sistema virtual de manufactura celular. Este método fue probado con algoritmo genético encontrándose que PSO consume menos tiempo de CPU y provee mejores soluciones. En (Anvari, Mehrabad, & Banzinpour, 2010) se realiza un estudio en el que se da a conocer una técnica híbrida entre algoritmo PSO y algoritmo genético. Para ello se utilizó un operador de mutación que fue incluido en la ecuación de actualización de la velocidad con el fin de evitar llegar a soluciones óptimas locales.

2.4. ALGORITMO PSO

El algoritmo de Optimización del Enjambre de partículas, PSO por sus siglas en inglés *Particle Swarm optimization* es un modelo meta-heurístico que se clasifica dentro de las técnicas de optimización inteligente y que está catalogado como un algoritmo de optimización estocástica basado en poblaciones. Fue introducido en 1995 por el psicólogo-sociólogo James Kennedy y por el Ingeniero electrónico Russell Eberhart. Desde entonces son numerosas las mejoras y variaciones que se han introducido al algoritmo con el fin de mejorar su rendimiento y adaptarlo mejor al problema particular al cual se aplica.

El funcionamiento del algoritmo simula el siguiente comportamiento social: un grupo de aves buscan aleatoriamente comida en el área. Solo hay una ración de comida en el área que está siendo explorada. Las aves no saben en dónde está la comida, pero saben qué tan lejos está en cada iteración. ¿Cuál es entonces la mejor estrategia para encontrar la comida? La solución efectiva es seguir al ave que esté más cerca de la comida.

Las ventajas atribuidas a los algoritmos de optimización inteligente radica en que pueden explorar el espacio de la solución en múltiples direcciones simultáneamente, haciéndolos adecuados para problemas con gran número de variables donde el espacio de la solución es de gran tamaño como para realizar una búsqueda exhaustiva en tiempos razonables. Adicionalmente, este tipo de algoritmos no requieren de un conocimiento detallado de la estructura y comportamiento del sistema a modelar pues en lugar de usar información específica para guiar la búsqueda, este algoritmo se vale de cambios aleatorios en las soluciones candidatas.

En el algoritmo PSO, el objetivo del cúmulo es explorar el espacio de búsqueda hasta llegar a su objetivo, representado por la mejor posición en el espacio; por lo

tanto cada partícula, o de manera más estricta su posición, se considera una solución candidata (Kennedy, Eberhart, & Shi, Swarm Intelligence, 2001). Para evaluar la bondad de la posición de una partícula se define una función de adaptación (*fitness*), la cual debe tener la suficiente información para cuantificar qué tan buena es con respecto al objetivo y las restricciones propias del problema. Cada partícula es una solución cooperativa, en el sentido en que su posición es el resultado de intercambios de información con las partículas vecinas del cúmulo durante la ejecución del algoritmo (Casallas Fonseca & Vargas Aristizábal).

2.4.1. Descripción del algoritmo

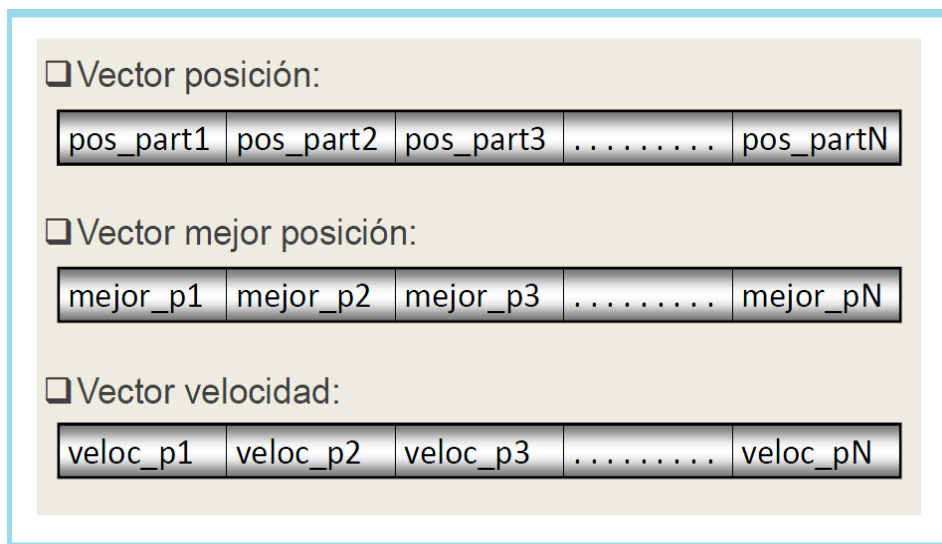
Un algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas. La posición de cada partícula representa una solución potencial al problema evaluado. En general, cada partícula p_i está compuesta por cinco elementos principales: tres vectores y dos valores de fitness (García Nieto, 2006):

- El vector $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- El vector $pBest_i = (p_{i1}, p_{i2}, \dots, p_{in})$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ almacena el gradiente (dirección) según el cual se moverá la partícula.
- El valor de fitness $fitness_{x_i}$ almacena el valor de adecuación de la solución actual (vector x_i).
- El valor de fitness $fitness_{pBest_i}$ almacena el valor de adecuación de la mejor solución local encontrada hasta el momento (vector $pBest_i$).

El cúmulo se inicializa generando las posiciones y las velocidades iniciales de las partículas: las posiciones en el espacio de búsqueda se pueden generar bien sea

aleatoriamente, con ayuda de un heurístico de construcción o a través de una combinación de ambas formas. Una vez generadas las posiciones, se calcula el fitness de cada una y se actualizan los valores de $fitness_{x_i}$ y $fitness_{pBest_i}$. Por su parte las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-V_{max}; V_{max}]$, donde V_{max} indica la velocidad máxima que puede tomar una partícula en cada movimiento.

Figura 1: Componentes de una partícula PSO



Fuente: (Garro Retamal & Domínguez Cuadrado)

Inicializado el cúmulo, las partículas se deben mover dentro del proceso iterativo. Una partícula se mueve desde una posición del espacio de búsqueda hasta otra, simplemente, añadiendo al vector posición x_i el vector velocidad v_i para obtener un nuevo vector posición. Una vez calculada la nueva posición de la partícula, se evalúa actualizando $fitness_{x_i}$. Además, si el nuevo fitness es el mejor valor de adaptación encontrado hasta el momento, se deben actualizar los valores $pBest_i$ y $fitness_{pBest}$. El vector velocidad de cada partícula es modificado en cada iteración utilizando la velocidad anterior, un componente cognitivo y un

componente social¹⁸. El modelo matemático resultante y que representa el corazón del algoritmo PSO está expresado en términos de las siguientes ecuaciones:

$$v_i^{k+1} = w * v_i^k + c_1 * rand_1 * (pBest_i - x_i^k) + c_2 * rand_2 * (p_{gi} - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Donde:

- v_i^k : velocidad de la partícula i en la iteración k
- w : factor de inercia¹⁹
- c_1, c_2 : constantes de aceleración que controlan los componentes cognitivo y social²⁰
- $rand_1, rand_2$: números reales aleatorios con distribución uniforme entre 0 y 1
- x_i^k : posición actual de la partícula i en la iteración k
- $pBest_i$: mejor posición (solución) encontrada por la partícula i hasta el momento
- p_{gi} : representa la posición de la partícula con el mejor $pBest_fitness$ del entorno de p_i (lBest o localbest) o de todo el cúmulo (gBest o globalbest).

La primera ecuación (1) refleja la actualización del vector velocidad de cada partícula i en cada iteración k . El componente cognitivo está modelado por el factor ' $c_1 * rand_1 * (pBest_i - x_i^k)$ ' y representa la distancia entre la posición actual y la mejor conocida por esa partícula, esto es, la decisión que tomará la partícula

¹⁸ El componente cognitivo permite que la partícula se comporte como un agente aislado y dependiente exclusivamente de su experiencia mientras que el componente social hace que cada partícula ignore su propia experiencia ajustándose a la experiencia exitosa de sus vecinas.

¹⁹ Regula la influencia de la velocidad anterior de la partícula v_i en el cálculo de la nueva velocidad v_i^{k+1} , a manera de regular el vuelo de la partícula haciendo un balance entre explotación (intensificación) y exploración (diversificación) del espacio de búsqueda. Controla el excesivo crecimiento de las velocidades, lo cual es un fenómeno natural que sucede durante la ejecución del algoritmo.

²⁰ Los valores asignados a la constante de aceleración c_1 regulan la influencia que tiene la mejor posición alcanzada por la partícula ($pBest$) para guiar su nueva dirección, en tanto que los valores asociados a la constante de aceleración c_2 regulan la influencia del líder del cúmulo (p_g) en la dirección de búsqueda de la partícula.

influenciada por su propia experiencia a lo largo de su vida. El componente social está modelado por ' $c_2 * rand_2 * (p_{gi} - x_i^k)$ ' y representa la distancia entre la posición actual y la mejor posición del vecindario, es decir, la decisión que tomará la partícula según la influencia que el resto del cúmulo ejerce sobre ella. Por su parte, la segunda ecuación (2) modela el movimiento de cada partícula i en cada iteración k .

En la Figura 2 se presenta el pseudo-código clásico que caracteriza al algoritmo básico PSO.

Figura 2: Pseudo-código PSO clásico

```

S ← Inicializar Cúmulo ()
while no se alcance la condición de parada do
  for i = 1 to size(S) do
    evaluar cada partícula  $x_i$  del cúmulo  $S^{22}$ 
    if  $fitness(x_i)$  es mejor que  $fitness(pBest_i)$  then
       $pBest_i \leftarrow x_i$ ,  $fitness(pBest_i) \leftarrow fitness(x_i)$ 
    end if
    if  $fitness(pBest_i)$  es mejor que  $fitness(gBest)$  then
       $gBest \leftarrow pBest_i$ ,  $fitness(gBest) \leftarrow fitness(pBest_i)$ 
    end if
  end for
  for i = 1 to size(S) do
     $v_i \leftarrow w * v_i + c_1 * rand_1 * (pBest_i - x_i) + c_2 * rand_2 * (gBest - x_i)$ 
     $x_i \leftarrow x_i + v_i$ 
  end for
end while
Salida: la mejor solución encontrada

```

Fuente: (García Nieto, 2006)

2.4.2. Tipos de algoritmo PSO

Los algoritmos PSO se pueden clasificar atendiendo a dos criterios diferentes: según la importancia de los pesos cognitivo y social y según el tipo de vecindario utilizado.

Dependiendo de la influencia de los factores cognitivo y social (constantes c_1 y c_2 respectivamente) sobre la dirección de velocidad que toman las partículas en (Kennedy, *The Particle Swarm: Social Adaptation of Knowledge*, 1997) se identifican cuatro tipos de algoritmos PSO:

- Modelo completo: $c_1, c_2 > 0$. Tanto el componente cognitivo como el social intervienen en el movimiento.
- Modelo sólo cognitivo: $c_1 > 0$ y $c_2 = 0$. Únicamente el componente cognitivo interviene en el movimiento.
- Modelo sólo social: $c_1 = 0$ y $c_2 > 0$. Únicamente el componente social interviene en el movimiento.
- Modelo sólo social exclusivo: $c_1 = 0, c_2 > 0$ y $p_{gi} \neq x_i$. La posición de la partícula en sí no puede ser la mejor de su entorno.

Estudios realizados sobre la influencia de los coeficientes de aceleración recomiendan valores de $c_1 = c_2 = 2$ (Kennedy, Eberhart, & Shi, *Swarm Intelligence*, 2001)²¹.

Por otra parte, desde el punto de vista del vecindario, es decir, la cantidad y posición de las partículas que intervienen en el cálculo de la distancia en la componente social, se identifican dos tipos de algoritmos (Flores Mendoza & Mezura Montes, 2008):

²¹ De hecho, en el paquete computacional propio de este algoritmo los parámetros c_1 y c_2 asumen por defecto valores iguales a 2.

- PSO local: para esta variante el cúmulo completo se divide en un número n de vecindarios con igual número de individuos, y en cada vecindario existe un líder local el cual influencia el vuelo de las partículas dentro de su vecindario. Esto quiere decir que las partículas sólo se comunican con sus vecinas e ignoran las posiciones de las demás²².
- PSO global: existe un solo líder en el cúmulo y las partículas se relacionan socialmente con todos los individuos del cúmulo debido a la influencia de un líder global. Esto implica que cada partícula se comunica con todas las demás en el cúmulo, pues puede saber quién es la mejor de entre todas ellas²³.

2.4.3. Toolbox PSO para Matlab

Para la programación de los algoritmos a comparar en el presente trabajo de grado se ha seleccionado el lenguaje de programación Matlab, el cual es un programa de cálculo numérico orientado a matrices, muy adecuado al problema a resolver dado que tanto la captura de datos de entrada así como una de las variables de respuesta a obtener (configuración “óptima” de las celdas a conformar) están naturalmente agrupados en vectores o matrices, según sea el dato.

Para la implementación del algoritmo PSO es necesario instalar la herramienta (toolbox) PSOT para entorno Matlab. Esta herramienta es un optimizador de enjambre de partículas genérico para encontrar el mínimo o máximo de cualquier función Matlab. Permite implementar cuatro tipos diferentes de PSO: PSO

²² A este tipo de algoritmos le cuesta más converger favoreciendo así la diversificación, pero no cae fácilmente en óptimos locales.

²³ Esta versión converge más rápido pues la visibilidad de cada partícula es mejor y se acercan más a la mejor del cúmulo favoreciendo la intensificación. Por esta razón, también cae más fácilmente en óptimos locales.

Common, Trelea tipo 1, Trelea tipo 2 y Clerc Clase 1²⁴. Cualquiera que sea el tipo de PSO elegido, se trata de una versión vectorizada que remueve el ciclo *for* asociado con el número de partículas. También requiere que la función objetivo tenga una sola entrada (denominada 'in')²⁵ que representa todas las dimensiones de búsqueda.

La herramienta funciona básicamente a través de dos funciones en código Matlab. La primera de ellas se denomina "parámetrospso.m" y a través de ella se introducen los valores de los parámetros del algoritmo. En la Figura 3 está representado el esqueleto de esta función principal. En ella puede verse que en la última línea se hace el llamado a una función denominada "pso_trelea_vectorized.m", la segunda de las funciones que componen el algoritmo, de hecho el corazón del mismo. En esta última función se describe el funcionamiento detallado del algoritmo, basado en los parámetros de entrada establecidos y en la función objetivo²⁶ a minimizar o maximizar, según sea el caso.

A continuación se hace una breve descripción del modo de uso de la función principal:

ENTRADAS:

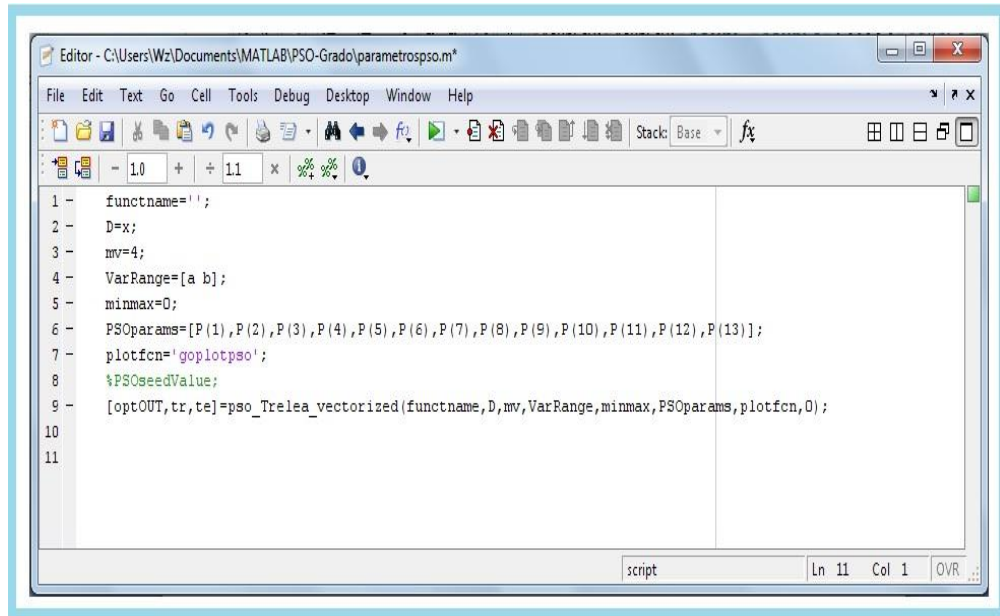
- functname: nombre del archivo de Matlab que contiene la función a optimizar
- D: número de entradas de la función; dimensión del problema; número de componentes de cada partícula

²⁴ La diferencia de los algoritmos radica en la fórmula de actualización de velocidad de las partículas

²⁵ En el apartado 4.3.2 se explica con mayor detalle a qué se refiere este argumento de entrada en la función objetivo a minimizar para el problema específico de formación de celdas de manufactura.

²⁶ La programación de la función objetivo corresponde a un fichero diferente, programado por el usuario según la formulación matemática propia del tipo de problema a optimizar. Este fichero es llamado por la función "pso_trelea_vectorized.m", y retorna a ella el valor fitness de las partículas por cada iteración para que el algoritmo continúe con su ejecución hasta encontrar la solución adecuada.

Figura 3: Función principal Toolbox PSOt



```
1 - functname='';
2 - D=x;
3 - mv=4;
4 - VarRange=[a b];
5 - minmax=0;
6 - PSOparams=[P(1),P(2),P(3),P(4),P(5),P(6),P(7),P(8),P(9),P(10),P(11),P(12),P(13)];
7 - plotfcn='goplotpso';
8 - %PSOseedValue;
9 - [optOUT,tr,te]=psot_Trelea_vectorized(functname,D,mv,VarRange,minmax,PSOparams,plotfcn,0);
10
11
```

Fuente: Autor

ENTRADAS OPCIONALES:

- mv: máxima velocidad de las partículas. Puede ser un escalar o un vector de longitud D, lo cual permite que cada componente tenga su propia velocidad máxima. Valor por defecto = 4
- VarRange: matriz de rangos para cada variable de entrada, por defecto de -100 a 100, de la forma: [min1 max1; min2 max2; ... ; minD maxD]
- minmax =0, la función es minimizada (por defecto)
 - 1, la función es maximizada
 - 2, la función es enfocada a P(12) (minimiza distancia a error deseado)

PARÁMETROS PSO

- P(1): épocas entre actualizaciones de pantalla, por defecto = 100

SALIDAS OPCIONALES

- tr: Gbest de cada iteración, sigue el vuelo del enjambre
- tv: épocas alcanzadas. Retorna como un vector desde 1 hasta el número final de épocas

De lo anterior se deduce que existen dos criterios de parada del algoritmo, correspondientes al máximo número de iteraciones que el algoritmo ejecuta y al requerimiento de mínimo error establecido.

3. FORMULACIÓN MATEMÁTICA GENERAL DEL PROBLEMA DE FORMACIÓN DE CELDAS DE MANUFACTURA

La formulación matemática general concerniente al problema de formación de celdas de manufactura fue tomada de (Gómez Gómez, 2007), quien a su vez se basó en los trabajos y publicaciones hechas por autores como Vakharia y Chang; Mungwatanna y Ahmed, Tavakkoli y Safeei²⁸.

Basado en una estrategia de agrupamiento simultáneo, se asigna individualmente cada operación a una determinada celda y se calculan los requerimientos de máquina que corresponden a la satisfacción de su demanda. Con estos requerimientos se calcula el costo total de máquinas, y adicionalmente con la asignación de las operaciones a las celdas se calcula el número de transferencias intercelulares necesarias. Con este número de transferencias se calcula el costo de transferir productos entre celdas y al ser sumado con el costo total de máquinas se obtiene el costo total del diseño de celdas de manufactura.

3.1. SUPUESTOS

- Los parámetros conocidos como datos de entrada son constantes y conocidos.
- La totalidad de la demanda debe ser satisfecha a lo largo del período.
- El tiempo de producción de un lote de productos por operación por tipo de producto incluye el tiempo de alistamiento de máquina en caso de haberlo.
- El transporte intercelular de productos se hace por lotes y no requiere tiempo adicional.
- El costo de los movimientos intercelulares no depende de la distancia recorrida.

²⁸ Se toma la formulación descrita por (Gómez Gómez, 2007) y no la de los autores en los que se basó para la misma debido a que introduce una importante modificación en la formulación de la función objetivo que no había sido tenida en cuenta por los autores de referencia.

- Las máquinas de cada tipo se consideran réplicas con idéntica capacidad e idéntico costo.
- Las máquinas están disponibles desde el comienzo del período dentro del sistema.
- Cada máquina dentro del sistema solo puede procesar un producto a la vez.
- La capacidad de las máquinas en el sistema es cien por ciento confiable.
- No se consideran productos defectuosos ni paradas de máquinas.
- La operación del sistema no admite re-procesos de productos.
- No se consideran tiempos de espera en producción.
- No se considera mantenimiento de inventarios.
- No se considera el valor del dinero en el tiempo a lo largo del período.

3.2. DATOS DE ENTRADA

Sin ellos es imposible determinar una configuración celular adecuada que minimice los costos de operación del sistema productivo.

- Mezcla de productos a fabricar
- Demanda por tipo de producto en número de lotes por período
- Secuencia de operación de cada producto, incluyendo el tipo de máquina en la cual se realiza
- Tiempo de operación requerido para fabricar un lote de cada producto de la mezcla de productos
- Capacidad de máquina en términos de tiempo disponible por tipo de máquina
- Costo de máquina por tipo de máquina²⁹
- Costo de transferencias intercelulares
- Tamaño mínimo de celdas, es decir, número mínimo de máquinas que debe contener la celda de fabricación

²⁹ Puede hacer referencia al valor de depreciación o bien al valor de arrendamiento de las máquinas. Incluye costos como el de mantenimiento, consumo de energía, costo de herramientas y utilería y demás.

- Tamaño máximo de celdas, esto es, número máximo permitido de máquinas por celda en el sistema

3.3. DECISIONES DE DISEÑO

- Asignar las operaciones de cada producto a las celdas
- Determinar el número de máquinas necesarias para la ejecución de dicha asignación de operaciones

3.4. OBJETIVOS DE DISEÑO

El enfoque de resolución del problema de formación de celdas de manufactura para este trabajo de grado está expresado en términos de minimización de costos puesto que éstos constituyen una buena medida de qué tan eficiente es un sistema productivo en su operación del día a día.

- Costo de máquina: costo por período de tener máquinas operando en el sistema. Se calcula multiplicando el costo unitario por el número total de máquinas de cada tipo dedicadas a la producción de la mezcla de productos.
- Costo de transferencias intercelulares³⁰: se determina multiplicando el número de lotes de cada tipo de producto a transferir por el costo de transportar un lote de producto entre cualquier par de celdas.

3.5. RESTRICCIONES

- La demanda de la mezcla de productos establecida para el período debe ser satisfecha en su totalidad por el sistema de manufactura celular.

³⁰ Las transferencias intercelulares ocurren cuando determinados productos, en este caso lotes de productos, no pueden ser completamente manufacturados en la celda a la cual fueron asignados inicialmente, puesto que requieren de un trabajo realizado por una máquina que pertenece a una celda de manufactura diferente.

- Se debe especificar el tamaño deseado de las celdas en términos de número de máquinas. Pueden usarse valores máximos y mínimos en lugar de un número específico.
- Se debe especificar el número de celdas a conformar en el sistema.

3.6. NOTACIONES

3.6.1. Significado de los índices utilizados en la formulación

- c : índice para las celdas de manufactura ($c=1,2,\dots,C$)
- m : índice para los tipos de máquinas en el sistema ($m=1,2,\dots,M$)
- p : índice para los productos o tipos de producto a fabricar ($p=1,2,\dots,P$)
- j : índice para las operaciones requeridas por el producto p ($j=1,\dots, Op$)

3.6.2. Para los datos de entrada

- D_p : demanda del producto p para el período. Vector fila de tamaño P
- O_p : número de operaciones necesarias para la fabricación del producto p . Vector fila de tamaño P
- M_{jp} : matriz de incidencia parte-máquina de tamaño $O_p \times P$. Contiene números entre 1 y M que representan el tipo de máquina en la que debe realizarse cada operación de cada producto
- Top_{jp} : matriz de tiempos de operación de tamaño $O_p \times P$. Representa el tiempo requerido para completar cada operación de cada producto.
- CAP_m : capacidad disponible de una máquina tipo m durante el período. Vector fila de tamaño M .
- CM_m : costo de tener una máquina tipo m durante el período. Vector columna de tamaño M
- CTI : costo de transferir un lote de producto entre cualquier par de celdas
- $TMin$: número mínimo de máquinas por celda

- TMax: número máximo de máquinas por celda

3.6.3. Para las variables de decisión

- AsignacionMaq_Celda_{mc}: matriz de tamaño M x C que contiene el número de máquinas tipo m a asignar a la celda c durante el período
- MatrizAsignación_{jpc}: matriz de tamaño Op x P con: 1 si la operación j del producto p es asignada a la celda c; 0 de lo contrario.

3.7. FUNCIÓN OBJETIVO

Minimizar

$$\text{COSTO} = \sum_{c=1}^C \sum_{m=1}^M \text{AsignacionMaq_Celda}_{mc} * CM_m + \text{CTI} \left[\sum_{p=1}^P Dp \left(\sum_{j=1}^{Op-1} \sum_{c=1}^C | \text{MatrizAsignacion}_{(j+1)pc} - \text{MatrizAsignacion}_{jpc} | \right) \right] \quad (3)$$

Sujeto a:

$$\sum_{c=1}^C \text{MatrizAsignacion}_{jpc} = 1 \quad \text{Para todo } j, p \quad (4)$$

$$\sum_{p=1}^P \sum_{j=1}^{Op} Dp * Top_{jp} * M_{jp} * \text{MatrizAsignacion}_{jpc} \leq \text{CAP}_m * \text{AsignacionMaq_Celda}_{mc} \quad \text{Para todo } m, c \quad (5)$$

$$\sum_{m=1}^M \text{AsignacionMaq_Celda}_{mc} \geq \text{TMin} \quad \text{Para todo } c \quad (6)$$

$$\sum_{m=1}^M \text{AsignacionMaq_Celda}_{mc} \leq \text{TMax} \quad \text{Para todo } c \quad (7)$$

$$\text{AsignacionMaq_Celda}_{mc} \geq 0 \quad (\text{Entero}) \quad \text{Para todo } m, c \quad (8)$$

$$\text{MatrizAsignacion}_{jpc} \in [0,1] \quad (\text{Binario}) \quad \text{Para todo } j, p, c \quad (9)$$

En la ecuación de la función objetivo (3), el primer término calcula el costo total de tener máquinas operando en el sistema y se obtiene mediante la suma del número de máquinas de cada tipo en cada celda con sus respectivos costos. El segundo término representa el costo total de las transferencias intercelulares de lotes de producto y se calcula multiplicando el número total de lotes transferidos por el costo de mover un lote de cualquier tipo de producto entre cualquier par de celdas.

En cuanto a las ecuaciones del conjunto de restricciones, en (4) se limita la asignación de cada operación de cada tipo de producto a una celda de manufactura; en (5) se garantiza que no se excedan las capacidades de cada tipo de máquina y de cada celda y que se puedan satisfacer los requerimientos de la demanda; en (6) y (7) se asegura que el número de máquinas por cada celda no sea menor al valor establecido como mínimo ni superior al establecido como máximo; por último, en (8) y (9) se señala la naturaleza de las variables de decisión dentro del modelo.

4. PROGRAMACIÓN DEL ALGORITMO PSO PARA EL DISEÑO DE SISTEMAS DE MANUFACTURA CELULAR

Para el presente trabajo de investigación se utilizó un algoritmo PSO completo y global³¹. Para su implementación se utilizó la toolbox PSOt anteriormente descrita, desarrollada por Brian Birge el 22 de abril del 2005. Esta herramienta puede ser descargada desde la dirección en internet:

<http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox>. Corresponde a la versión actualizada el 20 de marzo del 2006. Sin embargo, fue necesario adaptar el algoritmo programado en esta herramienta al problema de formación de celdas de manufactura.

4.1. VARIABLES Y PARÁMETROS DE ENTRADA

Corresponden a los valores que se deben establecer en la función principal del algoritmo (parametrospso.m)

✓ D:

Cada partícula, más estrictamente cada posición que asume, se definió como una matriz de tamaño $O_p \times P$, siendo O_p el mayor número de operaciones requeridas para alguno de los productos y P el número total de productos de la mezcla. Esto quiere decir que las partículas se componen de una sola dimensión, pero esta dimensión corresponde a un arreglo matricial. La herramienta PSOt viene diseñada para trabajar con valores escalares de D , por lo que fue necesario hacer una importante modificación al código de las dos funciones del algoritmo, de manera que aceptara trabajar y hacer cálculos y operaciones con matrices como

³¹ Remitirse al apartado 2.4.2

datos de entrada y a su vez retornara como solución una variable, o las que fueran necesarias, en este mismo tipo de configuración matricial.

En ese orden de ideas, fue necesario modificar la estructura de la velocidad de las partículas: de los valores rand1 y rand2 necesarios para el cálculo de las nuevas velocidades y de los vectores posición (pos), mejor posición (pbest) y velocidad (vel) para cada partícula³²; de la matriz 'bestpos'³³ a la cual además fue necesario agregarle nuevas columnas para guardar datos propios del problema de formación de celdas de manufactura. La representación de cada partícula se presenta en la Figura 4; las variables mencionadas arriba adoptan esta misma configuración luego de la modificación realizada al algoritmo.

Figura 4: Configuración de cada partícula

		Producto (p)			
		1	2	...	P
Operación (i)	1				
	2				
	...				
	...				
	Op				

Fuente: Autor

✓ funcname:

El nombre de la función objetivo a optimizar se programó en el archivo denominado "OBJETIVO_INDIVIDUO.m", el cual será explicado más adelante.

³² Originalmente estas variables correspondían a vectores de números escalares. Con el ajuste efectuado siguieron siendo vectores pero ya no de escalares sino de matrices.

³³ En el algoritmo original corresponde a una matriz con número de filas igual al número de iteraciones definidas y con número de columnas igual al número de dimensiones D+1. En ella se almacena el mejor valor de cada dimensión por iteración junto con el mejor valor de aptitud encontrado por las partículas durante cada iteración (gbest).

✓ VarRange:

Como se estableció anteriormente, cada partícula se compone de una sola dimensión. El rango de posibles valores que puede adoptar se estableció en $[0.0001 C]^{34}$. Es decir que la matriz $Op \times P$ que define la dimensión de la partícula contiene valores entre este rango, valores que representan la celda a la cual es asignada cada operación de cada producto.

✓ Parámetros de PSO

Inicialmente se tomaron los valores asignados por defecto. Más adelante, en el numeral 5 correspondiente al diseño de experimentos se ajustan los valores de los parámetros principales de modo que el algoritmo logre el mejor rendimiento posible.

4.2. SALIDAS

✓ OptOUT:

Se configuró como un arreglo de celdas³⁵ de una fila y 4 columnas que almacenan los resultados “óptimos” arrojados por el algoritmo:

³⁴ Debido a la naturaleza misma de las ecuaciones del algoritmo PSO, tanto las posiciones iniciales como las posiciones actualizadas al aplicar el vector velocidad adoptan valores decimales dentro del rango establecido. Uno de los tantos ajustes realizados sobre el mismo consistió en la introducción de una línea de comando dentro del código que aproxima todos los valores al entero superior, de manera que la matriz que representa a las partículas se llena únicamente con los números enteros comprendidos en el rango.

³⁵ Colección de contenedores llamados *celdas* en los cuales se pueden almacenar diferentes tipos de datos (escalares, vectores, matrices, hipermatrices, cadenas de caracteres, etc)

- Columna1 -> 'gbest': mejor posición hallada. Matriz de tamaño Op x P con la mejor configuración de celdas encontrada por el enjambre (aquella que genera el mínimo costo)
- Columna2 -> 'gbestval': mínimo costo encontrado. Escalar con el valor de la mejor aptitud hallada por el enjambre, valor calculado con base en la mejor posición encontrada
- Columna3 -> 'bestAsignacionMaq_Celda': necesidades de máquina resultante de la mejor configuración de celdas o mejor posición encontrada. Corresponde a una matriz de tamaño M x C
- Columna4 -> 'EstadoRestriccion': variable binaria asociada con las necesidades de máquina resultantes. Toma el valor de 0 si las necesidades de máquina, y por consiguiente la partícula, corresponde a una solución no factible dado que viola la restricción de tamaños de celda establecida. De lo contrario toma el valor de 1.

✓ Datos:

Variable que guarda la información almacenada en el arreglo de celdas 'bestpos'. Este arreglo contiene la misma información que la variable 'optOUT' pero no solo de las mejores soluciones de todo el algoritmo, sino de la mejor solución encontrada en cada iteración, así que corresponde a un arreglo de celdas de filas igual al número de iteraciones programadas y con el mismo número y significado de columnas descritas en la variable de salida anterior.

4.3. FUNCIONES DE PROGRAMACIÓN DEL ALGORITMO PSO

A continuación se hace una descripción general de la información contenida en las tres funciones, cada una con su respectivo código en lenguaje Matlab, utilizadas para la programación del algoritmo. Dos de ellas corresponden a las incluidas en

la toolbox PSOt. En la última de las funciones se programó la función fitness objeto de minimización por parte del algoritmo.

Los códigos de programación completos de las tres funciones pueden ser consultados en los anexos A, B y C.

4.3.1. Función principal 'parametrospsom'

Como se describió en el marco teórico conceptual, en esta función se introducen los valores de los parámetros del algoritmo con los que se va a trabajar. De estos valores ya se hizo una descripción en el apartado inmediatamente anterior.

4.3.2. Función 'pso_trelea_vectorized.m'

Representa el corazón del algoritmo pues en ella está programado todo su funcionamiento y el comportamiento social de la naturaleza que trata de imitar. A continuación se describen los pasos principales del código de programación que llevan al algoritmo a encontrar la mejor solución al problema de formación de celdas de manufactura configurado:

1. Crear una matriz máscara de tamaño $Op \times P$ (matriz máscara de operaciones) con 1's en las casillas donde cada producto P tiene operaciones; 0's en donde no
2. Crear dos matrices máscara de tamaño $Op \times P$ (matrices máscara de velocidad) para controlar la velocidad de las partículas: una llena con los valores mv y otra llena con los valores $-mv$
3. Crear dos matrices máscara de tamaño $Op \times P$ (matrices máscara de posición) para controlar los rangos de posibles valores de las partículas: una con el límite inferior del intervalo definido en 'VarRange' y otra con el límite superior

4. Construir las posiciones iniciales aleatorias: crear un vector columna de matrices (vector posiciones), de tamaño igual al de la población, en el que cada matriz representa una partícula del cúmulo. Se llenan con números aleatorios pertenecientes al rango 'VarRange', estos números se aproximan al entero superior más cercano y cada matriz es multiplicada por la matriz máscara de operaciones con el fin de eliminar valores en las casillas donde no hay asignada ninguna operación
5. Construir las velocidades iniciales aleatorias: crear un vector columna de matrices (vector velocidades), una para cada partícula, con números aleatorios entre $-mv$ y mv , igualmente multiplicadas por la matriz máscara de operaciones

Los pasos listados a continuación, desde el 6 hasta el 11, solo se ejecutan una vez durante todo el algoritmo pues corresponden a la inicialización de la población y de variables necesarias para almacenar datos del problema de formación de celdas a través de toda la corrida del algoritmo:

6. Crear el vector de matrices con las mejores posiciones encontradas por las partículas (vector pbest). Inicialmente, dado que no se ha realizado ninguna iteración, este vector asume los mismos valores que las posiciones iniciales construidas en el numeral 4
7. Enviar el vector posiciones (a través de la variable denominada 'in') a la función 'OBJETIVO_INDIVIDUO' para que evalúe cada partícula, retornando un vector de números escalares con los valores del costo asociado a la configuración particular de cada partícula, un valor por cada una. Este vector retorna a la función 'pso_trelea_vectorized' en una variable denominada 'out'
8. Crear el vector de valores de las mejores posiciones (vector pbestval), es decir un vector de escalares que contenga el mejor valor de la función objetivo de cada partícula. Inicialmente este vector equivale al retornado por la función objetivo, out

9. Encontrar el mínimo valor en el vector `pbestval`, junto con el número de la partícula asociado a él, y almacenarlo en las variables `gbestval` y `idx1` respectivamente. Adicionalmente guardar tanto las necesidades de máquina como la información de si la partícula `idx1` es una solución factible, en las variables `bestAsignacionMaq_Celda` y `EstadoRestriccion` respectivamente
10. Crear el arreglo de celdas `bestpos` de tamaño igual al número de iteraciones, en el cual serán almacenados los valores `gbest`, `gbestval`, `bestAsignacionMaq_Celda` y `EstadoRestriccion` descritos en los pasos anteriores y en el apartado 4.2
11. Crear la variable `gbest` la cual almacenará la mejor configuración de celdas encontrada por el cúmulo. En este caso, esta configuración corresponderá a la de la partícula `idx1` del vector `pbest`

Los pasos descritos a continuación se ejecutarán tantas veces como iteraciones se hayan definido en los parámetros del algoritmo

12. Construir dos vectores columna de matrices de tamaño $O_p \times P$ (vector de números aleatorios 1 y 2 respectivamente), una matriz de cada vector por cada partícula, con números aleatorios que corresponden a los términos `rand1` y `rand2` de la fórmula de la velocidad de las partículas propia del algoritmo PSO clásico
13. Multiplicar las matrices de los vectores de números aleatorios por los términos `c1` y `c2`, respectivamente, definidos en los parámetros de la función `parametrosps0`
14. Con los resultados del numeral anterior y los demás datos necesarios, aplicar la fórmula de la velocidad a cada una de las matrices que componen el vector velocidades previamente creado
15. A las matrices de velocidades calculadas en el numeral anterior multiplicar únicamente los valores que exceden el rango de velocidad permitida ($[-mv \quad mv]$) por las máscaras de velocidades

16. Actualizar las posiciones de las partículas aplicando la fórmula del algoritmo (ecuación (2), apartado 2.4.1) A este resultado, nuevo vector posiciones, multiplicarlo por la matriz máscara de operaciones
17. Ajustar los valores del nuevo vector posiciones por medio de las máscaras de posición, de manera que los valores resultantes luego de actualizar las posiciones no excedan los límites del intervalo establecido en VarRange
18. Enviar el nuevo vector posiciones a la función objetivo para ser evaluado, pero esta vez incluyendo al final, como última fila, la variable gbest. Esto debido a que durante todo el algoritmo las partículas deben tener la información acerca de cuál ha sido la mejor solución encontrada hasta el momento por todo el cúmulo
19. Una vez retornados los valores de la función objetivo (vector 'out') almacenar los valores gbest, gbestval, bestAsignacionMaq_Celda y EstadoRestriccion correspondientes a la iteración en el arreglo 'bestpos'
20. Encontrar el número de las partículas en el vector pbestval cuyo valor fitness sea mayor o igual a los del nuevo vector out y reemplazarlos por los valores del mismo. De la misma manera, reemplazar en el vector 'pbest' la configuración de estas partículas con la configuración correspondiente contenida en el nuevo vector posiciones
21. Con la información del vector pbestval actualizada, encontrar el mínimo de entre todos los valores que contiene. Si este valor mínimo, que corresponde al mínimo valor de la función objetivo de la iteración, es menor al mínimo valor encontrado hasta el momento (gbestval), hacer la actualización de gbestval, gbest, bestAsignacionMaq_Celda y EstadoRestriccion correspondiente
22. Volver al paso 12 hasta terminar el número de iteraciones programadas

En la Figura 5 se muestra el diagrama de flujo con los principales pasos del procedimiento descrito en este apartado.

4.3.3. Función objetivo 'OBJETIVO_INDIVIDUO.m'

Esta función es alimentada con la configuración de las partículas contenidas en el vector posiciones. Contiene además todos los datos de entrada del problema necesarios para el cálculo del valor de la función objetivo de las partículas. Es una función principal que contiene a su vez a tres sub-funciones denominadas 'NECESIDADES DE MAQUINA', 'COSTO DE MAQUINA' y 'COSTOS INTERCELULARES'

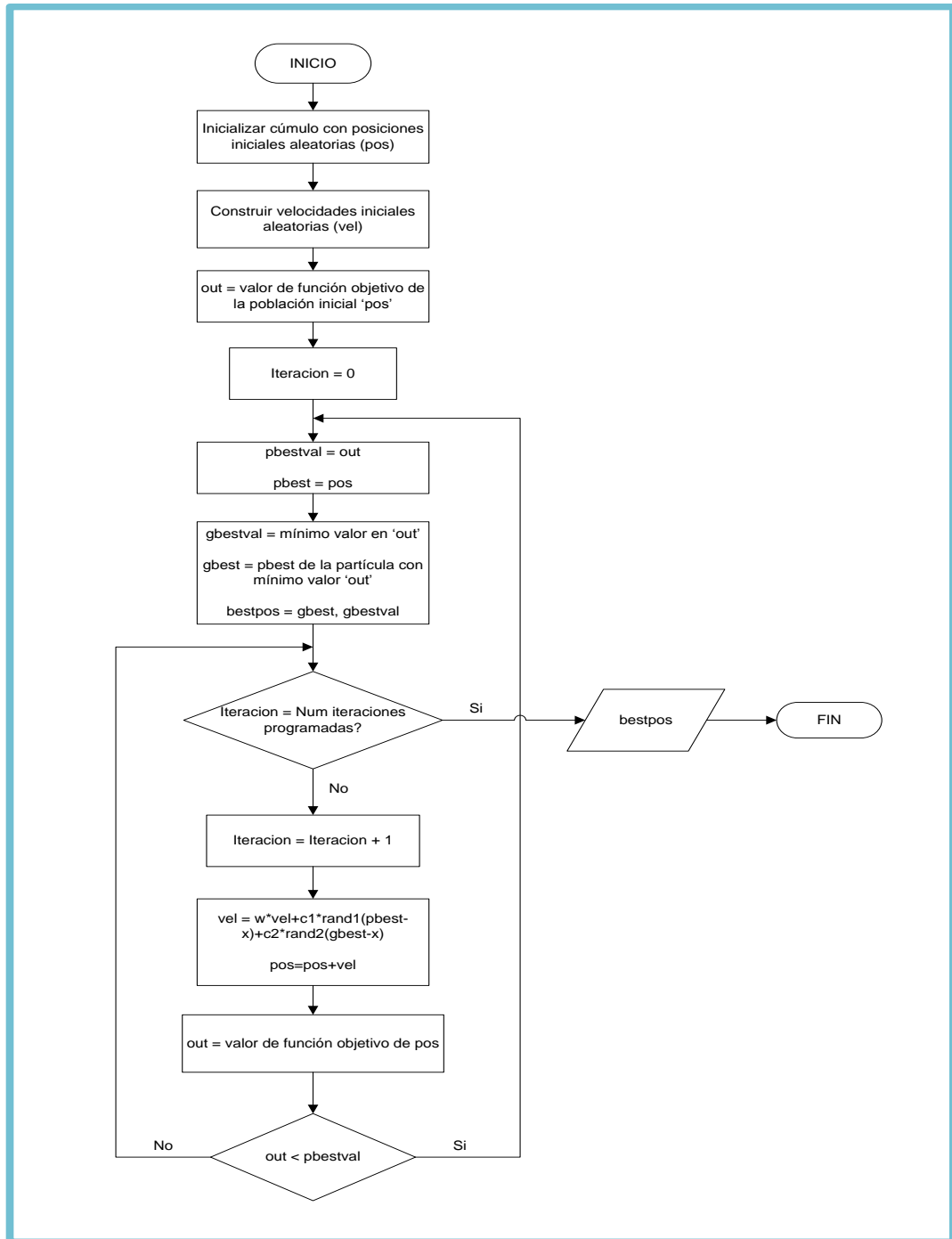
Los siguientes son los principales pasos que sigue el código de programación hasta obtener el valor de aptitud de todas las partículas enviadas por la función de la toolbox PSOt:

1. Toma la primera partícula o primera matriz del vector posiciones, para ser evaluada
2. Calcula las necesidades de máquina por tipo de máquina con base en los tiempos totales de operación, (que resultan de multiplicar los tiempos de operación de la matriz Top_{jp} por la demanda de cada producto), la configuración de la matriz M_{jp} y la configuración de celdas contenida en la partícula
3. Calcula el costo de máquinas multiplicando las necesidades de cada tipo de máquina por el costo unitario de mantener cada tipo de máquina en el sistema
4. Calcula el número de transferencias intercelulares, las cuales ocurren cuando operaciones del mismo producto son asignadas a celdas de fabricación diferentes
5. Calcula el costo de transferencias intercelulares multiplicando el número de transferencias calculado por el costo de realizar una transferencia intercelular
6. Calcula el costo total (valor de la función objetivo) sumando el costo de máquinas más el costo de transferencias intercelulares
7. Verifica el cumplimiento de restricción de tamaño de celda. Si el número de máquinas sobrepasa el límite establecido se penaliza el valor *fitness* de la

partícula en una cantidad proporcional al número de máquinas por fuera de los tamaños de celda definidos

8. Almacena los valores de la función objetivo calculados en el vector de escalares 'out' y lo devuelve a la función 'pso_trelea_vectorized.m' para que el algoritmo continúe su rutina. Retorna además las necesidades de máquina de cada partícula dentro del arreglo denominado 'AsignacionMaq_Celda' y la información de si es una solución factible (1) o si por el contrario excede los tamaños de celda (0) dentro de la variable denominada 'Restricción'

Figura 5: Diagrama de flujo algoritmo PSO



Fuente: Autor

5. AJUSTE DE PARÁMETROS DEL ALGORITMO PSO

De la literatura relacionada con el algoritmo PSO, tomando como referencia la modelación matemática propia del algoritmo descrita en el marco teórico y luego de cierto número de corridas de prueba del algoritmo, se estableció que los parámetros fundamentales que caracterizan al algoritmo son los coeficientes de aceleración c_1 y c_2 y el factor de inercia w . Adicionalmente se contó como parámetro a ajustar el tamaño de la población, para un total de cuatro factores de interés para el diseño de experimentos. Debido a la naturaleza de estos factores, que en teoría pueden tomar como valor cualquier número comprendido en el conjunto de los números reales no negativos³⁶, se hace difícil la aplicación de cualquier tipo de diseño factorial en tanto que no se cuenta con un número definido o deseado de niveles asociados a cada factor. Bajo tales consideraciones, el ajuste de los parámetros se realizará a través de un Diseño Completamente al Azar (DCA)³⁷

El ajuste se realizó a través de la resolución del primero de los problemas planteados para la etapa de experiencia computacional de la siguiente forma: inicialmente, el algoritmo se corrió para cuatro valores dados de cada factor, moviendo uno a la vez, con lo que se tenían 16 tratamientos a aplicar. Se realizaron 5 réplicas por tratamiento, para un total de 80 corridas experimentales del algoritmo. Con los resultados obtenidos en las 80 observaciones se realizó un análisis de varianza a fin de determinar si los parámetros seleccionados influían en el desempeño mostrado por el algoritmo.

Si existía suficiente evidencia para concluir que alguno de los parámetros influía realmente en la respuesta encontrada por el algoritmo, se realizaría un nuevo

³⁶ Con excepción del tamaño de la población cuyo dominio es cualquier subconjunto de los números enteros positivos

³⁷ Es el más simple de todos los diseños que se utilizan para comparar dos o más tratamientos dado que solo consideran dos fuentes de variabilidad: los tratamientos y el error aleatorio

diseño, esta vez solo con los parámetros que demostraron tener efecto en la solución hallada. Los nuevos valores a asignar de estos parámetros para las nuevas corridas dependerían del valor del parámetro con el cual se obtuvo el mejor valor de aptitud en la primera etapa del diseño. Nuevamente se llevaría a cabo un análisis de varianza para determinar finalmente el valor de cada parámetro estudiado en la segunda etapa del diseño que mejor rendimiento le generara al algoritmo. Este será el valor que se utilizará para la resolución de los seis problemas restantes propuestos en la investigación.

El problema a resolver se tomó del trabajo de investigación desarrollado en (Gómez Gómez, 2007). El problema consta de 30 productos a fabricar, con el empleo de 10 tipos de máquina a ser agrupadas en 4 celdas de manufactura. El tamaño máximo y mínimo de celda fue de 10 y 2 máquinas respectivamente. La capacidad de máquina, cualquiera que fuera su tipo se fijó en 480 horas por período³⁸. El costo de transferencias intercelulares se fijó en una (1) unidad monetaria (UM). En el ANEXO N se encuentra la información sobre la demanda del período de cada uno de los productos a fabricar, los costos de máquina por período, el tipo de máquina en la que debe realizarse cada operación de cada producto y los tiempos de operación requeridos para la fabricación de un lote de cada producto. Adicionalmente, del trabajo de grado citado se tiene que el costo mínimo, hallado a través de un software de programación lineal entera mixta, para esta configuración del problema de formación de celdas es de 22006 UM.

5.1. COEFICIENTE DE ACCELERACIÓN C1

El tamaño de la población se fijó en 300 partículas mientras que para los demás parámetros se tomaron los valores asignados por defecto en la toolbox. En la

³⁸ Resultado de una jornada laboral de 8 horas diarias, 5 días a la semana, 12 semanas (3 meses) por período

Tabla 2 se muestran los cuatro diferentes niveles en los que se trabajó el parámetro, a criterio del investigador. Recuérdese que el valor por defecto, y por demás recomendado en la revisión bibliográfica realizada, es igual a dos (2).

Tabla 2: Niveles del parámetro C1

PARÁMETRO A AJUSTAR	Nivel1	Nivel2	Nivel3	Nivel4
C1	2	4	6	8

Fuente: Autor

En la Tabla 3 se observan los resultados consolidados del diseño experimental aplicado para este primer factor. En ella pueden verse los valores promedio y desviación estándar tanto del costo de la mejor solución encontrada (COSTO) como del tiempo de ejecución del algoritmo (TIEMPO), de las cinco réplicas tomadas de cada nivel³⁹, así como el porcentaje de soluciones factibles encontradas por el cúmulo, a manera de información secundaria sobre su desempeño. Adicionalmente en los anexos D, E, F, G y H se encuentran los resultados detallados por réplica del diseño aplicado para este y para todos los parámetros a ajustar.

Según la información contenida en la tabla, parece haber una diferencia significativa en las medias muestrales de los tratamientos en cuanto a la variable COSTO se refiere. Por el contrario, en lo que respecta a la variable de respuesta TIEMPO las diferencias encontradas en las medias muestrales no parecen ser lo suficientemente significativas. En todo caso en las siguientes dos secciones se realiza el Análisis de Varianza (ANOVA) correspondiente a cada una de las variables de respuesta estudiadas en el diseño.

³⁹ Dado que el algoritmo admite soluciones no factibles que violan los tamaños de celda y a pesar de la penalización aplicada es posible que resulten teniendo mejores valores de función de aptitud que aquellas soluciones que sí son factibles, se ha tomado no el mejor valor encontrado sino el mejor valor factible encontrado por el algoritmo

Tabla 3: Resultados consolidados del diseño experimental aplicado al factor Coeficiente C1

COEF. C1	COSTO		TIEMPO		% FACTIBLES
	PROMEDIO	DESV. EST.	PROMEDIO	DESV. EST.	
A = 2	22435,6000	43,8668	447,4641	5,7581	74,39%
B = 4	22356,0000	80,4177	447,8165	7,1340	66,00%
C = 6	22555,2000	139,7773	450,1105	12,1432	57,18%
D = 8	22540,8000	94,8483	446,1851	4,2382	49,09%

Fuente: Autor

5.1.1. Análisis de varianza para el COSTO

En la Tabla 4 se presenta el ANOVA de la variable COSTO de las observaciones colectadas.

A partir de los resultados arrojados por el ANOVA se concluye con una confianza del 95% que existe suficiente evidencia estadística para rechazar la hipótesis (nula) de que los tratamientos del factor, esto es, los diferentes valores dados al parámetro, no tienen efecto sobre la variable de respuesta COSTO ($F_o > \text{Valor crítico de } F$) y se acepta que al menos un par de valores dados al coeficiente C1 tienen un costo promedio diferente. Ahora bien, ¿cuáles valores asignados a C1 son estadísticamente diferentes entre sí? Para responder a esta pregunta se construye el gráfico de medias, con el método LSD⁴⁰, el cual se presenta en la Figura 6.

⁴⁰ Método de comparaciones múltiples que permite identificar cuáles tratamientos resultaron diferentes luego de rechazar la hipótesis nula en un análisis de varianza, a través de la construcción de intervalos de confianza para las medias con base en una cantidad calculada conocida como 'diferencia mínima significativa'. De esta manera, si dos intervalos se traslapan significa que no hay diferencias entre las medias de los tratamientos correspondientes.

Tabla 4: Análisis de Varianza variable COSTO Factor C1

<i>Origen de las variaciones</i>	<i>SC*</i>	<i>GL**</i>	<i>CM***</i>	<i>Fo****</i>	<i>valor-p*****</i>	<i>Valor crítico F</i>
Entre grupos	132183	3	44061,000	4,773	0,015	3,239
Dentro de grupos	147700,8	16	9231,300			
Total	279883,8	19				

Fuente: Autor

*SC: Suma de Cuadrados

**GL: Grados de Libertad

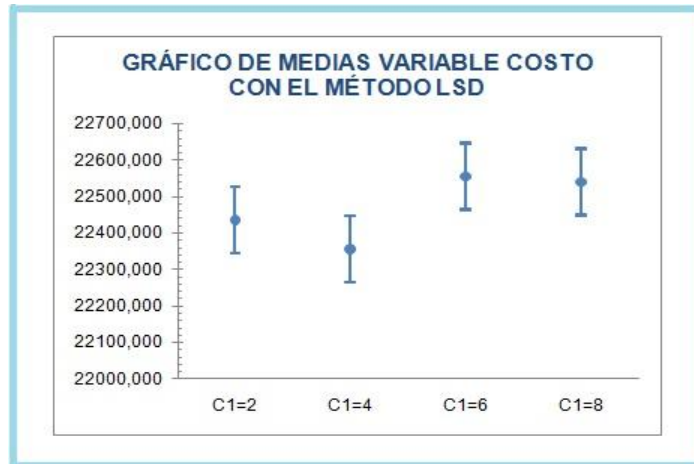
***CM: Cuadrados Medios

****Fo: estadístico de prueba

*****valor-p: significancia observada

Del gráfico se concluye que el método LSD detecta con una confianza del 95% que $(C1=4) \neq (C1=6)$ y $(C1=4) \neq (C1=8)$. De esta manera la conclusión práctica del experimento es que el mejor valor del parámetro en cuanto al costo es $C1 = 4$, ya que estadísticamente los costos que genera son menores que los de los tratamientos 3 y 4. Entre los tratamientos 1 y 2 no existen diferencias significativas; sin embargo la media del tratamiento 2 sigue siendo la de menor valor, una razón más para concluir que el mejor rendimiento del algoritmo se logra con $C1=4$.

Figura 6: Gráfico de medias para la variable COSTO del factor C1



Fuente: Autor

5.1.2. Análisis de varianza para el TIEMPO

Se realiza el mismo análisis y procedimiento llevado a cabo para el costo. En la Tabla 5 se presenta el ANOVA resultante.

Tabla 5: Análisis de Varianza variable TIEMPO Factor C1

Origen de las variaciones	SC	GL	CM	F (Fo)	valor-p	Valor crítico F
Entre grupos	40,120617	3	13,374	0,214	0,885	3,239
Dentro de grupos	997,87758	16	62,367			
Total	1037,9982	19				

Fuente: Autor

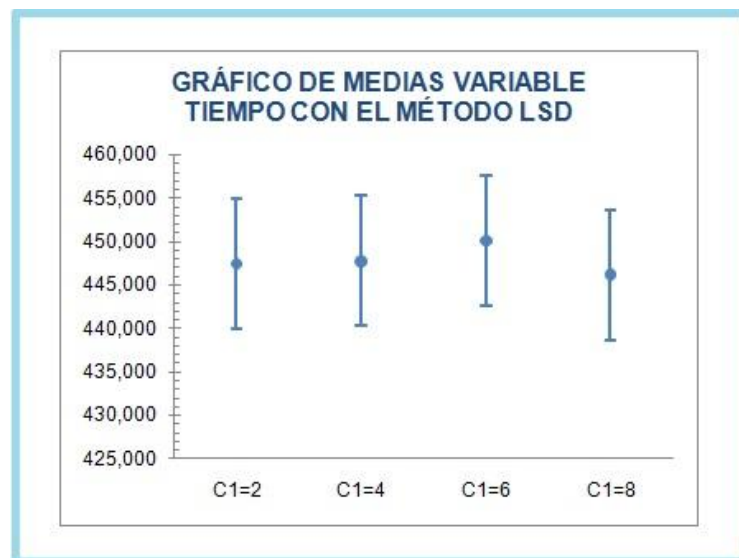
Como resultado se acepta la hipótesis nula (ya que $F_o < \text{Valor crítico de } F$) y se concluye con una confianza del 95% que los tiempos promedio de los cuatro tratamientos son estadísticamente iguales. Del análisis se observa además que la variabilidad debida al error (CM dentro de grupos) es considerablemente mayor

que la variabilidad debida a los tratamientos (CM entre grupos) lo cual constituye una prueba más a favor de la hipótesis nula que se aceptó.

En todo caso, en la Figura 7 se presenta el gráfico de medias para la variable TIEMPO. Se confirman una vez más, esta vez de manera gráfica, la validez de los resultados obtenidos a través del ANOVA.

5.1.3. Verificación de los supuestos del modelo

Figura 7: Gráfico de medias variable TIEMPO Factor C1



Fuente: Autor

Cuando se realiza el ANOVA y su resultado es el rechazo de la hipótesis nula o de igualdad de medias, se procede a estimar el modelo ajustado, que para el caso del DCA se puede escribir como:

$$\hat{Y}_{ij} = \bar{Y}_{..} + (\bar{Y}_{i.} - \bar{Y}_{..}) = \bar{Y}_{i.} \quad (10)$$

Lo que quiere decir la anterior ecuación es que la respuesta predicha para cada observación (\hat{Y}_{ij}) es la media muestral del tratamiento i correspondiente ($\bar{Y}_{i.}$)⁴¹

Para la comprobación de los supuestos se utilizó la muestra de residuos ya que al ser la diferencia entre la respuesta observada (Y_{ij}) (j -ésima observación en el tratamiento i) y la respuesta predicha por el modelo (\hat{Y}_{ij}) permiten hacer un diagnóstico más directo de la calidad del modelo, pues su magnitud señala qué tan bien describe a los datos del mismo. Tomando como base la expresión de la ecuación (10), el residual o residuo asociado a la observación Y_{ij} está dado por:

$$e_{ij} = Y_{ij} - \hat{Y}_{ij} = Y_{ij} - \bar{Y}_{i.}$$

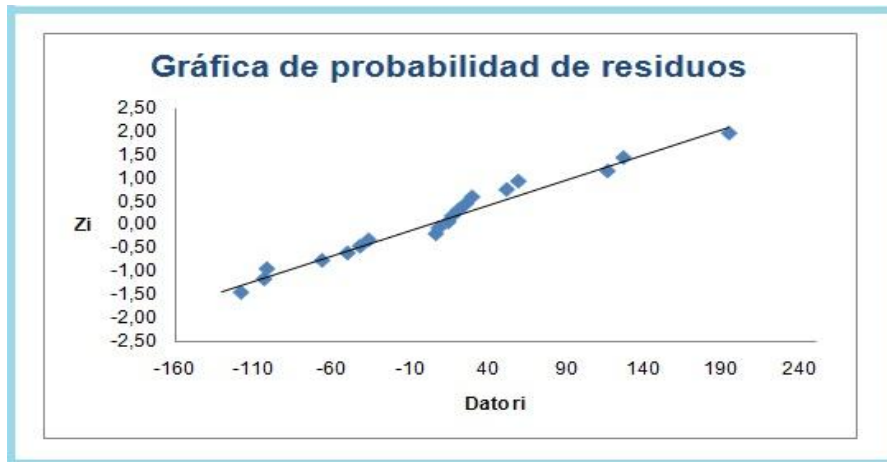
Para esta investigación, sea que los resultados del ANOVA resultaran o no significativos se realizó la comprobación de los supuestos del modelo, ya que es una manera de validar el diseño de experimentos aplicado. Para este primer diseño correspondiente al ajuste del primer factor se realizó la comprobación de cada supuesto paso por paso, a manera de ejemplo. La comprobación de los supuestos correspondientes a los ajustes de los demás factores pueden ser consultados en los anexos J, K, L, y M.

✓ Normalidad

Se utilizó el procedimiento de gráfica de probabilidad normal. Los residuos resultantes de los tratamientos aplicados y demás datos necesarios para la construcción de la gráfica, de todos los parámetros a ajustar, pueden ser igualmente consultados en el ANEXO I.

⁴¹ Para más información consultar en (Gutiérrez Pulido & De la Vara Salazar, 2008)

Figura 8: Gráfica de probabilidad normal de residuos factor C1



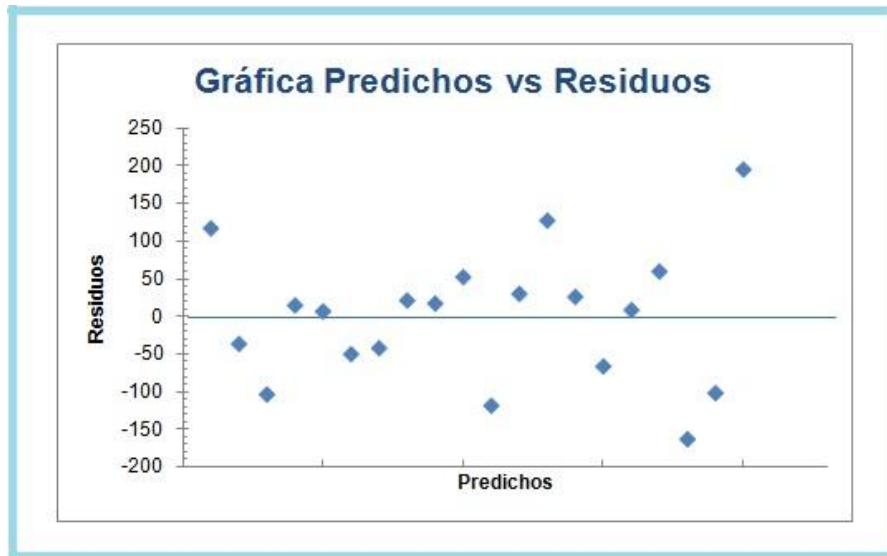
Fuente: Autor

En la Figura 8 se encuentra la gráfica de probabilidad normal de los residuos de las 20 observaciones realizadas. Como la dispersión de los puntos en la gráfica tiende a quedar alineada en una línea recta se puede concluir que los residuos e_{ij} de las observaciones del diseño siguen una distribución normal con media cero.

✓ Varianza constante

En la Figura 9 se muestra la gráfica de valores predichos (ver apartado 5.1.3) contra residuos, la cual es un indicador de la semejanza de varianzas en los tratamientos aplicados. Como los puntos en esta gráfica se distribuyen de manera aleatoria en una banda horizontal (sin ningún patrón claro y contundente) se concluye que se cumple el supuesto de que los tratamientos aplicados tienen igual varianza σ^2 .

Figura 9: Gráfica de dispersión de valores predichos contra residuos del factor C1

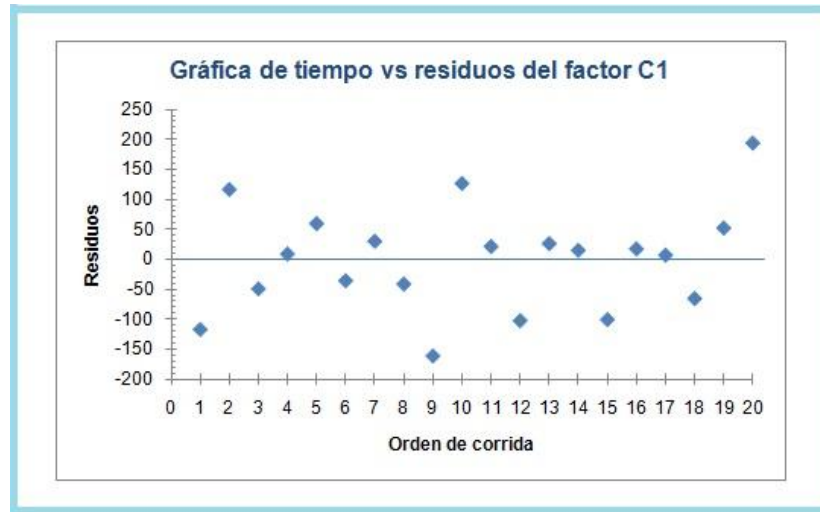


Fuente: Autor

✓ Independencia

Para verificar este supuesto se grafica el orden en que se obtuvo un dato contra el residuo correspondiente, comportamiento que se ve ilustrado en la Figura 10. Como no se detecta una tendencia no aleatoria claramente definida se concluye que el supuesto de que los e_{ij} son independientes entre sí se cumple a cabalidad en el diseño aplicado.

Figura 10: Gráfica de tiempo vs residuos para el factor C1



Fuente: Autor

5.1.4. Conclusiones sobre el diseño experimental realizado

- Del análisis de varianza realizado se concluye que el diseño cumple con los tres supuestos del modelo, lo cual valida los resultados obtenidos por el diseño aplicado.
- En cuanto a la variable de respuesta COSTO, los tratamientos 1, 3 y 4 son estadísticamente iguales entre sí, al igual que los tratamientos 1 y 2. La hipótesis nula se rechazó debido a que las medias muestrales de los tratamientos 2-3 y 2-4 son significativamente diferentes.
- En cuanto a la variable de respuesta TIEMPO no existen diferencias estadísticamente significativas en los niveles del factor trabajados. Es decir, el coeficiente de aceleración C1 no tiene efecto sobre el tiempo de ejecución consumido por el algoritmo, por lo que esta variable no influirá de manera alguna en la selección del valor 'óptimo' del parámetro.
- Aunque los tratamientos 1 y 2 son estadísticamente iguales, fue con el segundo de ellos que se obtuvo el menor valor promedio de costo (22356 UM).

Además, si bien este tratamiento no reportó la menor de las desviaciones del experimento fue la segunda de las cuatro desviaciones calculadas. Por ello se concluye del diseño que el valor del coeficiente C1 que le reporta un mejor rendimiento al algoritmo es igual a cuatro (4), y es con este valor con el que se trabajará este parámetro en el resto de la investigación.

- Las medias muestrales de los tratamientos 3 y 4 resultaron muy superiores comparadas con las de los tratamientos 1 y 2 en especial. Se concluye que no es bueno trabajar con valores superiores a 4 en este parámetro, para el problema específico de formación de celdas, dado que afecta negativamente el rendimiento que pueda lograr el algoritmo.
- El porcentaje de soluciones factibles encontradas por el cúmulo varía de manera inversamente proporcional a los valores asumidos por el coeficiente C1. Aún así, esta estadística no es más importante que las variables de respuesta establecidas, y siempre y cuando el algoritmo encuentre soluciones factibles, sin importar la cantidad y de hecho dichas soluciones factibles tengan mejor valor de aptitud que las no factibles, este porcentaje no influirá en la decisión sobre cuál nivel del factor es el más adecuado para el desempeño del algoritmo.
- Como la variación entre las medias de los niveles 1 y 2 del parámetro no fue lo suficientemente significativa, mientras que entre los niveles 2 y 3 el algoritmo desmejoró considerablemente su desempeño, no se considera necesario realizar una segunda etapa del diseño para ajustar con mayor precisión el parámetro en cuestión.

5.2. COEFICIENTE DE ACELERACIÓN C2

El tamaño de la población se fijó en 300 partículas, el valor de C1 en 4 y los demás parámetros se fijaron en los valores asignados por defecto en la toolbox.

Tabla 6: Niveles del factor C2

PARÁMETRO A AJUSTAR	Nivel1	Nivel2	Nivel3	Nivel4
C2	2	4	6	8

Fuente: Autor

En la Tabla 6 se muestran los cuatro diferentes niveles en los que se trabajó el parámetro, los cuales corresponden a los mismos niveles utilizados para el diseño del factor C1.

La Tabla 7 refleja los resultados consolidados del diseño experimental aplicado para este factor.

Tabla 7: Resultados consolidados del diseño experimental aplicado al factor Coeficiente C2

COEF. C2	COSTO		TIEMPO		% FACTIBLES
	PROMEDIO	DESV. EST.	PROMEDIO	DESV. EST.	
A = 2	22433,8000	77,5609	439,2259	9,2440	66,86%
B = 4	22495,4000	34,0044	442,8490	5,8074	49,89%
C = 6	22528,4000	83,8409	441,1553	10,0794	38,04%
D = 8	22557,0000	55,9687	443,7632	7,2370	33,08%

Fuente: Autor

Según la información de la tabla, a simple vista las medias de los tratamientos de la variable COSTO difieren notablemente entre sí. En cuanto a la variable TIEMPO lo más probable es que el ANOVA no resulte significativo a juzgar por los valores promedio de los cuatro tratamientos aplicados. El comportamiento en cuanto a la estadística porcentaje de factibles se mantiene respecto al diseño aplicado al factor C1, es decir, varía de manera inversa al valor asignado al parámetro.

5.2.1. Análisis de varianza para el COSTO

En la Tabla 8 se presenta la información correspondiente al ANOVA para la variable de respuesta COSTO. En ella se ve que el estadístico de prueba F_o es prácticamente idéntico al valor crítico de F y que el valor- p es exactamente igual al nivel de significancia α de la prueba. Con una confianza del 95% se acepta la hipótesis nula y se concluye que las medias de los tratamientos aplicados son estadísticamente iguales y que por ende ninguno de ellos tiene efecto sobre la variable observada.

Tabla 8: Análisis de Varianza variable COSTO Factor C2

<i>Origen de las variaciones</i>	<i>SC</i>	<i>GL</i>	<i>CM</i>	<i>F_o</i>	<i>valor-p</i>	<i>Valor crítico F</i>
Entre grupos	42029,35	3	14009,783	3,233	0,050	3,239
Dentro de grupos	69335,2	16	4333,450			
Total	111364,6	19				

Fuente: Autor

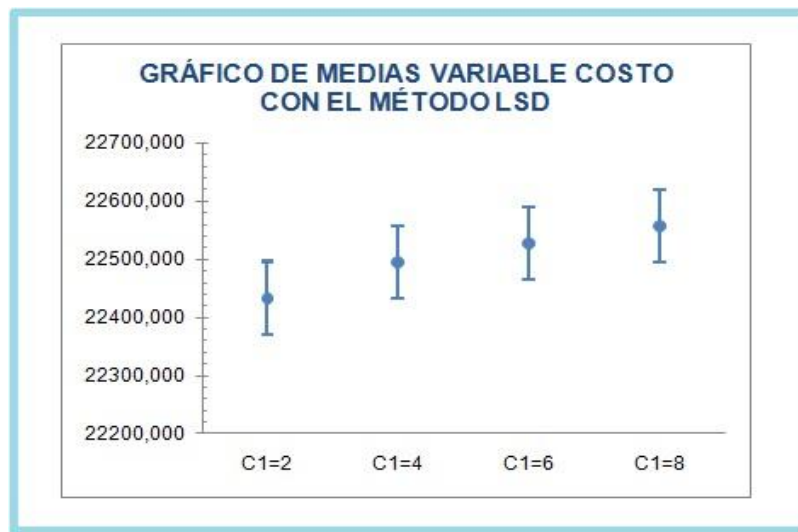
Para corroborar los resultados obtenidos con el ANOVA, en la Figura 11 se presenta el gráfico de medias de los tratamientos aplicados. En él se observa que los intervalos de los tratamientos se traslapan entre sí, lo cual significa que son estadísticamente iguales en cuanto a sus medias. Sin embargo, los tratamientos 1 y 4 parecen no cumplir con esa condición. Sucede que “el método LSD tiene una potencia importante, por lo que en ocasiones declara significativas aún pequeñas diferencias” (Gutiérrez Pulido & De la Vara Salazar, 2008). De todas formas recuérdese que la hipótesis nula se aceptó por una diferencia despreciable entre los estadísticos de prueba F_o y F .

5.2.2. Análisis de varianza para el TIEMPO

En la Tabla 9 se presenta el ANOVA resultante.

En este caso también se acepta la hipótesis nula y se concluye con una confianza del 95% que no hay evidencia suficiente para considerar que las medias de los tiempos de los cuatro tratamientos son diferentes entre sí. Del análisis se observa además que la variabilidad debida al error es considerablemente mayor que la variabilidad debida a los tratamientos, lo cual constituye una prueba más a favor de la hipótesis nula que se aceptó.

Figura 11: Gráfico de medias variable COSTO Factor C2



Fuente: Autor

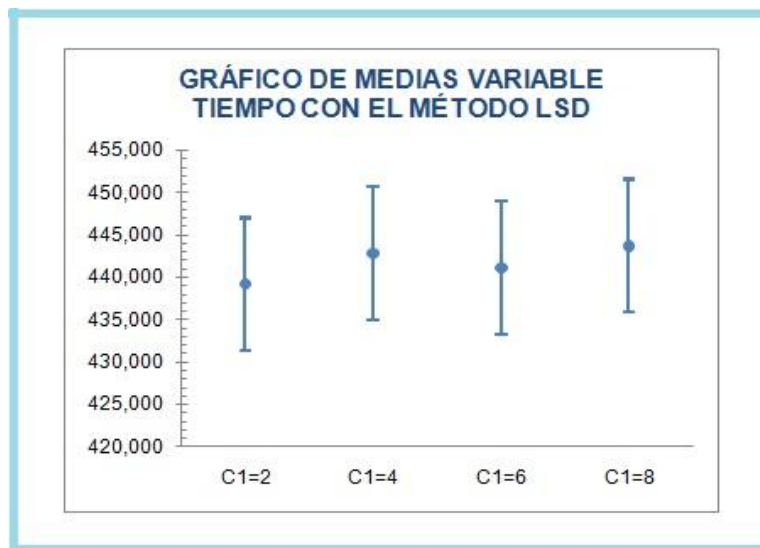
Tabla 9: Análisis de Varianza variable TIEMPO Factor C2

Origen de las variaciones	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	59,928	3	19,976	0,293	0,830	3,239
Dentro de grupos	1092,579	16	68,286			
Total	1152,507	19				

Fuente: Autor

En la Figura 12 se muestra el gráfico de medias para la variable de respuesta TIEMPO. De él se concluye, nuevamente con una confianza del 95%, que efectivamente los tratamientos aplicados son estadísticamente iguales en cuanto a sus medias.

Figura 12: Gráfico de medias variable TIEMPO Factor C2



Fuente: Autor

5.2.3. Conclusiones sobre el diseño experimental realizado

- El diseño aplicado cumple con los tres supuestos del modelo: normalidad, igualdad de varianzas e independencia, lo cual valida los resultados obtenidos por el mismo.
- Tanto para la variable COSTO como para la variable TIEMPO se concluye que los tratamientos aplicados tienen medias iguales entre sí e iguales a cero, por lo que no tienen efecto sobre las respuestas observadas.
- Aunque todos los tratamientos son estadísticamente iguales, es con el primer nivel del factor con el que se obtiene el menor valor para las dos variables de respuesta observadas (22433,8 UM y 439,2259 s respectivamente). Por lo

tanto se concluye del diseño que el valor del coeficiente C2 que le proporciona un mejor rendimiento al algoritmo es igual a dos (2), y es con este valor con el que se trabajará este parámetro en el resto de la investigación.

- El porcentaje de soluciones factibles encontradas por el cúmulo varía de manera inversamente proporcional a los valores asumidos por el coeficiente C2. Nuevamente se aclara que esta estadística no es más importante que las variables de respuesta establecidas.
- Como los tratamientos resultaron estadísticamente iguales no es necesario realizar una segunda etapa del diseño para ajustar con mayor precisión el parámetro en cuestión.

5.3. FACTOR DE INERCIA W

El tamaño de la población se fijó en 300 partículas, el valor de C1 en 4, el valor de C2 en 2 y los demás parámetros se fijaron en los valores asignados por defecto en la toolbox. En esta, en realidad existen dos valores de inercia, uno inicial y uno final, y durante la ejecución del algoritmo este valor va cambiando por sí solo dentro del rango $[W_{inicial} \ W_{final}]$, siempre de manera descendente. Para el ajuste de este parámetro se movió solamente el valor inicial de inercia pues según la bibliografía consultada el valor final se introduce solamente como una forma de evitar que este parámetro llegue hasta valores muy cercanos a cero afectando negativamente el movimiento de las partículas, pues al estar multiplicado por la velocidad anularía esos movimientos. En la Tabla 10 se muestran los cuatro diferentes niveles en los que se trabajó el parámetro.

Tabla 10: Niveles del factor W

PARÁMETRO A AJUSTAR	Nivel1	Nivel2	Nivel3	Nivel4
W	1.0	0.9	0.8	0.7

Fuente: Autor

Por su parte, en la Tabla 11 se muestran los resultados consolidados del diseño experimental aplicado para este factor.

Tabla 11: Resultados consolidados del diseño experimental aplicado al factor peso inercial W

FACTOR W	COSTO		TIEMPO		% FACTIBLES
	PROMEDIO	DESV. EST.	PROMEDIO	DESV. EST.	
A = 1.0	22484,4000	148,3924	433,1537	3,5087	58,00%
B = 0.9	22418,6000	65,3322	433,6244	4,9194	62,95%
C = 0.8	22486,6000	109,3837	430,4741	2,6477	74,82%
D = 0.7	22487,6000	121,7284	434,0046	0,9392	83,20%

Fuente: Autor

Tal parece que el único tratamiento diferente a los demás, en cuanto al costo, es el tratamiento con $W = 0.9$. Para la variable TIEMPO el comportamiento se mantiene igual al de los dos diseños de los factores ya ajustados, es decir, que los diferentes tratamientos aplicados son estadísticamente iguales en cuanto a esta variable se refiere. En cuanto al porcentaje de factibles, entre menor sea el valor del peso inercial W mayor será el número de soluciones factibles encontradas por el cúmulo y viceversa.

5.3.1. Análisis de varianza para el COSTO

La Tabla 12 contiene la información del ANOVA correspondiente. Se concluye que existe suficiente evidencia estadística para aceptar la hipótesis nula según la cual las medias de los tratamientos aplicados son iguales entre sí e iguales a cero.

Por su parte la Figura 13 muestra el gráfico de medias correspondiente a la variable COSTO. En ella se observa que los intervalos de los tratamientos se traslapan entre sí, lo cual significa, con una confianza del 95%, que son estadísticamente iguales en cuanto a sus medias.

Tabla 12: Análisis de Varianza variable COSTO Factor C2

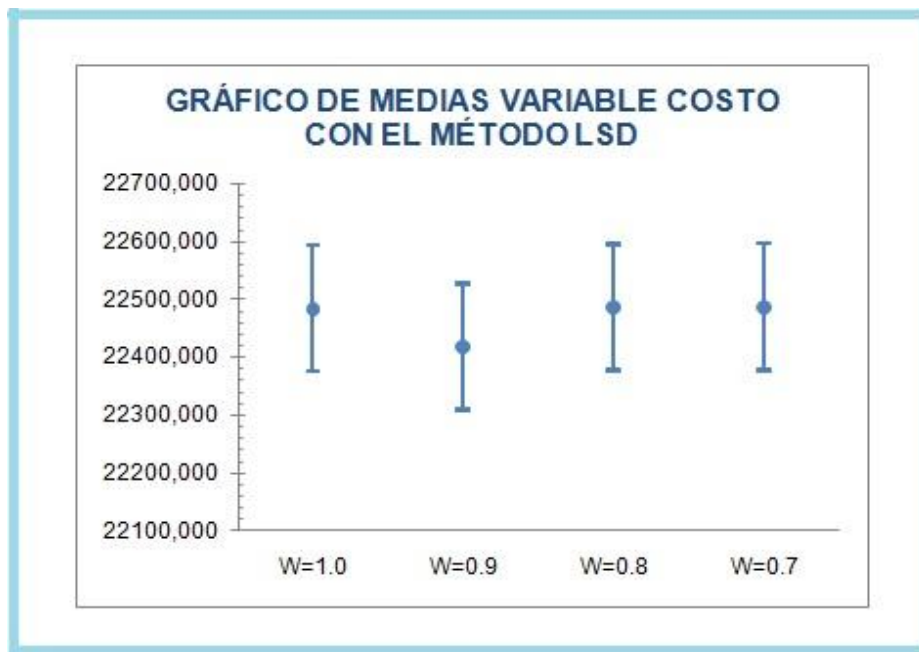
Origen de las variaciones	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	17163,4	3	5721,133	0,431	0,734	3,239
Dentro de grupos	212284,8	16	13267,800			
Total	229448,2	19				

Fuente: Autor

5.3.2. Análisis de varianza para el TIEMPO

En la Tabla 13 se presenta el ANOVA resultante.

Figura 13: Gráfico de medias variable COSTO Factor peso W



Fuente: Autor

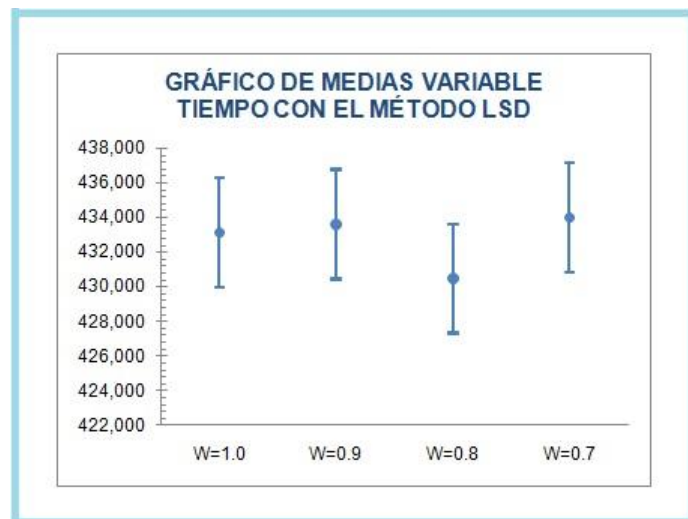
Tabla 13: Análisis de Varianza variable TIEMPO Factor C2

Origen de las variaciones	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	38,325	3	12,775	1,151	0,359	3,239
Dentro de grupos	177,615	16	11,101			
Total	215,940	19				

Fuente: Autor

Como en el caso de los dos parámetros ya ajustados, se acepta la hipótesis nula y se concluye con una confianza del 95% que no existen diferencias estadísticas significativas entre los tratamientos aplicados en cuanto al tiempo de ejecución del algoritmo se refiere.

Figura 14: Gráfico de medias variable TIEMPO Factor peso W



Fuente: Autor

En la Figura 14 se muestra el gráfico de medias para la variable de respuesta TIEMPO. Se concluye con una confianza del 95%, que los tratamientos aplicados son estadísticamente iguales en cuanto a sus medias.

5.3.3. Conclusiones sobre el diseño experimental realizado

- Del análisis de varianza realizado se concluye que el diseño cumple con los supuestos de normalidad, igualdad de varianzas e independencia.
- Tanto para la variable de respuesta COSTO como TIEMPO, los efectos de los tratamientos resultaron estadísticamente iguales entre sí e iguales a cero.
- De los cuatro tratamientos aplicados, aún cuando las medias resultaron ser estadísticamente iguales, es con el segundo tratamiento con el que se logró la menor de todas las medias de la variable COSTO (22418,6 UM). Además este tratamiento fue el que presentó la menor desviación estándar del diseño. No obtuvo la menor de las medias en cuanto al tiempo pero su diferencia a simple vista respecto a las medias de los otros tratamientos es verdaderamente mínima. Por todo ello se concluye del diseño que el valor del peso inercial W que le proporciona un mejor rendimiento al algoritmo es igual a 0,9, y es con este valor con el que se trabajará este parámetro durante la investigación.
- Mientras el algoritmo actualizaba la pantalla con las mejores soluciones encontradas cada cierto número de iteraciones, se observó que con los valores 0.8 y especialmente 0.7 del factor inercial el algoritmo tiende a quedar atrapado en óptimos locales: aunque durante las primeras iteraciones se encontraban mejores soluciones, a partir de la mitad de las mismas o un poco después raras veces el algoritmo volvió a cambiar su valor g_{best} , hasta el término de la corrida. Se concluye que no debe trabajarse con valores de este parámetro inferiores a los asignados por defecto, y que de hecho el valor asignado por defecto en la toolbox efectivamente le proporciona un muy buen desempeño al algoritmo.
- A mayor magnitud del valor asignado al parámetro peso inercial W el algoritmo disminuye el número de soluciones factibles encontradas. Por el contrario, cuando el valor asignado al parámetro disminuye, el porcentaje de partículas factibles aumenta.

- Dado que al hacer mínimos cambios en el valor del parámetro (una décima) los resultados fueron tan contundentes, esto es, aumentando una décima se desmejoró el valor encontrado de la variable COSTO y disminuyendo una y dos décimas no solo se desmejoró la respuesta sino que además el cúmulo empezó a estancarse en óptimos locales, no se considera necesaria la aplicación de un segundo diseño para un mejor ajuste del parámetro.

5.4. TAMAÑO DE LA POBLACIÓN

Para este diseño $C1=4$, $C2=2$ y $W=0.9$. Los demás parámetros se fijaron en los valores asignados por defecto en la toolbox. En la Tabla 14 se muestran los cuatro diferentes niveles en los que se trabajó el parámetro, a criterio del investigador y tomando en cuenta que para el algoritmo genético se empleó un tamaño de población igual a 500 cromosomas por generación.

Tabla 14: Niveles factor tamaño de población

PARÁMETRO A AJUSTAR	Nivel1	Nivel2	Nivel3	Nivel4
Tamaño de la población	300	500	800	1200

Fuente: Autor

La Tabla 15 muestra los resultados consolidados del diseño experimental aplicado para este factor. Estos resultados sugieren una clara diferencia entre las medias de los primeros tres tratamientos y el último en cuanto a la variable COSTO se refiere. Para el caso de la variable TIEMPO parece ser que todos los tratamientos son diferentes entre sí y tienen efecto sobre la respuesta. El porcentaje de soluciones factibles encontradas es independiente del tamaño de la población establecido y dado que supera el 60% en todos los tratamientos es un número aceptable de precisión lograda por el algoritmo.

Tabla 15: Resultados consolidados del diseño experimental aplicado al factor tamaño de población

TAMAÑO POBLACIÓN	COSTO		TIEMPO		% FACTIBLES
	PROMEDIO	DESV. EST.	PROMEDIO	DESV. EST.	
A = 300	22356,4000	38,7337	442,4113	6,0327	66,53%
B = 500	22377,8000	51,6159	738,1641	13,9711	64,18%
C = 800	22392,4000	40,2778	1179,4600	14,3146	62,24%
D = 1200	22222,8000	21,3120	1779,5000	17,1326	65,08%

Fuente: Autor

5.4.1. Análisis de varianza para el COSTO

En la Tabla 16 se presenta el ANOVA para la variable COSTO.

Tabla 16: Análisis de Varianza variable COSTO Factor tamaño de población

<i>Origen de las variaciones</i>	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	90756,55	3	30252,183	19,389	0,000	3,239
Dentro de grupos	24964	16	1560,250			
Total	115720,6	19				

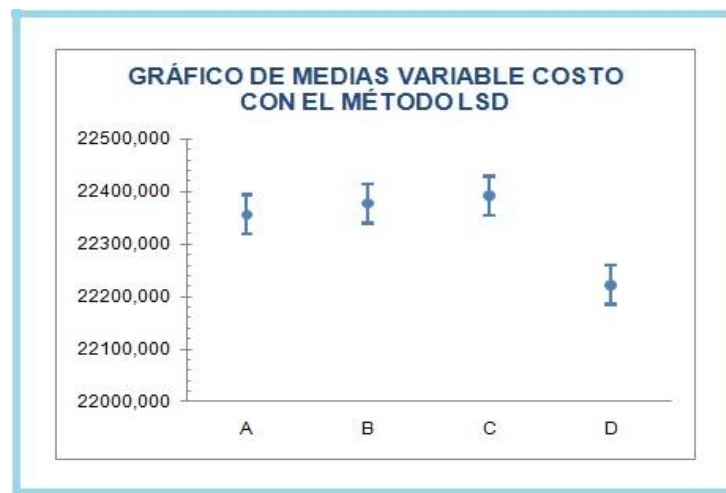
Fuente: Autor

Los resultados del ANOVA son contundentes. Con una confianza del 95% se rechaza la hipótesis nula y se concluye que existe suficiente evidencia estadística para considerar que al menos dos de los tratamientos aplicados son estadísticamente diferentes entre sí y tienen un efecto sobre la variable de respuesta observada

Aunque con base en los resultados consolidados de la Tabla 15 se intuye que el tratamiento que resulta diferente a los demás es el cuarto, en la Figura 15 se presenta el gráfico de medias de los tratamientos aplicados.

Efectivamente, el tratamiento que es diferente a los demás es el cuarto, y tiene un efecto muy importante en el costo observado en las corridas. Si solo se tratara de mejorar el rendimiento en términos del costo hallado, el tamaño de población a elegir sería definitivamente igual a 1200 partículas. Sin embargo, antes de tomar una decisión es necesario realizar el ANOVA correspondiente al tiempo de ejecución consumido por el algoritmo.

Figura 15: Gráfico de medias para la variable COSTO factor tamaño de población



Fuente: Autor

5.4.2. Análisis de varianza para el TIEMPO

En la Tabla 17 se presenta el ANOVA resultante.

Tabla 17: Análisis de Varianza variable TIEMPO factor Tamaño de población

Origen de las variaciones	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	5072108,9	3	1690703	9263,88	0,000	3,239
Dentro de grupos	2920,0767	16	182,505			
Total	5075029	19				

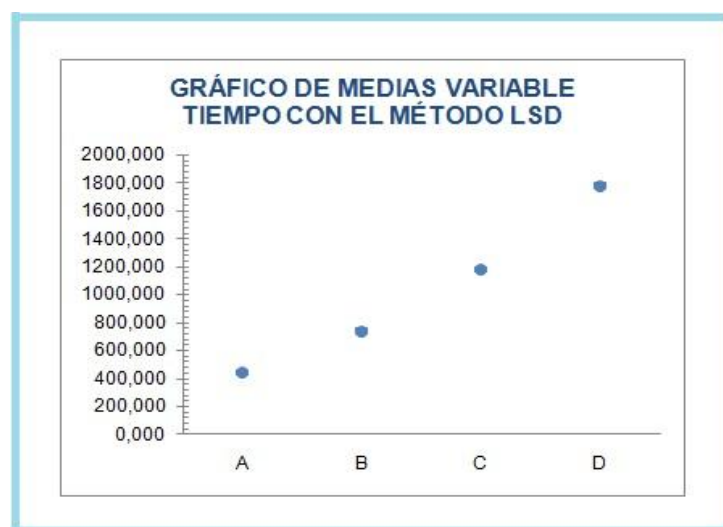
Fuente: Autor

Al igual que con la variable COSTO, de hecho de manera aún más contundente, el resultado arroja que con una confianza del 95% se rechaza la hipótesis nula y se concluye que las medias de al menos dos de los tratamientos aplicados son diferentes entre sí y por ende tienen efecto sobre el tiempo de ejecución consumido.

La Figura 16 muestra el gráfico de medias correspondiente para la variable TIEMPO.

Del gráfico que se concluye que todos los tratamientos son diferentes entre sí. También se concluye que el mejor tratamiento a aplicar en cuanto al tiempo de ejecución consumido es el tratamiento 1, que corresponde a un tamaño de población igual a 300 partículas, y que el tratamiento que genera el peor rendimiento del algoritmo es el cuarto, que corresponde a un tamaño de población igual a 1200 partículas, pero este es el mismo tamaño de población que reporta los mejores valores encontrados para la variable COSTO. Las diferencias entre las medias de los tratamientos son muy significativas.

Figura 16: Gráfico de medias variable TIEMPO factor Tamaño de población



Fuente: Autor

5.4.3. Conclusiones sobre el diseño experimental realizado

- El diseño de experimentos aplicado cumple con los supuestos de normalidad, igualdad de varianzas e independencia.
- Para la variable COSTO los tratamientos 1, 2 y 3 son estadísticamente iguales mientras que el tratamiento 4 es el que hace la diferencia y tiene un efecto muy importante sobre la respuesta observada. La media de este último tratamiento es la de menor costo (22222,8 UM) y le proporciona al algoritmo un rendimiento notablemente superior en comparación con los otros tres tratamientos aplicados.
- Para la variable TIEMPO las diferencias en los tratamientos son mucho más marcadas, y todos ellos son diferentes entre sí. El tiempo de ejecución del algoritmo aumenta de manera directamente proporcional al tamaño de la población establecido. En este caso, es el tratamiento 1 el que le proporciona el mejor rendimiento al algoritmo (442,4113 s), mientras que el tratamiento 4, que fue el que obtuvo mejores resultados en cuanto al costo, afecta de la peor manera el rendimiento del algoritmo al emplear tiempos de ejecución hasta cuatro veces mayores que los del mejor tratamiento (1779,5 s).
- El porcentaje de soluciones factibles encontradas por el cúmulo no está condicionado al tamaño de la población que se escoja. En todo caso, estos porcentajes, al ser superiores al 60% en todos los tratamientos, constituyen un resultado aceptable para el rendimiento del algoritmo en cuanto a esta estadística se refiere.
- Considerando que: i) los resultados del costo obtenidos con el mejor tratamiento, en este caso 1200 partículas, son amplia y significativamente mejores a los de los demás tamaños de población seleccionados, pero a su vez el tiempo de ejecución consumido por este tamaño es el más alto de los tiempos consumidos por todos los tratamientos. ii) aunque el tratamiento correspondiente a 300 partículas no obtuvo los mejores resultados en cuanto al costo, tampoco resultaron ser los peores de todo el diseño, mientras que el

tiempo de ejecución consumido por este tamaño sí fue el más bajo, mucho mejor que el consumido por las 1200 partículas. Bajo tales consideraciones, se concluye que es necesario realizar un quinto diseño de experimentos que permita determinar cuál de los tamaños de población mencionados es el más adecuado para el rendimiento del algoritmo tanto en términos de costo como de tiempo de ejecución. Este último diseño se realizó con base en el número de iteraciones programadas.

5.5. TAMAÑO DE POBLACIÓN CON BASE EN EL NÚMERO DE ITERACIONES

De toda la investigación realizada hasta el momento se ha observado que es muy probable que al aumentar el número de iteraciones el cúmulo de partículas encuentre mejores soluciones que las encontradas hasta la iteración número 2000. También se observó en el ajuste del tamaño de población, que la mejor solución encontrada con 1200 partículas en realidad se encontró en una iteración muy anterior a la de finalización de la corrida. En ese orden de ideas, se realizó un ajuste especialmente diseñado para tratar de solucionar este problema encontrado a partir del ajuste del parámetro tamaño de población realizado en el apartado inmediatamente anterior.

Para este diseño se corrió el algoritmo con un tamaño de 300 partículas para dos valores de iteraciones: 2000 que es el valor por defecto y un valor más, superior a 2000 pues la idea es averiguar si con mayor número de iteraciones el algoritmo llega a mejores soluciones afectando lo menos posible el tiempo de ejecución pues como es lógico, a mayor número de iteraciones mayor tiempo de ejecución consumido. A su vez, se corrió el algoritmo con un tamaño de 1200 partículas igualmente para dos valores de iteraciones: 2000 que es el valor por defecto y un valor más, inferior a 2000 pues la idea es averiguar si al reducir el número de iteraciones y con ellas el tiempo de ejecución, el algoritmo sigue encontrando

soluciones tan buenas como las encontradas en el diseño de experimentos ya aplicado. Se tomaron 5 réplicas del diseño.

En la tabla 18 se presenta la codificación de los tratamientos que fueron aplicados en esta etapa del ajuste del parámetro tamaño de población con número de iteraciones.

En la Tabla 19 se muestran los resultados consolidados del diseño experimental aplicado para esta etapa.

Tabla 18: Tratamientos para el segundo ajuste de tamaño de población

TRATAMIENTOS A APLICAR
A: 300 partículas, 2000 iteraciones
B: 300 partículas, 2500 iteraciones
C: 1200 partículas, 2000 iteraciones
D: 1200 partículas, 1400 iteraciones

Fuente: Autor

Aunque el objetivo de reducir el tiempo de ejecución para un tamaño de 1200 partículas y de no afectar tanto esta variable de respuesta para 300 partículas con más de 2000 iteraciones se cumplió, los resultados para el costo no parecen arrojar diferencias significativas respecto a los resultados obtenidos con la primera etapa del ajuste.

Tabla 19: Resultados consolidados segunda etapa del diseño aplicado al factor tamaño de población (con iteraciones)

TRATAMIENTO	COSTO		TIEMPO	
	PROMEDIO	DESV. EST.	PROMEDIO	DESV. EST.
A	22356,4000	38,7337	442,4113	6,0327
B	22353,4000	29,9383	551,0902	9,2439
C	22222,8000	21,3120	1779,5000	17,1326
D	22332,2000	20,4377	1217,5800	6,9139

Fuente: Autor

5.5.1. Análisis de varianza para el COSTO

En la Tabla 20 se presenta el ANOVA correspondiente al diseño de experimentos aplicado.

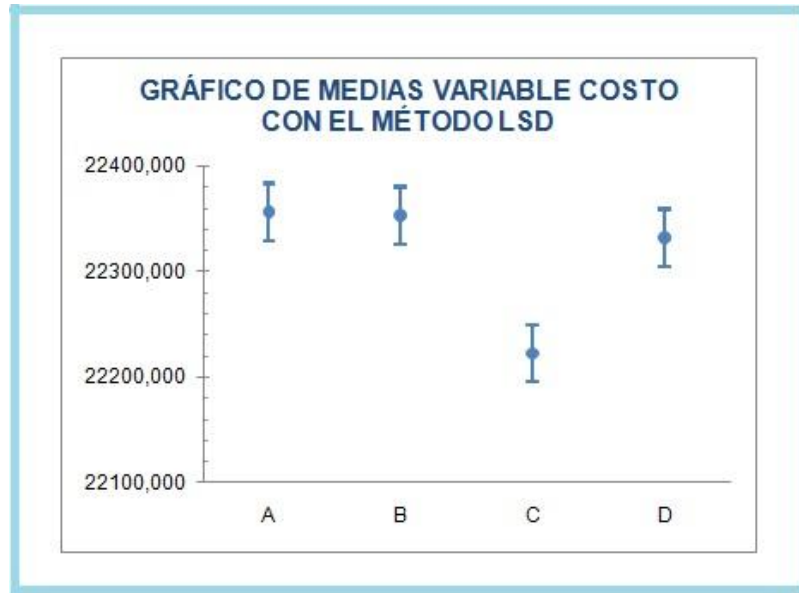
Tabla 20: Análisis de Varianza variable COSTO para Tamaño de población con iteraciones

<i>Origen de las variaciones</i>	<i>SC</i>	<i>GL</i>	<i>CM</i>	<i>Fo</i>	<i>valor-p</i>	<i>Valor crítico F</i>
Entre grupos	59897,2	3	19965,73333	24,434	0,000	3,239
Dentro de grupos	13074	16	817,125			
Total	72971,2	19				

Fuente: Autor

Con una confianza del 95% se rechaza la hipótesis nula y se acepta que al menos dos de los tratamientos son diferentes entre sí y tienen efecto sobre la variable de respuesta observada.

Figura 17: Gráfico de medias variable COSTO para tamaño de población con iteraciones



Fuente: Autor

La Figura 17 muestra el gráfico de medias correspondiente a la variable COSTO. En ella se ve que el mejor tratamiento en cuanto a costo se refiere continúa siendo el de 1200 partículas con 2000 iteraciones, es por demás el único tratamiento diferente de todo el diseño aplicado.

5.5.2. Análisis de varianza para el TIEMPO

En la Tabla 21 se presenta el ANOVA resultante. La evidencia estadística es concluyente: con una confianza del 95% se acepta la hipótesis alternativa según la cual las medias de al menos dos de los tratamientos aplicados son diferentes entre sí.

En la Figura 18 se muestra el gráfico de medias para la variable de respuesta TIEMPO. Se concluye con una confianza del 95% que las medias de todos los tratamientos aplicados son estadísticamente diferentes entre sí, y que el

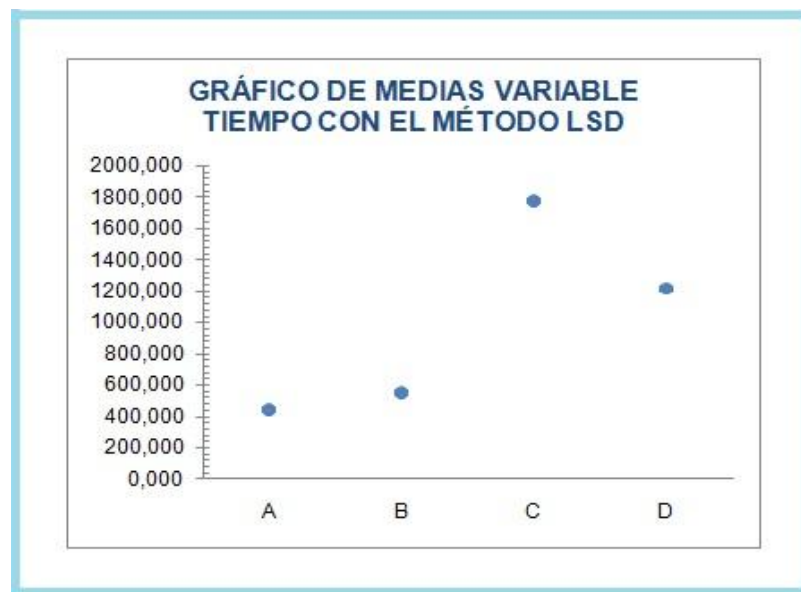
tratamiento de 1200 partículas con 2000 iteraciones continúa siendo el de peores resultados en cuanto a tiempo de ejecución consumido por el algoritmo.

Tabla 21: Análisis de Varianza variable TIEMPO Tamaño de población con iteraciones

Origen de las variaciones	SC	GL	CM	Fo	valor-p	Valor crítico F
Entre grupos	5836821,2	3	1945607,1	16802,5	0,000	3,239
Dentro de grupos	1852,6847	16	115,79279			
Total	5838673,9	19				

Fuente: Autor

Figura 18: Gráfico de medias variable TIEMPO Tamaño de población con iteraciones



Fuente: Autor

5.5.3. Conclusiones sobre el diseño experimental realizado

- Del análisis de varianza realizado se concluye que el diseño cumple con los supuestos de normalidad, igualdad de varianzas e independencia.

- Para la variable COSTO el mejor tratamiento continúa siendo con 1200 partículas y 2000 iteraciones. Las suposiciones acerca de que aumentando el número de iteraciones para 300 partículas podría mejorarse el costo encontrado y de que disminuyendo el número de iteraciones para 1200 partículas posiblemente no se afectaría el rendimiento del algoritmo resultaron falsas.
- El número de iteraciones programadas tiene un efecto importante en el costo hallado cuando el tamaño de la población es igual a 1200 partículas, no tanto así cuando el tamaño se reduce a 300 partículas.
- El número de iteraciones programadas influye de manera directa sobre el tiempo de ejecución empleado por el algoritmo, en forma proporcional.
- Aunque al disminuir el número de iteraciones para 1200 partículas el tiempo se redujo considerablemente, su valor promedio continuó siendo significativamente superior al valor promedio consumido por 300 partículas. Como era de esperarse el tiempo para 300 partículas con un mayor número de iteraciones aumentó aunque esto no produjo ningún efecto en la variable de respuesta COSTO.
- Para el problema estudiado de formación de celdas, es mucho más importante que el algoritmo encuentre la mejor solución posible, sin importar el tiempo que tarde en encontrarla. Por esta razón, pese a reportar los mayores valores de consumo de tiempo, el tamaño de la población seleccionado corresponde a 1200 partículas, pues las soluciones encontradas con este valor son significativamente más económicas que las encontradas con cualquier tamaño de población inferior a este valor. Son tan buenos los resultados obtenidos con este valor que no se hace necesario experimentar con valores mayores que en todo caso sí consumirían mucho más tiempo del ya consumido por las 1200 partículas.
- Aún reportando los mayores valores de tiempo de ejecución, éste no supera los 30 min en promedio, lo cual es una buena estadística si se considera el tamaño y la complejidad del problema que se está resolviendo con el algoritmo.

6. RESULTADOS DETALLADOS DEL ALGORITMO PSO AJUSTADO

Del ajuste de parámetros desarrollado en la sección anterior se determinó que los valores de los mismos que mejor rendimiento le generan al algoritmo, y con los cuales se realizará la experiencia computacional de comparación entre meta-heurísticas son (ver Tabla 22):

Tabla 22: Valores de los parámetros ajustados

PARÁMETRO	VALOR AJUSTADO
COEFICIENTE C1	4
COEFICIENTE C2	2
PESO INERCIAL W	0.9
TAMAÑO DE POBLACIÓN	1200 partículas
NÚMERO DE ITERACIONES	2000

Fuente: Autor

Los resultados por réplica del algoritmo ajustado para el problema descrito en la sección 5 se muestran en la Tabla 23

Tabla 23: Resultados por réplica algoritmo PSO ajustado

RESPUESTA OBSERVADA	PROMEDIO	RESULTADOS POR RÉPLICA				
		REP1	REP2	REP3	REP4	REP5
COSTO	22222,8	22251	22209	22219	22198	22237
TIEMPO	1779,5	1751,7	1776,9	1782,3	1795	1791,6

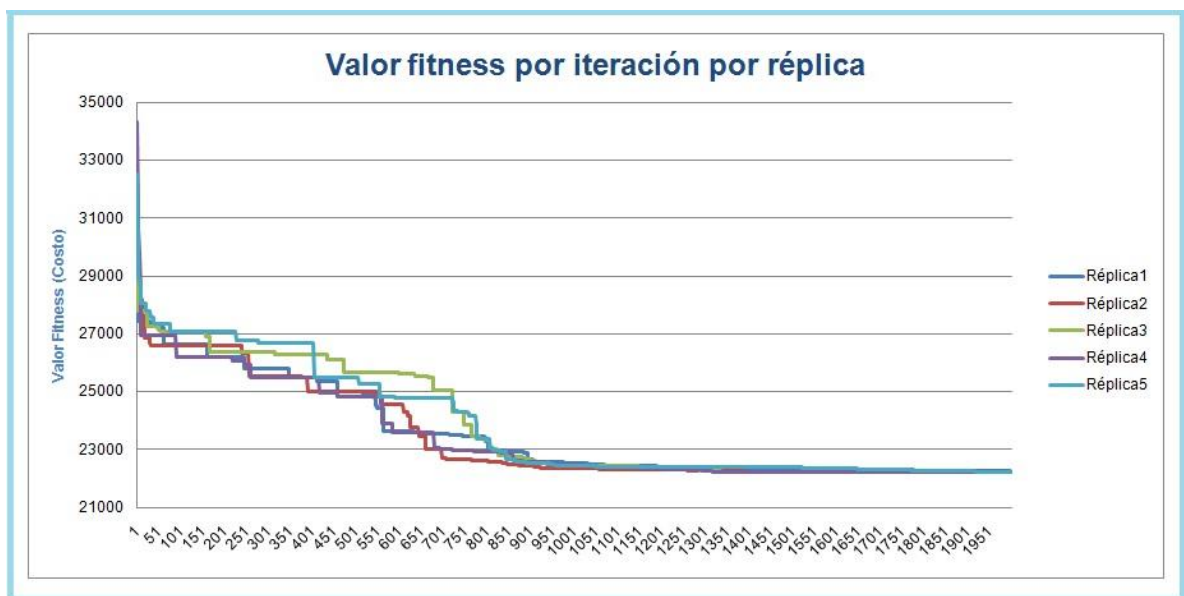
Fuente: Autor

De la Tabla 23 se observa que la mejor solución se encontró en la réplica 4 y corresponde a un valor de 22198 UM, con un tiempo de ejecución de 1795 s.

Sabiendo que el costo óptimo hallado por medio de una metodología exacta⁴² fue de 22006 UM, se concluye que el rendimiento del algoritmo es considerablemente alto pues la solución encontrada por el mismo es muy cercana a la solución óptima del problema planteado.

A manera de información, en la Figura 19 se presenta la gráfica de valor fitness encontrado en cada iteración del algoritmo por réplica. En ella puede verse de manera más detallada la capacidad del algoritmo de encontrar mejores soluciones a medida que el cúmulo se mueve, influenciado por los mejores valores encontrados en cada iteración realizada.

Figura 19: Valor fitness por iteración por réplica



Fuente: Autor

Finalmente, en las Tablas 24 y 25 se muestra la configuración (asignación de celdas) de la mejor solución encontrada por el cúmulo y la asignación de máquinas correspondiente a dicha configuración, respectivamente. Esta

⁴² Ver en (Gómez Gómez, 2007)

configuración corresponde al mejor valor 'gbest' encontrado por el algoritmo en las cinco réplicas realizadas, la cual genera un costo de 22198 UM, encontrado en la réplica 4 a partir de la iteración número 1840.

Tabla 24: Mejor configuración de celdas encontrada por el algoritmo ajustado

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	2	1	3	3	4	4	3	2	3	3	1	4	2	3
2	4	2	4	3	3	4	1	3	2	3	4	2	4	2	3
3	4	2	4	3			3		2	4	4	2	4	2	
4		2	2	4						4	4	2	4		
5		2								4	4				
6		2								4	4				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	2	3	4	2	4	3	4	4	4	2	2	3	4	4
2	2	2	3	2	2	4	3	4	4	4	2	2	4	4	4
3	2		2	2		2	1	4	4	1	2		3		1
4	2			4			3		3				3		2
5							3		2						
6							3		2						

Fuente: Autor

Tabla 25: Asignación de máquinas mejor solución encontrada

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	0	0	0	0	1	0	0	1	0	0	2
2	1	1	1	1	0	2	1	1	1	1	10
3	0	1	1	1	1	0	1	0	1	2	8
4	1	1	1	1	1	1	1	1	1	1	10
Total	2	3	3	3	3	3	3	3	3	4	

Fuente: Autor

En la Tabla 24 puede verse por ejemplo que la operación 1 del producto fue asignada a la celda de fabricación 4. De hecho todas las operaciones del producto 1 se realizan en la misma celda de manufactura (4) lo cual implica que para este producto específico no existen transferencias intercelulares. De la misma manera,

todas las operaciones requeridas para la elaboración del producto 2 fueron asignadas a la misma celda de manufactura, en este caso la celda 2. Para el producto 3, la primera operación fue asignada a la celda de fabricación 1 mientras que la operación 2 del mismo producto fue asignada a la celda de fabricación 4, y así sucesivamente.

Por otro lado, en la Tabla 25 se observa por ejemplo que la celda de fabricación 1 consta de una máquina tipo 5 y una máquina tipo 8 para la elaboración de los productos asignados a ella. Entre tanto la celda de fabricación 2 requiere de un ejemplar de máquinas tipo 1, 2, 3, 4, 7, 8, 9 y 10 y de dos ejemplares de máquina tipo 6. Adicionalmente, de la última columna, denominada 'Total', se puede comprobar que la asignación de máquinas resultante cumple efectivamente con las restricciones de tamaño de celda establecidas (mínimo dos máquinas y máximo diez máquinas por celda de fabricación).

7. EXPERIENCIA COMPUTACIONAL: COMPARACIÓN ENTRE MODELOS

La experiencia computacional de este trabajo de investigación se realizó en un procesador AMD Athlon II Dual-Core M320 de 2.10 Ghz con 3 GB de memoria RAM y 300Gb de disco duro.

El número de productos a fabricar, de máquinas a utilizar y de celdas a conformar para cada uno de los problemas a resolver se tomaron de la adaptación hecha en (Gómez Gómez, 2007) a partir de los estudios desarrollados en (Mungwatanna, 2000) y (Vakharia & Chang, 1997). Los demás datos de entrada definidos en la formulación matemática se generaron de manera aleatoria, bajo las siguientes indicaciones (tomadas de los mismos autores):

- La demanda de cada producto se generó de acuerdo con una distribución discreta uniforme entre 10 y 25 lotes.
- La longitud de la secuencia de producción de cada producto se generó de forma aleatoria utilizando una distribución de probabilidad especial entre 2 y 6 operaciones, como se muestra en la Tabla 26.

Tabla 26: Distribución de probabilidad para el número de operaciones

Número de operaciones	Probabilidad
2	30%
3	30%
4	20%
5	10%
6	10%

Fuente: Autor

- La máquina específica empleada para cada operación de cada producto se fijó de acuerdo a una distribución de probabilidad discreta uniforme entre 1 y M.
- Los tiempos de fabricación de un lote de cualquier producto por operación se generaron de forma aleatoria utilizando una distribución discreta uniforme entre 1 y 10 minutos.
- Para el tiempo disponible de máquina se asumió una jornada laboral de 8 horas diarias, 5 días a la semana y 12 semanas por período (3 meses) para un total de 480 horas por período por tipo de máquina.
- El costo del transporte intercelular para cualquier lote de producto fue de 1 unidad monetaria.
- El costo de máquina por período por tipo de máquina se generó aleatoriamente con una distribución discreta uniforme entre 100 y 2000 unidades monetarias.
- El tamaño mínimo y máximo de celda se fijó en 2 y 10 máquinas respectivamente.

En la tabla 27 se consolida la configuración particular de cada uno de los seis problemas resueltos por medio de las dos meta-heurísticas.

Tabla 27: Configuración de los seis problemas resueltos

Atributo	P1	P2	P3	P4	P5	P6
Productos	30	30	40	40	50	50
Tipos de máquina	10	10	10	10	20	20
Número de celdas	4	4	5	5	6	6

Fuente: Autor

Los datos completos de los seis problemas generados y resueltos pueden ser consultados en los anexos O, P, Q, R, S y T.

En la Tablas 28 se muestran los mejores resultados por problema obtenidos por los dos algoritmos, indicando el número de réplica en los que se obtuvieron, en

términos tanto del costo de la mejor solución encontrada (COSTO) como del tiempo de ejecución empleado (TIEMPO). Adicionalmente, se presenta el costo hallado por un método exacto de programación lineal entera mixta desarrollado en el lenguaje de programación GAMS a través del solver denominado CPLEX⁴³ luego de dos horas de corrida de este algoritmo.

Tabla 28: Mejores resultados por algoritmo

Problema	PSO			AG.			Costo Met. Exacto
	Costo	Tiempo	Réplica	Costo	Tiempo	Réplica	
1	21581	1626	5	21544	728,1	2	21349
2	33732	1721,1	1	33699	652,6	2	33448
3	36115	2186,7	2	35917	679,3	3	33678
4	31392	2104,3	3	31436	857,3	2	31038
5	46689	5781,2	3	48025	1400,8	3	46084
6	48624	4345,0	4	48919	1142,7	4	48019

Fuente: Autor

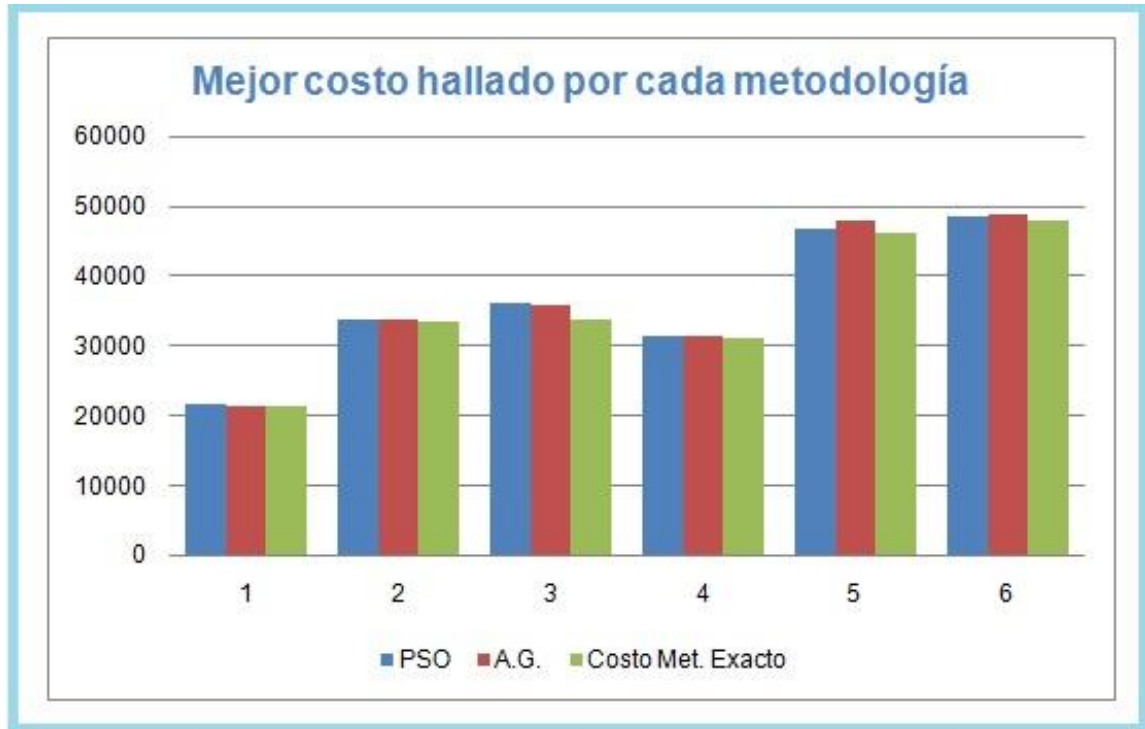
Los resultados completos por réplica de la experiencia computacional realizada pueden ser consultados en el ANEXO U.

7.1. RESULTADOS EN TÉRMINOS DEL COSTO

En la Figura 20 se observan los resultados obtenidos por cada meta-heurística y sus diferencias en cuanto al desempeño, en cada uno de los problemas resueltos, en términos del costo de la mejor solución encontrada. En esta gráfica no se aprecian diferencias notables en el rendimiento de las dos meta-heurísticas y aunque el costo hallado por el método exacto siempre es menor en todos los problemas resueltos, no está muy lejos de las soluciones halladas por las metodologías meta-heurísticas implementadas.

⁴³ Consultar en (Gómez Gómez, 2007)

Figura 20: Mejores resultados en cuanto al costo por metodología



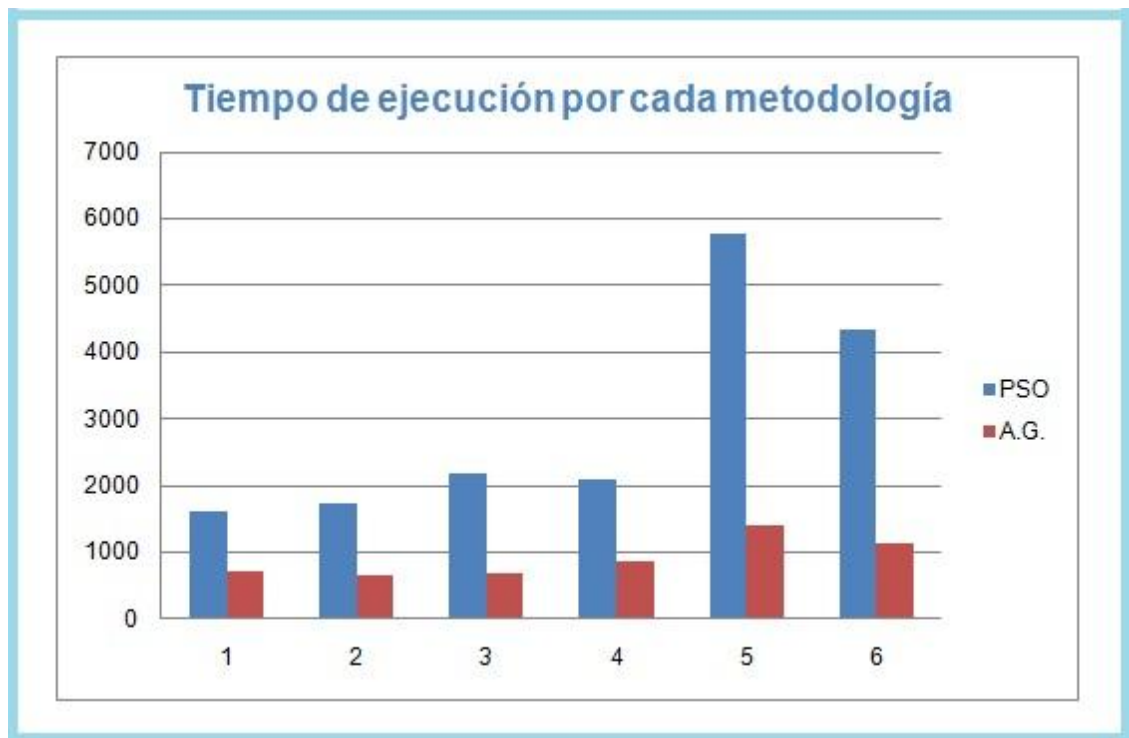
Fuente: Autor

En los primeros tres problemas el algoritmo genético obtuvo mejores soluciones que las halladas por el algoritmo PSO, sin embargo, esta diferencia en los costos resultó siempre menor al 0,6%. Entre tanto, en los últimos tres problemas el algoritmo PSO tuvo una leve mejora y sus costos hallados se mantuvieron por debajo de los hallados por el algoritmo genético. En este caso las diferencias en los costos fueron cercanas al 3%, específicamente en el caso del problema 5. Se concluye entonces que el algoritmo PSO mostró un mejor desempeño en la búsqueda de soluciones factibles a los problemas de formación de celdas planteados. Esto no quiere decir que el desempeño del algoritmo genético sea deficiente, puesto que también llegó a mejores soluciones. Se recalca que la diferencia en el desempeño de los algoritmos fue de tan solo un 3%.

7.2. RESULTADOS EN TÉRMINOS DEL TIEMPO DE EJECUCIÓN

En la Figura 21 se presenta el gráfico comparativo del rendimiento de las dos meta-heurísticas en términos del tiempo de ejecución empleado.

Figura 21: Tiempos de ejecución empleados por metodología por problema



Fuente: Autor

En el caso del rendimiento de los algoritmos en cuanto a tiempo de ejecución consumido los resultados son irrefutables. El rendimiento del algoritmo PSO es altamente deficiente respecto al rendimiento logrado por el algoritmo genético en la resolución de los problemas de aplicación planteados.

Estos tiempos de ejecución para el algoritmo PSO son en promedio 3,1 veces superiores a los consumidos por el algoritmo genético, llegando incluso a ser 4,1

veces mayor, como en el caso del problema 5, como se puede observar en la Tabla 29.

Tabla 29: Comparación tiempo de ejecución consumido por las metodologías

Problema	Tiempo		Razón
	PSO	A.G.	
1	1626,0	728,1	2,2
2	1721,1	652,6	2,6
3	2186,7	679,3	3,2
4	2104,3	857,3	2,5
5	5781,2	1400,8	4,1
6	4345,0	1142,7	3,8

Fuente: Autor

7.3. RESULTADOS DETALLADOS POR PROBLEMA

A continuación se presentan los resultados del primer problema de aplicación, a manera de ejemplo. Los resultados del algoritmo PSO y Algoritmo genético correspondientes a los otros problemas se encuentran incluidos en los anexos V, W, X, Y, Z y AA, BB, CC, DD, EE respectivamente.

La Tabla 30 muestra los resultados detallados de las corridas realizadas por los algoritmos, en términos del costo.

En esta tabla puede verse que el algoritmo PSO tuvo un mejor desempeño, con un promedio de 21612,6 UM. Además, la desviación estándar reportada por las corridas de este algoritmo estuvo poco más de tres veces por debajo de la desviación estándar del algoritmo genético, lo cual da una medida de la buena consistencia del algoritmo PSO en la búsqueda de soluciones a un mismo problema. Aún así, el menor de todos los costos encontrados en las corridas

realizadas se obtuvo con el algoritmo genético, en la segunda réplica, correspondiente a 21544 UM.

Tabla 30: Resultados detallados Problema 1

Problema	Réplica	PSO	AG.	Prom PSO	Desv PSO	Prom AG.	Desv AG.
1	1	21599	21677	21612,6	39,253025	21693,4	166,73722
	2	21585	21544				
	3	21621	21586				
	4	21677	21971				
	5	21581	21689				

Fuente: Autor

En las Figuras 22 y 23 se muestra la evolución por réplica por iteración y por generación para el algoritmo PSO y algoritmo genético, respectivamente.

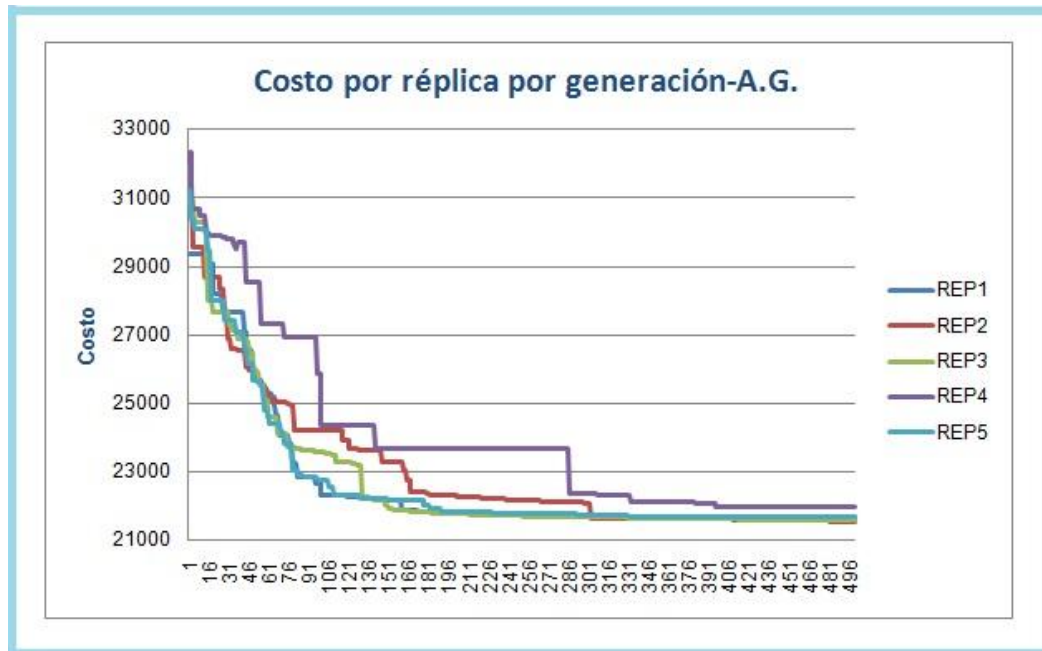
De la gráfica se ve que el algoritmo PSO tiene muy buena capacidad de convergencia progresiva hasta la iteración 650. A partir de ese momento, el cúmulo disminuye su ritmo de exploración y aumenta su capacidad de explotación de la mejor solución encontrada hasta ese momento. Vuelve a tener algunos cambios en el valor fitness de la mejor solución encontrada hasta converger casi en el mismo valor de aptitud en sus 5 réplicas.

Figura 22: Costo por réplica por iteración, problema 1, algoritmo PSO



Fuente: Autor

Figura 23: Costo por réplica por generación, problema 1, Algoritmo genético



Fuente: Autor

Por su parte el algoritmo genético también muestra una muy buena capacidad de convergencia, más radical que progresiva pues los valores de los costos encontrados cambian de manera más abrupta. Tiene importantes cambios en los valores de aptitud hasta la generación 180 y en ese momento, al igual que el algoritmo PSO, termina por encontrar algunas soluciones mejores que la encontrada hasta ese momento, hasta llegar al mejor valor reportado por el algoritmo. Se puede ver que la convergencia por cada réplica es mucho más dispersa en comparación a las del algoritmo PSO.

Finalmente, las tablas 31 y 32 muestran los resultados obtenidos por el algoritmo genético en la réplica 2 que corresponden a la mejor solución encontrada para el problema 1. El costo total resultó ser de 21544 UM, encontrado en la generación 500 luego de 728,1 s de ejecución del algoritmo.

Tabla 31: Configuración de la mejor solución encontrada al problema 1

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	3	2	1	1	1	2	2	1	3	2	3	2	1
2	2	3	3	2	1	2	2	2	2	1	3	2	3	2	1
3		4	2		1	2	1	3		1		2	3	2	
4			2		2	2		3		3				2	
5														1	
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	3	2	4	2	3	2	1	3	2	2	3	1	1	1	2
2	3	3	3	2	1	2	3	3	2	2	3	1	2	1	4
3		3	1			4	2	3	1	2	3		2	1	
4			3				2	3						1	
5			3				2	3						1	

Fuente: Autor

La tabla 31 muestra la mejor configuración de celdas encontrada mientras que la tabla 32 muestra las necesidades de máquina correspondientes a dicha

configuración. En esta misma tabla puede constatarse que la solución encontrada es factible al no exceder las restricciones de tamaño de celda impuestas.

Tabla 32: Necesidades de máquina de la mejor solución encontrada para el problema 1

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	0	1	1	0	0	1	1	1	0	1	6
2	1	0	1	1	1	0	1	1	1	1	8
3	0	1	2	1	1	1	0	0	1	1	8
4	1	1	0	0	0	0	0	0	0	0	2
Total	2	3	4	2	2	2	2	2	2	3	

Fuente: Autor

CONCLUSIONES

- ✓ El desempeño de las dos meta-heurísticas, en cuanto a búsqueda de la mejor solución posible a problemas de formación de celdas de manufactura planteados, es igualmente eficiente. En unos casos el algoritmo genético reportó mejores soluciones, y en otros fue el algoritmo PSO el que presentó el mejor desempeño. Sin embargo, las diferencias encontradas en las respuestas no fueron superiores al 3%.
- ✓ En general, el desempeño de las dos meta-heurísticas resultó ser muy eficiente en la búsqueda de soluciones cercanas a la óptima, pues los resultados arrojados por ambos métodos se mantuvieron en todos los casos de aplicación muy cercanos a la solución encontrada por un modelo exacto de programación.
- ✓ El rendimiento de las dos meta-heurísticas, en cuanto a tiempo de ejecución empleado resultó ser altamente deficiente para el algoritmo PSO en comparación con los tiempos computacionales consumidos por el algoritmo genético. El tiempo de ejecución de PSO llegó a estar por encima del tiempo de ejecución del algoritmo genético hasta en un 400%, y siempre se mantuvo por encima del 200% respecto al mismo.
- ✓ De todo el trabajo de investigación realizado se concluye que para la resolución del problema de formación de celdas de manufactura es más recomendable implementar algoritmo genético, pues llega a soluciones muy cercanas, algunas veces mejores que las encontradas por medio de algoritmo PSO, pero utilizando tiempos computacionales considerablemente menores.
- ✓ La convergencia del algoritmo genético es más consistente y progresiva que la del algoritmo PSO. El algoritmo PSO, aunque converge en soluciones parecidas a las del algoritmo genético y de hecho muy cercanas a las óptimas, tiende a quedarse estancado en óptimos locales durante un número considerable de iteraciones.

- ✓ El parámetro de PSO que mayor influencia ejerce sobre el rendimiento que pueda lograr este algoritmo es el número de partículas que se asignen al cúmulo.
- ✓ A medida que aumenta el número de celdas a conformar, es decir el rango de posibles valores que pueda tomar la partícula, el algoritmo PSO comienza a mostrar un mejoramiento notable en su desempeño.
- ✓ La interfaz del algoritmo PSO y su forma de programación para guardar los resultados y cualquier otro dato generado durante su funcionamiento que se considere importante para la investigación es mucho más fácil, rápida y amigable con el usuario que la correspondiente al algoritmo genético.

RECOMENDACIONES

- ✓ Se recomienda utilizar una variante más avanzada que el algoritmo PSO clásico, o bien un híbrido con otra meta-heurística, que permita llegar a soluciones tan buenas como las encontradas por el algoritmo clásico pero en tiempos computacionales mucho más eficientes y competentes con los de otras meta-heurísticas, en este caso algoritmo genético.
- ✓ Tener cuidado de programar una velocidad de partícula acorde con los rangos de posibles valores que pueda tomar cada dimensión de la misma, en este caso, el número de celdas a conformar. Lo que sucede es que al aplicar la velocidad, los valores de las celdas que queden por fuera del rango especificado en la toolbox son forzados a tomar el valor límite del rango, recargando de esta manera la asignación de máquinas correspondiente a la primera y última celda de fabricación, lo cual afecta negativamente la posibilidad de encontrar soluciones factibles al problema.
- ✓ Al ejecutar cualquiera de los algoritmos, evitar en lo posible tener en funcionamiento otros programas que puedan disminuir el rendimiento del procesador y con ello aumentar el tiempo de ejecución consumido por los algoritmos.

BIBLIOGRAFÍA

A. C., & Lozano, S. (2006). A particle swarm optimization algorithm for part-machine grouping. *Robotics and Computer-Integrated Manufacturing*, 22 , 468-474.

Anvari, M., Mehrabad, M. S., & Banzinpour, F. (2010). Machine-part cell formation using a hybrid particle swarm optimization. *International Journal of Advanced Manufacturing Technology*, 47 , 745-754.

Caprihan, R., Slomp, J., Gusaran, & Agarwal, K. (2009). A quantum particle swarm optimization approach for the design of virtual manufacturing cells. *Proceedings of the IEEE IEEM* , 125-129.

Casallas Fonseca, C. E., & Vargas Aristizábal, S. F. (s.f.). ACIS. Recuperado el 13 de 02 de 2011, de ACIS: http://www.acis.org.co/fileadmin/Revista_113/uno.pdf

Castillo, E., Conejo, A. J., Pedregal, P., García, R., & Alguacil, N. (20 de 02 de 2002). *Universidad Santa maría - Escuela de Administración y Contaduría*. Recuperado el 12 de 02 de 2011, de Universidad Santa maría - Escuela de Administración y Contaduría: http://www.investigacion-operaciones.com/ARCHIVOS_LIBRO/LibroCompleto.pdf

Chan, C. Y., Lam, F. W., & Lee, C. P. (1999). Considerations for using cellular manufacturing. *Journal of Materials Processing Technology* 96 , 182-187.

Chan, F. S., Lau, K. W., & Chan, P. Y. (2004). A holistic approach to manufacturing cell formation: incorporation of machine flexibility and machine aggregation. *Journal of Engineering Manufacture*, 218 , 1279-1296.

Chen, W. H., & Srivastava, B. (1994). Simulated annealing procedures for forming machine cells in group technology. *European Journal of Operational Research*, 75 , 100-111.

Crainic, T. G., & Toulouse, M. (s.f.). *CiteSerrXbeta*. Recuperado el 13 de 02 de 2011, de *CiteSerrXbeta*: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.322>

Deljoo, V., Al-e-hashem, S. J., Deljoo, F., & Aryanezhad, M. B. (2010). Using genetic algorithm to solve dynamic cell formation problem. *Applied Mathematical Modelling*, 34 , 1078-1092.

Díaz Ortega, C. R. (s.f.). Fabricación automatizada: Ingeniería concurrente. Departamento de Tecnología- Universitat Jaume I de Castellón.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. Massachusetts Institute of Technology Press.

Durán, O., Rodríguez, N., & Consalter, L. A. (12 de 03 de 2008). *IEEE Xplore Digital Library*. Recuperado el 12 de 02 de 2011, de IEEE Xplore Digital Library: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4470352>

Filho, E. G., & Tiberti, A. J. (2006). A group genetic algorithm for the machine cell formation problem. *International Journal of Production Economics* , 1-21.

Flores Mendoza, J. I., & Mezura Montes, E. (2008). Looking Inside Particle Swarm Optimization in Constrained Search Spaces. *Proceedings of the 7th Mexican International Conference on Artificial Intelligence* (págs. 451–461, Vol. 5317). México: Springer-Verlag.

Foulds, L. R., French, A. P., & Wilson, J. M. (2006). The sustainable cell formation problem: manufacturing cell creation with machine modification cost. *Computers and Operations Research* 33 , 1010-1032.

García Nieto, J. M. (14 de 09 de 2006). *NEO: Networking and Emerging Optimization*. Recuperado el 12 de 02 de 2011, de NEO: Networking and Emerging Optimization: http://neo.lcc.uma.es/staff/jmgn/doc/Memoria_PFC_JMGN.pdf

Garro Retamal, A., & Domínguez Cuadrado, D. (s.f.). *Portal de Docencia de la ETESIT - Universidad Rey Juan Carlos*. Recuperado el 23 de 02 de 2011, de Portal de Docencia de la ETESIT - Universidad Rey Juan Carlos: <http://docencia.etsit.urjc.es/moodle/file.php/101/transparencias/PSO.pdf>

Giri, R., & Srinivas, K. (2007). An optimal design approach for a cellular manufacturing system. *Journal of Engineering Manufacture*, 22 , 1101-1106.

Glover, F. (1989). Tabu Search - Part I. *ORSA Journal on Computing* , 1:3, 190-206.

Gómez Gómez, S. (2007). *Formulación y evaluación de un modelo heurístico para el diseño de Sistemas de Manufactura Celular mediante la implementación de Algoritmos Genéticos*. Bucaramanga, Colombia: Trabajo de grado Universidad Industrial de Santander.

Grammatoula, P., & Wilson, J. M. (2009). The evolution of cell formation problem methodologies based on recent studies (1997-2008): review and directions for future research. *European Journal of Operational Research* , 509-521.

Gutiérrez Pulido, H., & De la Vara Salazar, R. (2008). *Análisis y diseño de experimentos*. México D.F.: McGraw-Hill Interamericana.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. University of Michigan Press.

Kennedy, J. (1997). The Particle Swarm: Social Adaptation of Knowledge. *IEEE International Conference on Evolutionary Computation* , 303-308.

Kennedy, J., & Eberhart, R. (1995). *Particle Swarm Optimization*. Piscataway, New Jersey: In Proceedings of the 1995 IEEE International Conference on Neural Networks.

Kennedy, J., Eberhart, R., & Shi, Y. (2001). *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers.

Kim, C. O., Baek, J. G., & Baek, J. K. (2004). A two phases heuristic algorithm for cell formation problem considering alternative part routes and machine sequences. *International Journal of Production Research* 42 , 3911-3927.

Kirkpatrick, S., Gelatt, C., Vecchi, J., & Vecchi, M. (13 de Mayo de 1983). Optimization by Simulated Annealing. *Science* , Vol 220 No. 4598 pp. 671-680.

Liu, C. M., & Wu, J. K. (1993). Machine cell formation using the Simulated Annealing algorithm. *International Journal of Computers & Operations Manufacturing* 6 , 335-349.

Mahapatra, S. S., & Sudhakara Pandian, R. (2008). Genetic cell formation using ratio level data in cellular manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 38 , 630-640.

Martí, R. (2003). Procedimientos meta-heurísticos en optimización combinatoria. *Matemáticas* 1 , 3-62.

Mehdizadeh, E., & Tavakkoli-Moghaddam, R. (2009). A fuzzy particle swarm optimization algorithm for a cell formation problem. *Proceedings of IFSA-EUSFLAT* , 1768-1772.

Mestre Genovart, J., & Parets Fiol, T. (16 de 03 de 2009). *Departament de Ciències Matemàtiques i Informàtica - Universitat de les Illes Balears*. Recuperado el 10 de 02 de 2011, de Departament de Ciències Matemàtiques i Informàtica - Universitat de les Illes Balears: http://dmi.uib.es/~burguera/download/sif/tecnologia_grups_20809.pdf

Ming, L. C., & Ponnambalama, S. G. (2008). A hybrid GA/PSO for the concurrent design of cellular manufacturing system. *IEEE International Conference on Systems, Man and Cybernetics* , 1855-1860.

Mukattash, A. M., Adil, M. B., & Tahboub, K. K. (2002). Heuristic approaches for part assignment in cell formation. *Computers & Industrial Engineering* 42 , 329-341.

Mungwatanna, A. (2000). *Design of Cellular Manufacturing Systems for Dynamic and Uncertain Production requirements with Presence of Routing Flexibility*. Blacksburg, EEUU: Tesis doctoral. Virginia Polytechnic Institute and State University.

Neto, A. P., & Filho, E. G. (2010). A simulation-based evolutionary multiobjective approach to manufacturing cell formation. *Computers & Industrial Engineering*, 59 , 64-74.

Onwubolu, G. C., & Songore, V. (2000). A tabu search approach to cellular manufacturing systems. *Production Planning & Control*, 11 , 153-164.

Osman, I. H., & Kelly, J. P. (1996). *Meta-heuristics: Theory and Applications*. Boston: Ed. Kluwer Academic.

Prabhakaran, G., Muruganandam, A., Asokan, P., & Girish, B. S. (2005). Machine cell formation for cellular manufacturing systems using an ant colony system approach. *International Journal of Advanced Manufacturing Technology*, 25 , 1013-1019.

Rajagopalan, R., & Fonseca, D. J. (2005). Volume sensitivity analysis for manufacturing cells: A genetic algorithm approach. *Journal of Advanced Manufacturing Systems*, 4 , 167-183.

Salto, C. (2009). *Meta-heurísticas híbridas paralelas para problemas industriales de corte, empaquetado y otros relacionados*. San Luis, Argentina: Tesis de Doctorado Universidad de San Luis.

Sampol Aznar, P., & Carmona Gómez, C. (11 de 03 de 2009). *Departament de Ciències Matemàtiques i Informàtica - Universitat de les Illes Balears*. Recuperado el 10 de 02 de 2011, de Departament de Ciències Matemàtiques i Informàtica - Universitat de les Illes Balears: http://dmi.uib.es/~burguera/download/sif/tecnologia_grups0809.pdf

Selim, M., Askin, R. G., & Vakharia, A. J. (1998). Cell formation in Group Technology: review evolution and directions for future research. *Computers & Industrial Engineering* 34 , 3-20.

Solimanpur, M., Saeedi, S., & Mahdavi, I. (2010). Solving cell formation problem in cellular manufacturing using ant-colony-based optimization. *International Journal of Advanced Manufacturing Technology*, DOI , 10.1007/s 00170-010-2587-5.

Spiliopoulos, K., & Sofianopoulo, S. (2008). An efficient ant colony optimization system for the manufacturing cells formation problem. *International Journal of Advanced Manufacturing Technology*, 36 , 589-597.

Spiliopoulos, K., & Sofianopoulo, S. (2003). Designing manufacturing cells: a staged approach and a Tabu search algorithm. *International Journal of Production Research*, 41 , 2531-2546.

Su, C. T., & Hsu, C. M. (1998). Multi-objective machine-part cell formation through parallel simulated annealing. *International Journal of Production Research*, 36 , 2185-2207.

Tamal, G., Sourav, S., Manojit, C., & Pranab, K. D. (01 de 08 de 2010). *GrowingScience*. Recuperado el 13 de 02 de 2011, de GrowingScience: http://growingscience.com/ijiec/Vol2/IJIEC_2010_28.pdf

Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Safaei, N., & Azaron, A. (2007). Solving a dynamic cell formation problem using meta-heuristics. *Applied Mathematics and Computation*, 170 , 761-780.

Tunnukij, T., & Hicks, C. (2009). Cell formation in group technology: a combinatorial search approach. *International Journal of Production Research*, 35 , 2025-2043.

Vakharia, A. J., & Chang, Y. L. (1997). Cell formation in group technology: a combinatorial search approach. *International Journal of Production Research*, 7 , 2025-2043.

Won, Y., & Lee, K. C. (2004). Modified p-median approach for efficient GT cell formation. *Computer & Industrial Engineering* 46 , 495-510.

Wu, X., Chu, C. H., & Wang, Y. W. (2002). A genetic algorithm for integrated cell formation and layout decisions. *IEEE Proceedings of the 2002 congress on Evolutionary Computation*, 2 , 1866-1871.

Yasuda, K., & Yin, Y. (2001). A dissimilarity measure for solving the cell formation problem. *Computers & Industrial Engineering* 39 , 1-17.

Zolfaghari, S., & Liang, M. (2003). A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. *Computers & Industrial Engineering*, 45 , 713-731.

Zolfaghari, S., & Liang, M. (2003). Machine cell/part family formation considering processing times and machine capacities: a simulated annealing approach. *Computers & Industrial Engineering*, 34 , 813-823.

ANEXOS

ANEXO A: CÓDIGO DE PROGRAMACIÓN ALGORITMO PSO FICHERO
'PARAMETROSPSO.M'

```
% ANÁLISIS DEL DESEMPEÑO DE DOS MODELOS META-HEURÍSTICOS EN EL PROBLEMA DE
% FORMACIÓN DE CELDAS DE MANUFACTURA

%ALGORITMO PSO

%AUTOR: JOHANNA RODRÍGUEZ LEÓN

%Fichero: parametrospso.m

funcname='OBJETIVO_INDIVIDUO';
D=cell(1,1);
D{1,1}=zeros(6,30);
mv=2;
VarRange=[0.0001 4];
minmax=0;
PSOparams=[50,2000,300,4,2,0.9,0.4,1500,1e-5,2000,NaN,0,0];
plotfcn='goplotpso';
%PSOseedValue;
[optOUT,datos,tr,te]=pso_Trelea_vectorized(funcname,D,mv,VarRange,minmax,...
PSOparams,plotfcn,0);
```

ANEXO B: CÓDIGO DE PROGRAMACIÓN ALGORITMO PSO FICHERO 'PSO_TRELEA_VECTORIZED.M'

```
% ANÁLISIS DEL DESEMPEÑO DE DOS MODELOS META-HEURÍSTICOS EN EL PROBLEMA DE
% FORMACIÓN DE CELDAS DE MANUFACTURA

%ALGORITMO PSO

%AUTOR: JOHANNA RODRÍGUEZ LEÓN

%Fichero: pso_trelea_vectorized.m

function [OUT,datos,varargout]=pso_Trelea_vectorized(funcname,D,varargin)

global Op

load SecOpAG.mat

tic
rand('state',sum(100*clock));

for p=1:size(Op,2)
    Mascara(:,p)=[repmat(1,[Op(p) 1]);repmat(0,[max(Op)-Op(p) 1])];
end

if nargin < 2
    error('Not enough arguments.');
```

```
end

% PSO PARAMETERS
if nargin == 8 % Funcname, D, mv, Varrange, minmax, and psoparams, plotfcn, PSOseedValue
    mv=varargin(1);
    if isnan(mv)
        mv=4;
    end
    VR=varargin(2);
    minmax=varargin(3);
    P = varargin(4); % psoparams
    plotfcn = varargin(5);
    PSOseedValue = varargin(6);
else
    error('Wrong # of input arguments.');
```

```
end

% sets up default pso params
Pdef = [100 2000 24 2 2 0.9 0.4 1500 1e-25 250 NaN 0 0];
Plen = length(P);
P = [P,Pdef(Plen+1:end)];

df = P(1);
me = P(2);
ps = P(3);
ac1 = P(4);
ac2 = P(5);
iw1 = P(6);
iw2 = P(7);
iwe = P(8);
ergrd = P(9);
ergrdep = P(10);
errgoal = P(11);
trelea = P(12);
PSOseed = P(13);
```

```

% error checking
if ((minmax==2) & isnan(errgoal))
    error('minmax= 2, errgoal= NaN: choose an error goal or set minmax to 0 or 1');
end

if (PSOseed==1 & ~exist('PSOseedValue') )
    error('PSOseed flag set but no PSOseedValue was input');
end

if exist('PSOseedValue')
    tmpsz=size(PSOseedValue);
    if size(D,2) < tmpsz(2)
        error('PSOseedValue column size must be D or less');
    end
    if ps < tmpsz(1)
        error('PSOseedValue row length must be # of particles or less');
    end
end

% preallocate variables for speed up
tr = ones(1,me)*NaN;

% take care of setting max velocity and position params here
if length(mv)==1
    velmaskmin = -mv*ones(size(D{1,1},1),size(D{1,1},2)); % min vel
    velmaskmax = mv*ones(size(D{1,1},1),size(D{1,1},2)); % max vel

elseif length(mv)==D
    velmaskmin = repmat(forcerow(-mv),ps,1); % min vel
    velmaskmax = repmat(forcerow( mv),ps,1); % max vel
else
    error('Max vel must be either a scalar or same length as prob dimension D');
end

posmaskmin = repmat(VR(1,1),size(D{1,1},1),size(D{1,1},2)); % min pos
posmaskmax = repmat(VR(1,2),size(D{1,1},1),size(D{1,1},2)); % max pos
posmaskmeth = 3; % 3=bounce method (see comments below inside epoch loop)

% INITIALIZE INITIALIZE INITIALIZE INITIALIZE INITIALIZE INITIALIZE

% initialize population of particles and their velocities at time zero,
% construct random population positions bounded by VR
for i=1:ps
    pos(i) = normmat(rand(size(D{1,1},1),size(D{1,1},2)),VR,1);
    pos(i) = ceil(pos(i));
    pos(i) = pos(i).*Mascara;
end
pos=pos';

% construct initial random velocities between -mv,mv
for i=1:ps
    vel(i) = normmat(rand(size(D{1,1},1),size(D{1,1},2)),[(-mv),(mv)],1);
    vel(i) = vel(i).*Mascara;
end
vel = vel';

% initial pbest positions vals
pbest = pos;

% VECTORIZE THIS, or at least vectorize cost funct call
[ out,AsignacionMaq_Celda,Restriccion ] = feval(funcname,pos);
% returns column of cost values (1 for each particle)

```

```

pbestval=out; % initially, pbest is same as pos

% assign initial gbest here also (gbest and gbestval)
if minmax==1
    % this picks gbestval when we want to maximize the function
    [gbestval,idx1] = max(pbestval);
elseif minmax==0
    % this works for straight minimization
    [gbestval,idx1] = min(pbestval);
    bestAsignacionMaq_Celda = AsignacionMaq_Celda(idx1);
    EstadoRestriccion = Restriccion(idx1,1);
elseif minmax==2
    % this works when you know target but not direction you need to go
    % good for a cost function that returns distance to target that can be either
    % negative or positive (direction info)
    [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal).^2);
    gbestval = pbestval(idx1);
end

% preallocate a variable to keep track of gbest for all iters
bestpos = cell(me,4);
for n=1:me
    bestpos(n,1) = D(1,1)*NaN;
    bestpos(n,2) = NaN;
    bestpos(n,3) = zeros(size(AsignacionMaq_Celda(1,1),1), ...
        size(AsignacionMaq_Celda(1,1),2))*NaN;
    bestpos(n,4) = NaN;
end
gbest = pbest(idx1); % this is gbest position

%tr(1) = gbestval; % save for output
bestpos(1,1) = gbest;
bestpos(1,3) = bestAsignacionMaq_Celda;
bestpos(1,4) = EstadoRestriccion;

for i=1:me % start epoch loop (iterations)

    [out,AsignacionMaq_Celda,Restriccion]=feval(funcname,[pos;gbest]);
    outbestval = out(end,:);
    out = out(1:end-1,:);
    % pbest=MatrizAsignacion;

    tr(i+1) = gbestval; % keep track of global best val
    te = i; % returns epoch number to calling program when done
    bestpos(i,1) = gbest;
    bestpos(i,2) = gbestval;
    bestpos(i,3) = bestAsignacionMaq_Celda;
    bestpos(i,4) = EstadoRestriccion;

    % recalculate best vals
    if minmax == 1
        [gbestval,idx1] = max(pbestval);
    elseif minmax==0
        [gbestval,idx1] = min(pbestval);
        bestAsignacionMaq_Celda = AsignacionMaq_Celda(idx1);
        EstadoRestriccion = Restriccion(idx1);
    elseif minmax==2 % this section needs work
        [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal).^2);
        gbestval = pbestval(idx1);
    end

    gbest = pbest(idx1,:);

```

```

% find particles where we have new pbest, depending on minmax choice
% then find gbest and gbestval
%[size(out),size(pbestval)]
if rstflg == 0
if minmax == 0
[tempi] = find(pbestval>=out); % new min pbestvals
pbestval(tempi,1) = out(tempi); % update pbestvals
pbest(tempi,:) = pos(tempi,:); % update pbest positions

[iterbestval,idx1] = min(pbestval);

if gbestval >= iterbestval
gbestval = iterbestval;
gbest = pbest(idx1);
bestAsignacionMaq_Celda = AsignacionMaq_Celda(idx1);
EstadoRestriccion = Restriccion(idx1,1);
end
elseif minmax == 1
[tempi,dum] = find(pbestval<=out); % new max pbestvals
pbestval(tempi,1) = out(tempi,1); % update pbestvals
pbest(tempi,:) = pos(tempi,:); % update pbest positions

[iterbestval,idx1] = max(pbestval);
if gbestval <= iterbestval
gbestval = iterbestval;
gbest = pbest(idx1,:);
% used with trainpso, for neural net training
% assign gbest to net at each iteration, these interim assignments
% are for plotting mostly
if strcmp(funcname,'pso_neteval')
net=setx(net,gbest);
end
end
end
end

% each epoch get new set of random numbers
for n=1:ps
rannum1(n) = rand([size(D{1,1},1),size(D{1,1},2)]);
end
rannum1=rannum1';
for n=1:ps
rannum2(n)=rand([size(D{1,1},1),size(D{1,1},2)]);
end
rannum2=rannum2';

if trelea == 2
% from Trelea's paper, parameter set 2
vel = 0.729.*vel... % prev vel
+1.494.*rannum1.*(pbest-pos)... % independent
+1.494.*rannum2.*(repmat(gbest,ps,1)-pos); % social
elseif trelea == 1
% from Trelea's paper, parameter set 1
vel = 0.600.*vel... % prev vel
+1.700.*rannum1.*(pbest-pos)... % independent
+1.700.*rannum2.*(repmat(gbest,ps,1)-pos); % social
elseif trelea ==3
% Clerc's Type 1" PSO
vel = chi*(vel... % prev vel
+ac1.*rannum1.*(pbest-pos)... % independent
+ac2.*rannum2.*(repmat(gbest,ps,1)-pos) ); % social
else

```

```

% common PSO algo with inertia wt
% get inertia weight, just a linear funct w.r.t. epoch parameter iwe
if i<=iwe
    iwt(i)=((iw2-iw1)/(iwe-1))*(i-1)+iw1;
else
    iwt(i) = iw2;
end
% random number including acceleration constants
for n=1:ps
    ac11(n)=rannum1(n).*ac1;    % for common PSO w/inertia
end
ac11=ac11';
for n=1:ps
    ac22(n)=rannum2(n).*ac2;
end
ac22=ac22';

for n=1:ps
    vel(n)=iwt(i).*vel(n)...           % prev vel
    +ac11(n).*(pbest(n)-pos(n))...    % independent
    +ac22(n).*(gbest-pos(n));        % social
end
end

% limit velocities here using masking
for n=1:ps
    vel(n)=(vel(n)<=velmaskmin).*velmaskmin+(vel(n)>velmaskmin).*vel(n);
    vel(n)=(vel(n)>=velmaskmax).*velmaskmax+(vel(n)<velmaskmax).*vel(n);
    vel(n)=vel(n).*Mascara;
end

% update new position (PSO algo)
for n=1:ps
    pos(n)=pos(n).*Mascara+vel(n);
end

% position masking, limits positions to desired search space
% method: 0) no position limiting, 1) saturation at limit,
%         2) wraparound at limit , 3) bounce off limit

for n=1:ps
    minposmask_throwaway(n)=pos(n)<=posmaskmin; % these are psXD matrices
    minposmask_keep(n)=pos(n)>posmaskmin;
    maxposmask_throwaway(n)=pos(n)>=posmaskmax;
    maxposmask_keep(n)=pos(n)<posmaskmax;

    if posmaskmeth == 1
        % this is the saturation method
        pos = ( minposmask_throwaway.*posmaskmin ) + ( minposmask_keep.*pos );
        pos = ( maxposmask_throwaway.*posmaskmax ) + ( maxposmask_keep.*pos );
    elseif posmaskmeth == 2
        % this is the wraparound method
        pos = ( minposmask_throwaway.*posmaskmax ) + ( minposmask_keep.*pos );
        pos = ( maxposmask_throwaway.*posmaskmin ) + ( maxposmask_keep.*pos );
    elseif posmaskmeth == 3
        % this is the bounce method, particles bounce off
        % the boundaries with -vel
        pos(n)=(minposmask_throwaway(n).*posmaskmin)+(minposmask_keep(n).*pos(n));
        pos(n)=(maxposmask_throwaway(n).*posmaskmax)+(maxposmask_keep(n).*pos(n));
        pos(n)=ceil(pos(n)).*Mascara;

        vel(n)=(vel(n).*minposmask_keep(n))+(-vel(n).*minposmask_throwaway(n));
    end
end

```

```

        else
            vel(n)=(vel(n).*maxposmask_keep(n))+(-vel(n).*maxposmask_throwaway(n));
        end
    end
end

% check for stopping criterion based on speed of convergence to desired
% error
tmp1 = abs(tr(i) - gbestval);
if tmp1 > ergrd
    cnt2 = 0;
elseif tmp1 <= ergrd
    cnt2 = cnt2+1;
    if cnt2 >= ergrdep
        if plotflg == 1
            fprintf(message,i,gbestval);
            disp(' ');
            disp(['--> Solution likely, GBest hasn't changed by at least ',...
                num2str(ergrd),' for ',...
                num2str(cnt2),' epochs.']);
            %eval(plotfcn);
        end
        break
    end
end
end

% this stops if using constrained optimization and goal is reached
if ~isnan(errgoal)
    if ((gbestval<=errgoal) & (minmax==0)) | ((gbestval>=errgoal) & (minmax==1))

        if plotflg == 1
            fprintf(message,i,gbestval);
            disp(' ');
            disp(['--> Error Goal reached, successful termination!']);

            %eval(plotfcn);
        end
        break
    end
end % end ~isnan if

end % end epoch loop

%% clear temp outputs
% evalin('base','clear temp_pso_out temp_te temp_tr;');
% output & return
OUT(1) = gbest;
OUT(2) = gbestval;
OUT(3) = bestAsignacionMaq_Celda;
OUT(4) = EstadoRestriccion;
datos=bestpos;
varargout(1)=[1:te];
varargout(2)=[tr(find(~isnan(tr)))]];

clear global Op

tiempoejecucion=toc;
disp(' ')
disp('Tiempo de ejecución')
disp(tiempoejecucion)

return

```

ANEXO C: CÓDIGO DE PROGRAMACIÓN FUNCIÓN OBJETIVO FICHERO 'OBJETIVO_INDIVIDUO.M'

```

% ANÁLISIS DEL DESEMPEÑO DE DOS MODELOS META-HEURÍSTICOS EN EL PROBLEMA DE
% FORMACIÓN DE CELDAS DE MANUFACTURA

%ALGORITMO PSO

%AUTOR: JOHANNA RODRÍGUEZ LEÓN

%Fichero OBJETIVO_INDIVIDUO.M

function [out,AsignacionMaq_Celda,Restriccion]=OBJETIVO_INDIVIDUO(in)

global Dp Op Top Mjp CMm CAPm CTI TMax TMin C

load DpAG.mat
load SecOpAG.mat
load TopAG.mat
load MjpAG.mat
load CMmAG.mat

CAPm(1:10)=480;
CTI=1;
TMax=10;
TMin=2;
C=4;

Matriz=in;
P=max(size(Matriz));
out=zeros(P,1);
AsignacionMaq_Celda=cell(P,1);
MatrizAsignacion=cell(P,1);
Restriccion=NaN(P,1);

for i=1:P
    MatrizAsignacion(i)=Matriz(i);
    Nec_Maquina=NECESIDADES_MAQUINA(MatrizAsignacion,Mjp,Top,CAPm,C,Dp);
    Costo_Maquina=COSTO_MAQUINA(Nec_Maquina,CMm);
    Costo_Intercelulares=TRANSFERENCIAS_INTERCELULARES(MatrizAsignacion,Op,Dp,CTI);
    out(i,1)=Costo_Maquina+Costo_Intercelulares;
    Nummaquinas=sum(ceil(Nec_Maquina));
    AsignacionMaq_Celda(i)=ceil(Nec_Maquina);
    Satisfaccion=~(any(Nummaquinas<TMin)|any(Nummaquinas>TMax));
    Restriccion(i,1)=Satisfaccion;
    if Satisfaccion==0
        NumeroMaquinasSobrantes=sum((Nummaquinas-TMax).*(Nummaquinas>TMax));
        NumeroMaquinasFaltantes=sum((TMin-Nummaquinas).*(Nummaquinas<TMin));
        NumeroMaquinas=NumeroMaquinasSobrantes+NumeroMaquinasFaltantes;
        out(i,1)=out(i,1)*(1+(NumeroMaquinas*0.1));
    end
end

clear global Dp Op Top Mjp CMm CAPm CTI TMax TMin C out AsignacionMaq_Celda

***** FUNCIONES UTILIZADAS *****

```

```

***** NECESIDADES DE MÁQUINA *****
function Nec_Maquina=NECESIDADES_MAQUINA(MatrizAsignacion,Mjp,Top,CAPm,C,Dp)
tiempostotales=Top.*repmat(Dp,[size(MatrizAsignacion(i),1) 1]);
for c=1:C
    Mascaracelda=(MatrizAsignacion(i)==c);
    for m=1:max(max(Mjp))
        Mascaramaquina=(Mjp==m);
        Nec_Maquina(m,c)=(sum(sum(tiempostotales.*Mascaracelda.*Mascaramaquina)))/(CAPm(m));
    end
end
end

***** COSTO DE MÁQUINA *****
function Costo_Maquina=COSTO_MAQUINA(Nec_Maquina,CMm)
Costo_Maquina=sum(sum(ceil(Nec_Maquina).*(repmat(CMm',[1 size(Nec_Maquina,2)]))));
end

***** COSTOS INTERCELULARES *****
function Costo_Intercelulares=TRANSFERENCIAS_INTERCELULARES(MatrizAsignacion,Op,Dp,CTI)
contador=0;
for p=1:size(MatrizAsignacion(i),2)
    for j=1:(Op(p)-1)
        if(MatrizAsignacion(i)(j,p)~=MatrizAsignacion(i)(j+1,p))
            contador=contador+Dp(p);
        end
    end
end
Costo_Intercelulares=contador*CTI;
end

**** Fin función principal ****
end

```

ANEXO D: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO C1

FACTOR: COEFICIENTE C1							
NIVEL	RÉPLICA	COSTO	TIEMPO	COSTO		TIEMPO	
				PROM	DESV EST	PROM	DESV EST
A = 2	Replica1	22386	440,7272	22435,6000	43,8668	447,4641	5,7581
	Replica2	22394	443,3401				
	Replica3	22457	450,1762				
	Replica4	22453	447,6866				
	Replica5	22488	455,3903				
B = 4	Replica1	22473	439,9468	22356,0000	80,4177	447,8165	7,1340
	Replica2	22320	440,3006				
	Replica3	22253	451,7221				
	Replica4	22371	454,9717				
	Replica5	22363	452,1411				
C = 6	Replica1	22564	444,6226	22555,2000	139,7773	450,1105	12,1432
	Replica2	22615	438,5665				
	Replica3	22393	468,8754				
	Replica4	22454	455,2784				
	Replica5	22750	443,2096				
D = 8	Replica1	22423	442,6043	22540,8000	94,8483	446,1851	4,2382
	Replica2	22571	443,0906				
	Replica3	22668	452,3417				
	Replica4	22567	448,8332				
	Replica5	22475	444,0556				

ANEXO E: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO C2

FACTOR: COEFICIENTE C2							
NIVEL	RÉPLICA	COSTO	TIEMPO	COSTO		TIEMPO	
				PROM	DESV EST	PROM	DESV EST
A = 2	Replica1	22407	431,0375	22433,8000	77,5609	439,2259	9,2440
	Replica2	22442	428,8448				
	Replica3	22319	446,9092				
	Replica4	22476	449,6036				
	Replica5	22525	439,7344				
B = 4	Replica1	22519	438,3055	22495,4000	34,0044	442,8490	5,8074
	Replica2	22542	440,1809				
	Replica3	22459	439,5263				
	Replica4	22479	452,6226				
	Replica5	22478	443,6095				
C = 6	Replica1	22400	435,7651	22528,4000	83,8409	441,1553	10,0794
	Replica2	22518	433,8143				
	Replica3	22517	457,4080				
	Replica4	22611	444,5463				
	Replica5	22596	434,243				
D = 8	Replica1	22611	449,8035	22557,0000	55,9687	443,7632	7,2370
	Replica2	22592	441,2307				
	Replica3	22585	451,3908				
	Replica4	22479	433,3779				
	Replica5	22518	443,0133				

ANEXO F: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO W

FACTOR: PESO INERCIAL W							
NIVEL	RÉPLICA	COSTO	TIEMPO	COSTO		TIEMPO	
				PROM	DESV EST	PROM	DESV EST
A = 1.0	Replica1	22732	429,4682	22484,4000	148,3924	433,1537	3,5087
	Replica2	22428	437,5634				
	Replica3	22423	435,1679				
	Replica4	22344	429,7340				
	Replica5	22495	433,8352				
B = 0.9	Replica1	22483	439,6029	22418,6000	65,3322	433,6244	4,9194
	Replica2	22383	426,1385				
	Replica3	22383	434,4253				
	Replica4	22494	435,4146				
	Replica5	22350	432,5409				
C = 0.8	Replica1	22612	427,6688	22486,6000	109,3837	430,4741	2,6477
	Replica2	22370	427,4929				
	Replica3	22384	432,3471				
	Replica4	22491	432,6743				
	Replica5	22576	432,1872				
D = 0.7	Replica1	22313	435,1225	22487,6000	121,7284	434,0046	0,9392
	Replica2	22589	433,9504				
	Replica3	22497	434,7552				
	Replica4	22429	432,9155				
	Replica5	22610	433,2793				

ANEXO G: RESULTADOS POR RÉPLICA AJUSTE DEL PARÁMETRO TAMAÑO DE POBLACIÓN

FACTOR: TAMAÑO DE POBLACIÓN							
NIVEL	RÉPLICA	COSTO	TIEMPO	COSTO		TIEMPO	
				PROM	DESV EST	PROM	DESV EST
A = 300	Replica1	22354	438,3259	22356,4000	38,7337	442,4113	6,0327
	Replica2	22305	439,3454				
	Replica3	22410	451,5478				
	Replica4	22341	437,2933				
	Replica5	22372	445,5443				
B = 500	Replica1	22452	727,2586	22377,8000	51,6159	738,1641	13,9711
	Replica2	22341	737,5152				
	Replica3	22410	725,4277				
	Replica4	22330	760,4135				
	Replica5	22356	740,2053				
C = 800	Replica1	22364	1170,8000	22392,4000	40,2778	1179,4600	14,3146
	Replica2	22371	1166,8000				
	Replica3	22359	1172,2000				
	Replica4	22452	1201,6000				
	Replica5	22416	1185,9000				
D = 1200	Replica1	22251	1751,7000	22222,8000	21,3120	1779,5000	17,1326
	Replica2	22209	1776,9000				
	Replica3	22219	1782,3000				
	Replica4	22198	1795,0000				
	Replica5	22237	1791,6000				

ANEXO H: RESULTADOS POR RÉPLICA AJUSTE DEL FACTOR TAMAÑO DE POBLACIÓN CON ITERACIONES

FACTOR: TAMAÑO DE POBLACIÓN CON NÚMERO DE ITERACIONES							
NIVEL	RÉPLICA	COSTO	TIEMPO	COSTO		TIEMPO	
				PROM	DESV EST	PROM	DESV EST
A	Replica1	22354	438,3259	22356,4000	38,7337	442,4113	6,0327
	Replica2	22305	439,3454				
	Replica3	22410	451,5478				
	Replica4	22341	437,2933				
	Replica5	22372	445,5443				
B	Replica1	22355	541,4981	22353,4000	29,9383	551,0902	9,2439
	Replica2	22344	559,2833				
	Replica3	22335	561,5439				
	Replica4	22404	542,5384				
	Replica5	22329	550,5874				
C	Replica1	22251	1751,7000	22222,8000	21,3120	1779,5000	17,1326
	Replica2	22209	1776,9000				
	Replica3	22219	1782,3000				
	Replica4	22198	1795,0000				
	Replica5	22237	1791,6000				
D	Replica1	22332	1215,0000	22332,2000	20,4377	1217,5800	6,9139
	Replica2	22315	1223,8000				
	Replica3	22321	1210,0000				
	Replica4	22326	1213,2000				
	Replica5	22367	1225,9000				

ANEXO I: DATOS NECESARIOS PARA VERIFICAR SUPUESTOS DE LOS DISEÑOS DE EXPERIMENTOS APLICADOS EN EL AJUSTE DE PARÁMETROS

✓ Factor C1

Residuos*	Orden Corrida	Dato ri**	Rango i	$(i-0,5)/N$	$Z_i = \Phi^{-1}((i-0,5)/N)$ ***
-49,6	3	-162,2	1	0,025	-1,96
-41,6	8	-117,8	2	0,075	-1,44
21,4	11	-103	3	0,125	-1,15
17,4	16	-101,2	4	0,175	-0,93
52,4	19	-65,8	5	0,225	-0,76
117	2	-49,6	6	0,275	-0,60
-36	6	-41,6	7	0,325	-0,45
-103	12	-36	8	0,375	-0,32
15	14	7	9	0,425	-0,19
7	17	8,8	10	0,475	-0,06
8,8	4	15	11	0,525	0,06
59,8	5	17,4	12	0,575	0,19
-162,2	9	21,4	13	0,625	0,32
-101,2	15	26,2	14	0,675	0,45
194,8	20	30,2	15	0,725	0,60
-117,8	1	52,4	16	0,775	0,76
30,2	7	59,8	17	0,825	0,93
127,2	10	117	18	0,875	1,15
26,2	13	127,2	19	0,925	1,44
-65,8	18	194,8	20	0,975	1,96

*Residuos: residuos de las 20 observaciones según orden presentado en las tablas de los anexos D, E, F, G, H

**Dato ri: Residuos de las observaciones ordenados de menor a mayor

***Zi: Distribución normal estándar acumulada

✓ Factor C2

Residuos	Orden Corrida	Dato ri	Rango i	$(i-0,5)/N$	$Z_i = \Phi^{-1}((i-0,5)/N)$
-26,8	2	-128,4	1	0,025	-1,96
8,2	7	-114,8	2	0,075	-1,44
-114,8	9	-78	3	0,125	-1,15
42,2	14	-39	4	0,175	-0,93
91,2	18	-36,4	5	0,225	-0,76
23,6	3	-26,8	6	0,275	-0,60
46,6	5	-17,4	7	0,325	-0,45
-36,4	12	-16,4	8	0,375	-0,32
-16,4	15	-11,4	9	0,425	-0,19
-17,4	19	-10,4	10	0,475	-0,06
-128,4	4	8,2	11	0,525	0,06
-10,4	6	23,6	12	0,575	0,19
-11,4	10	28	13	0,625	0,32
82,6	16	35	14	0,675	0,45
67,6	17	42,2	15	0,725	0,60
54	1	46,6	16	0,775	0,76
35	8	54	17	0,825	0,93
28	11	67,6	18	0,875	1,15
-78	13	82,6	19	0,925	1,44
-39	20	91,2	20	0,975	1,96

✓ Factor W

Residuos	Orden Corrida	Dato ri	Rango i	$(i-0,5)/N$	$Z_i = \Phi^{-1}((i-0,5)/N)$
247,6	1	-174,6	1	0,025	-1,96
-56,4	8	-140,4	2	0,075	-1,44
-61,4	9	-116,6	3	0,125	-1,15
-140,4	13	-102,6	4	0,175	-0,93
10,6	20	-68,6	5	0,225	-0,76
64,4	4	-61,4	6	0,275	-0,60
-35,6	5	-58,6	7	0,325	-0,45
-35,6	12	-56,4	8	0,375	-0,32
75,4	15	-35,6	9	0,425	-0,19
-68,6	17	-35,6	10	0,475	-0,06
125,4	3	4,4	11	0,525	0,06
-116,6	6	9,4	12	0,575	0,19
-102,6	11	10,6	13	0,625	0,32
4,4	14	64,4	14	0,675	0,45
89,4	19	75,4	15	0,725	0,60
-174,6	2	89,4	16	0,775	0,76
101,4	7	101,4	17	0,825	0,93
9,4	10	122,4	18	0,875	1,15
-58,6	16	125,4	19	0,925	1,44
122,4	18	247,6	20	0,975	1,96

✓ Factor tamaño de población

Residuos	Orden Corrida	Dato r_i	Rango i	$(i-0,5)/N$	$Z_i = \Phi^{-1}((i-0,5)/N)$
-2,4	3	-51,4	1	0,025	-1,96
-51,4	5	-47,8	2	0,075	-1,44
53,6	9	-36,8	3	0,125	-1,15
-15,4	13	-33,4	4	0,175	-0,93
15,6	18	-28,4	5	0,225	-0,76
74,2	4	-24,8	6	0,275	-0,60
-36,8	6	-21,8	7	0,325	-0,45
32,2	12	-21,4	8	0,375	-0,32
-47,8	15	-15,4	9	0,425	-0,19
-21,8	20	-13,8	10	0,475	-0,06
-28,4	2	-3,8	11	0,525	0,06
-21,4	7	-2,4	12	0,575	0,19
-33,4	10	14,2	13	0,625	0,32
59,6	16	15,6	14	0,675	0,45
23,6	19	23,6	15	0,725	0,60
28,2	1	28,2	16	0,775	0,76
-13,8	8	32,2	17	0,825	0,93
-3,8	11	53,6	18	0,875	1,15
-24,8	14	59,6	19	0,925	1,44
14,2	17	74,2	20	0,975	1,96

✓ Factor tamaño de población con iteraciones

Residuos	Orden Corrida	Dato r_i	Rango i	$(i-0,5)/N$	$Z_i = \Phi^{-1}((i-0,5)/N)$
-2,4	2	-51,4	1	0,025	-1,96
-51,4	5	-24,8	2	0,075	-1,44
53,6	11	-24,4	3	0,125	-1,15
-15,4	14	-18,4	4	0,175	-0,93
15,6	18	-17,2	5	0,225	-0,76
1,6	4	-15,4	6	0,275	-0,60
-9,4	8	-13,8	7	0,325	-0,45
-18,4	10	-11,2	8	0,375	-0,32
50,6	16	-9,4	9	0,425	-0,19
-24,4	20	-6,2	10	0,475	-0,06
28,2	1	-3,8	11	0,525	0,06
-13,8	7	-2,4	12	0,575	0,19
-3,8	12	-0,2	13	0,625	0,32
-24,8	13	1,6	14	0,675	0,45
14,2	19	14,2	15	0,725	0,60
-0,2	3	15,6	16	0,775	0,76
-17,2	6	28,2	17	0,825	0,93
-11,2	9	34,8	18	0,875	1,15
-6,2	15	50,6	19	0,925	1,44
34,8	17	53,6	20	0,975	1,96

ANEXO J: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR C2

✓ Normalidad

En la gráfica de probabilidad normal de los residuos de las 20 observaciones realizadas se ve que los puntos en ella tienden a estar alineados en una línea recta. Se concluye entonces que los residuos e_{ij} siguen una distribución normal con media cero.

Gráfica de probabilidad normal de residuos Factor C2



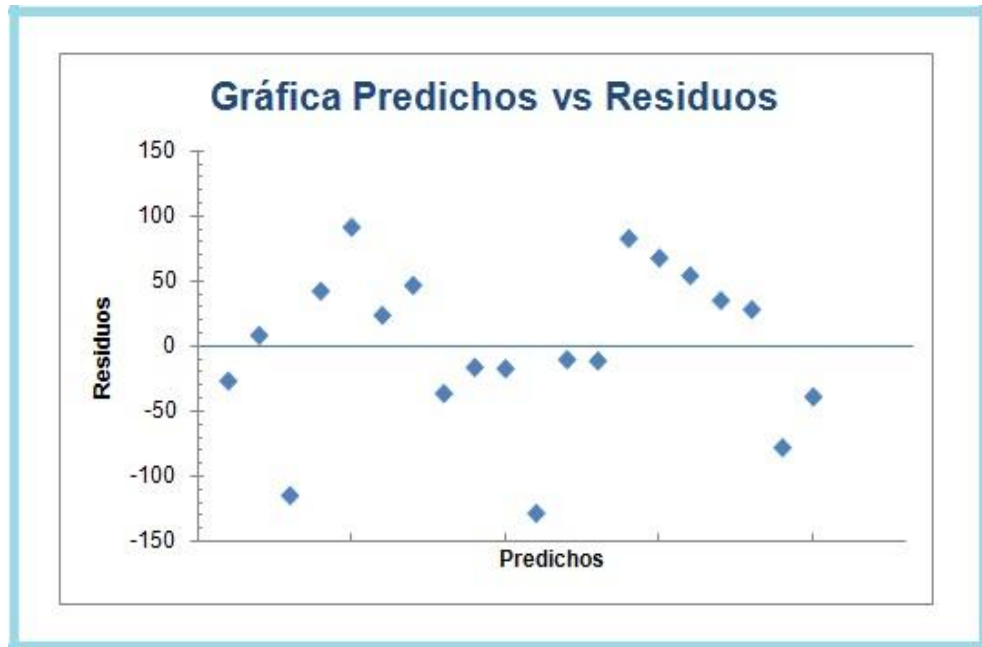
Fuente: Autor

✓ Varianza constante

La siguiente es la gráfica de valores predichos contra residuos. Los puntos en la gráfica se distribuyen de manera aleatoria sobre una banda horizontal, luego, se

concluye que se cumple el supuesto de que los tratamientos aplicados tienen igual varianza σ^2 .

Gráfica de dispersión de valores predichos contra residuos del factor C2

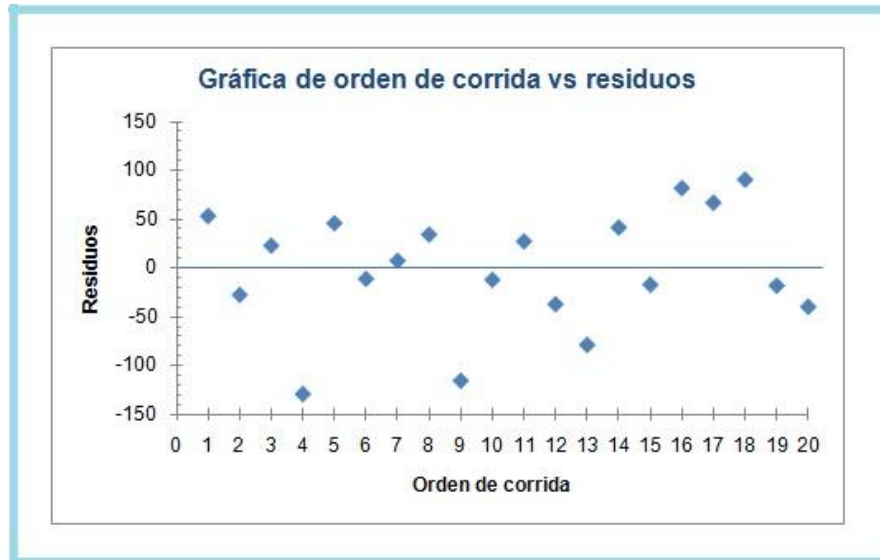


Fuente: Autor

✓ Independencia

Para comprobar este supuesto se grafica el orden en que se obtuvo un dato contra el residuo correspondiente. Como no se detecta una tendencia no aleatoria claramente definida se concluye que el supuesto de que los e_{ij} son independientes entre sí se cumple a cabalidad en el diseño aplicado.

Gráfica de tiempo vs residuos para el factor C2



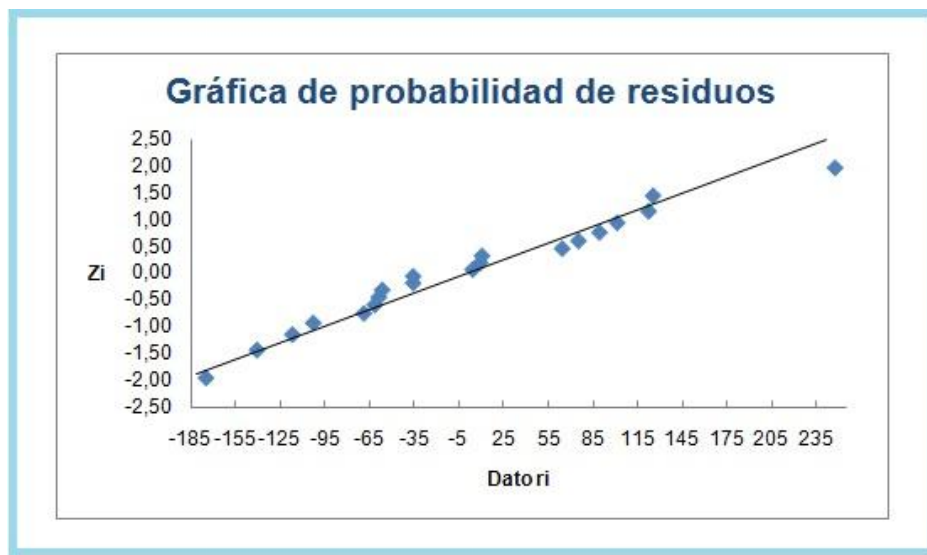
Fuente: Autor

ANEXO K: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR W

✓ Normalidad

En la figura a continuación se muestra la gráfica de probabilidad normal de los residuos de las 20 observaciones. Los puntos de la gráfica tienden a estar alineados en una línea recta; se concluye que los residuos e_{ij} siguen una distribución normal con media cero.

Gráfica de probabilidad normal de residuos Factor peso W

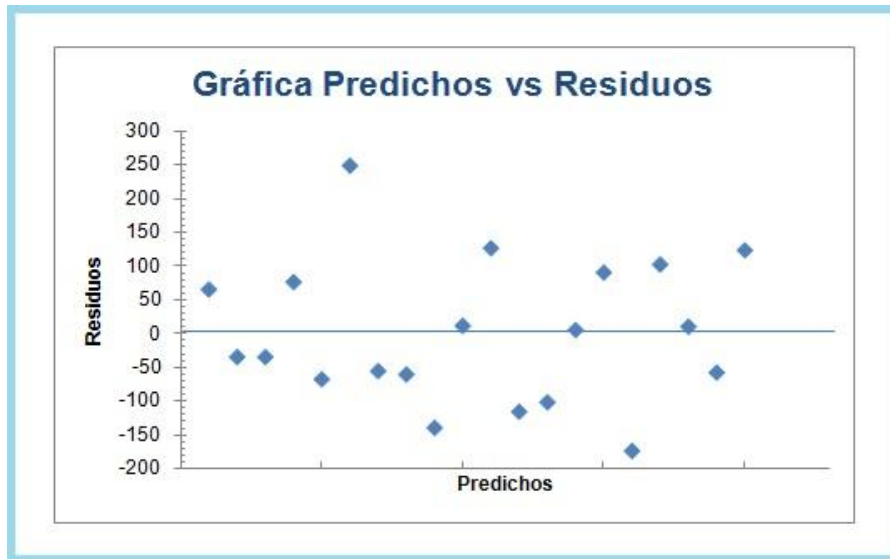


Fuente: Autor

✓ Varianza constante

De la gráfica de valores predichos contra residuos se observa que los puntos se distribuyen de manera aleatoria en una banda horizontal. Se concluye que se cumple el supuesto de que los tratamientos aplicados tienen igual varianza σ^2 .

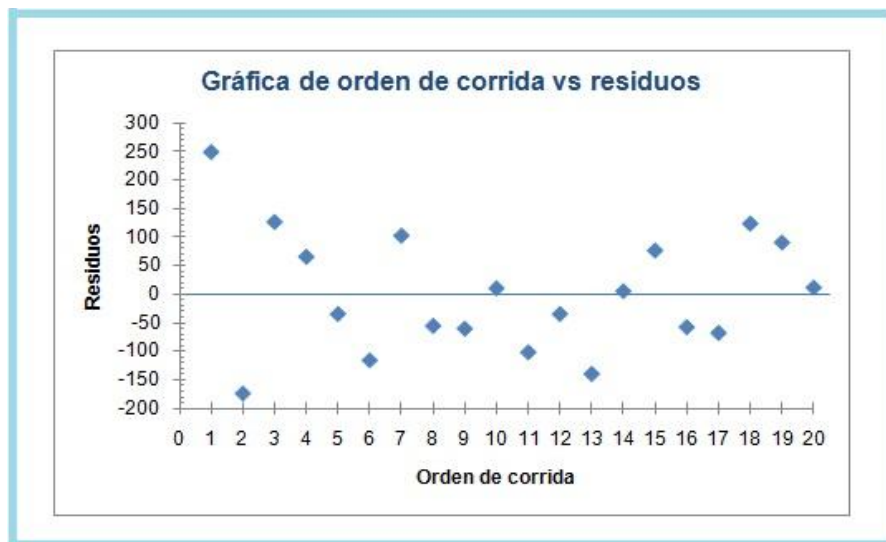
Gráfica de dispersión de valores predichos contra residuos del factor W



Fuente: Autor

✓ Independencia

Gráfica de tiempo vs residuos para el factor peso W



Fuente: Autor

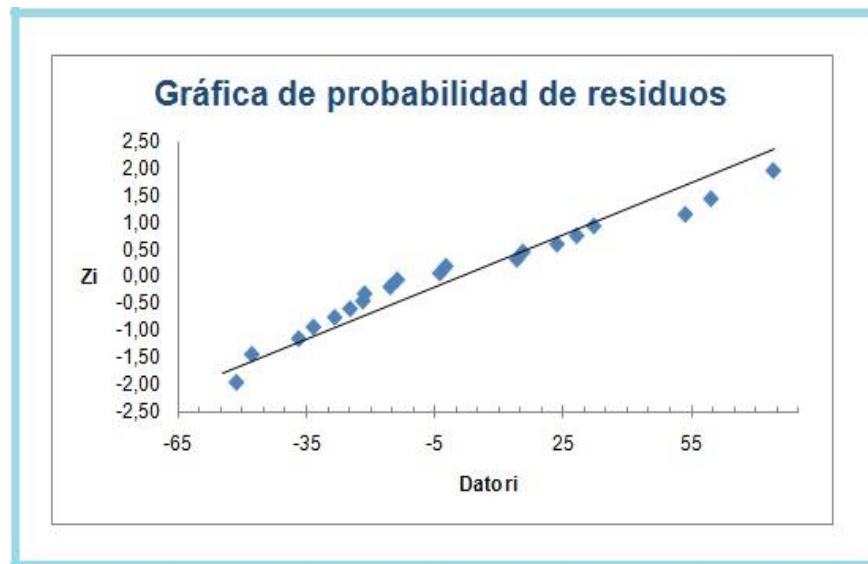
De la gráfica de orden de corrida contra residuos no se detecta una tendencia no aleatoria claramente definida por lo que se concluye que se cumple con el supuesto de que los e_{ij} son independientes entre sí.

ANEXO L: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR TAMAÑO DE POBLACIÓN

✓ Normalidad

De la siguiente gráfica de probabilidad normal se concluye que los residuos de las observaciones verifican el supuesto de normalidad con media cero, pues tienden a quedar alineados en una línea recta.

Gráfica de probabilidad normal de residuos factor C1

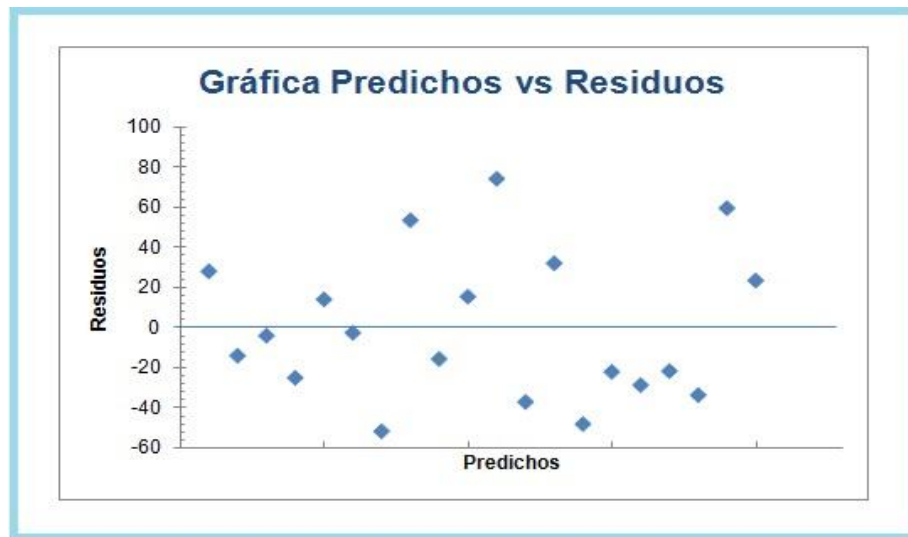


Fuente: Autor

✓ Varianza constante

En la gráfica de valores predichos contra residuos de las observaciones tomadas se ve que los puntos se distribuyen aleatoriamente sobre una banda horizontal, lo cual significa que se cumple el supuesto de que los tratamientos aplicados tienen igual varianza σ^2 .

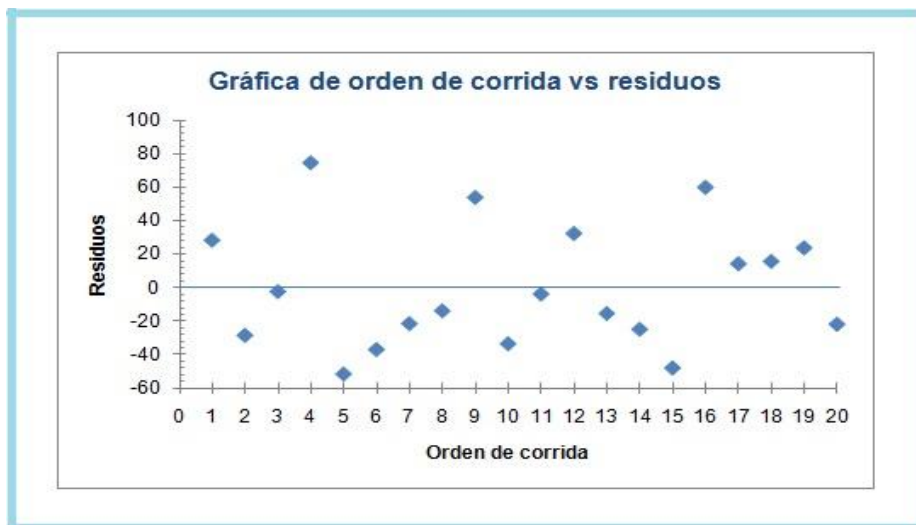
Gráfica de dispersión de valores predichos contra residuos factor Tamaño de población



Fuente: Autor

✓ Independencia

Gráfica de tiempo vs residuos para el factor Tamaño de población



Fuente: Autor

De la gráfica correspondiente a la verificación de este supuesto, como no se detecta una tendencia no aleatoria claramente definida se concluye que se cumple con el supuesto de que los residuos de las observaciones son independientes entre sí.

ANEXO M: VERIFICACIÓN DE SUPUESTOS MODELO DE AJUSTE FACTOR TAMAÑO DE POBLACIÓN CON ITERACIONES

✓ Normalidad

A continuación se muestra el gráfico de probabilidad de residuos correspondiente. Aunque existen un par de puntos aberrantes, el análisis de varianza resiste pequeñas y moderadas desviaciones al supuesto de normalidad. Por lo tanto para este caso se concluye que el diseño aplicado cumple con el supuesto de que los residuos e_{ij} siguen una distribución normal con media cero.

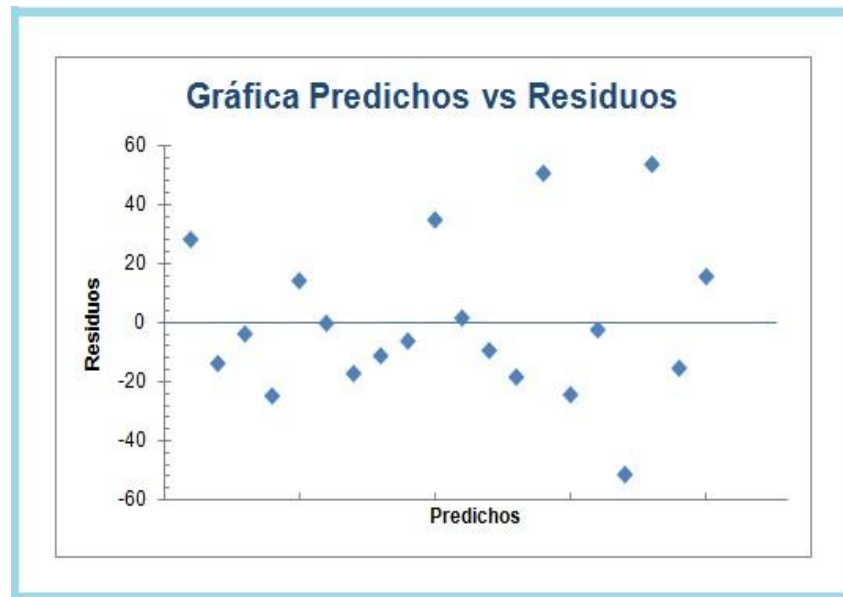
Gráfica de probabilidad normal de residuos para Tamaño de población con iteraciones



Fuente: Autor

✓ Varianza constante

Gráfica de dispersión de valores predichos contra residuos Tamaño de población con iteraciones



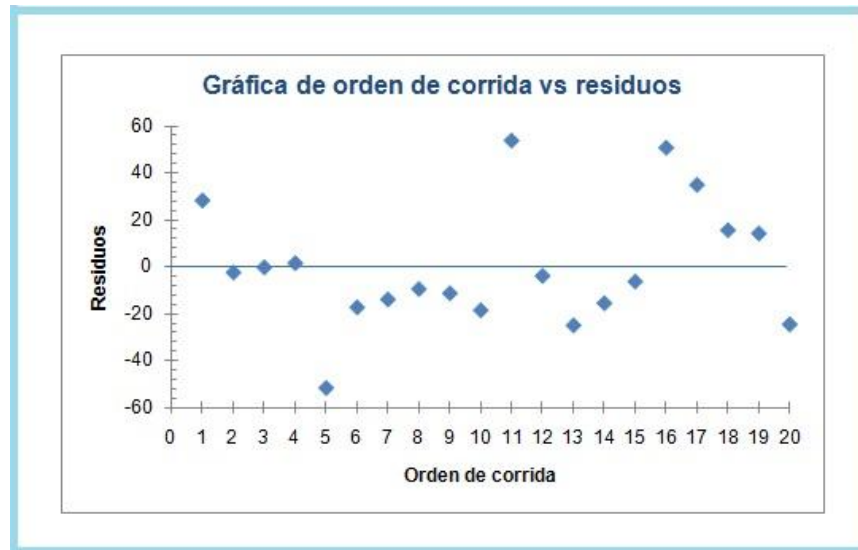
Fuente: Autor

En la gráfica de valores predichos contra residuos se observa que los puntos se distribuyen de manera aleatoria en una banda horizontal, sin ninguna tendencia o patrón claramente definido. Se concluye que se cumple el supuesto de que los tratamientos aplicados tienen igual varianza σ^2 .

✓ Independencia

Entre tanto en la siguiente figura se muestra la gráfica de orden de corrida contra residuos de las observaciones tomadas. Como no se detecta una tendencia no aleatoria claramente definida se concluye que se cumple con el supuesto de que los e_{ij} son independientes entre sí.

Gráfica de tiempo vs residuos Tamaño de población con iteraciones



Fuente: Autor

ANEXO N: CONJUNTO DE DATOS PROBLEMA DE AJUSTE EXPERIMENTAL DEL ALGORITMO PSO

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	30
Número de tipos de máquina	10
Número de celdas	4
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	10

- ✓ Demanda por producto (Dp)

Producto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Demanda	24	23	21	23	17	16	19	24	22	25	24	12	14	18	16
Producto	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demanda	10	15	17	15	11	10	21	22	12	17	22	20	13	22	25

- ✓ Asignación de tipo de máquina por operación por producto (Mjp)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	7	4	5	7	5	8	7	9	10	7	4	5	10	9	10
2	3	6	10	2	4	10	5	7	6	5	5	1	3	2	3
3	6	8	8	10			3		7	1	4	10	4	1	
4		4	7	1						9	9	2	3		
5		3								5	8				
6		9								8	2				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	10	8	4	4	4	4	4	3	10	10	1	1	5	6	2
2	7	4	9	1	1	7	10	7	6	9	3	6	1	9	5
3	8		6	9		3	8	10	2	5	9		3		8
4	6			5			10		5				5		2
5							2		3						
6							10		1						

- ✓ Tiempos de producción por operación por producto (Top)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	5	9	2	7	4	2	5	10	7	4	7	5	6	6	4
2	3	5	5	8	4	2	7	9	8	5	2	6	5	10	9
3	9	5	10	10			3		9	5	2	10	9	2	
4		3	4	7						2	3	8	10		
5		4								9	2				
6		4								4	4				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	9	10	7	9	6	2	5	7	3	4	2	8	8	8	9
2	8	7	8	3	3	5	6	6	2	5	6	10	5	3	7
3	10		6	5		7	10	6	9	10	9		5		6
4	8			2			7		9				5		4
5							8		4						
6							10		5						

- ✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máquina	Costo Unitario	Capacidad
1	286	480
2	468	480
3	477	480
4	944	480
5	1225	480
6	1590	480
7	457	480
8	340	480
9	441	480
10	855	480

ANEXO O: CONJUNTO DE DATOS PROBLEMA 1

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	30
Número de tipos de máquina	10
Número de celdas	4
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	10

- ✓ Demanda por producto (Dp)

Producto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Demanda	23	24	12	24	20	11	14	18	25	25	12	25	25	17	22
Producto	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demanda	12	16	24	22	25	20	10	23	24	20	22	21	16	20	12

- ✓ Asignación de tipo de máquina por operación por producto (Mjp)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	3	6	5	8	8	6	5	9	3	4	10	5	1	10
2	5	6	6	1	10	10	5	7	4	8	5	9	3	4	6
3		1	7		6	4	8	2		3		4	10	8	
4			4		3	8		5		3				7	
5														2	
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	9	8	1	8	4	8	7	9	7	4	2	2	6	2	4
2	5	2	9	3	10	9	6	2	4	1	10	3	1	7	2
3		4	3			2	1	4	8	8	5		10	6	
4			3				7	3						8	
5			5				10	6						2	

✓ Tiempos de producción por operación por producto (Top)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	7	4	5	6	1	8	7	3	8	6	10	4	2	8
2	4	8	7	6	5	5	3	4	2	1	3	9	10	10	2
3		10	4		3	1	6	4		4		1	6	2	
4			5		6	6		3		10				8	
5														8	
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	9	1	3	2	1	8	4	7	8	1	3	1	9	5	1
2	7	6	7	10	5	5	8	7	5	6	10	1	7	5	6
3		3	4			8	3	2	9	8	5		7	6	
4			9				3	2						1	
5			2				5	3						7	

✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máquina	Costo Unitario	Capacidad
1	908	480
2	868	480
3	860	480
4	312	480
5	906	480
6	1266	480
7	1978	480
8	518	480
9	773	480
10	606	480

ANEXO P: CONJUNTO DE DATOS PROBLEMA 2

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	30
Número de tipos de máquina	10
Número de celdas	4
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	10

- ✓ Demanda por producto (Dp)

Producto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Demanda	12	24	23	15	19	22	15	25	12	15	18	20	11	18	10
Producto	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Demanda	13	11	22	16	21	18	19	19	19	17	23	21	12	12	22

- ✓ Asignación de tipo de máquina por operación por producto (Mjp)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	2	8	1	1	10	7	1	2	5	2	8	10	1	5
2	3	6	7	3	10	7	1	10	1	2	9	2	3	10	8
3	2	4	6	9			5		5	7	7	8	9	10	
4		9	3	8						4	3	5	6		
5		8								5	1				
6		5								7	6				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	2	7	2	10	9	9	3	1	8	7	5	7	5	9
2	5	7	6	8	2	3	1	6	2	1	10	4	3	1	7
3	1		2	9		7	3	8	10	5	8		6		3
4	4			9			2		4				1		4
5							8		1						
6							6		1						

✓ Tiempos de producción por operación por producto (Top)

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	6	7	7	9	4	7	3	10	8	9	9	10	6	5	2
2	2	5	1	9	5	9	9	1	7	4	3	1	2	5	10
3	8	9	2	1			6		1	8	6	6	5	7	
4		5	3	5						3	1	8	4		
5		1								6	8				
6		9								1	7				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	8	5	7	10	9	5	7	7	4	2	1	4	10	6	3
2	6	9	9	9	4	7	9	1	1	5	8	9	8	6	8
3	4		8	10		10	4	2	9	10	1		10		1
4	8			10			10		7				9		6
5							4		1						
6							10		10						

✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máquina	Costo Unitario	Capacidad
1	1472	480
2	1384	480
3	1128	480
4	771	480
5	1267	480
6	888	480
7	858	480
8	776	480
9	1835	480
10	1784	480

ANEXO Q: CONJUNTO DE DATOS PROBLEMA 3

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	40
Número de tipos de máquina	10
Número de celdas	5
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	10

- ✓ Demanda por producto (Dp)

Producto	Demanda		Producto	Demanda
1	10		21	21
2	14		22	19
3	22		23	21
4	14		24	16
5	13		25	24
6	23		26	16
7	23		27	12
8	19		28	21
9	15		29	23
10	19		30	16
11	17		31	12
12	15		32	10
13	14		33	11
14	23		34	20
15	18		35	22
16	20		36	22
17	15		37	25
18	10		38	12
19	20		39	23
20	14		40	17

✓ Asignación de tipo de máquina por operación por producto (Mjp)

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	10	9	10	10	8	2	6	9	5	4	8	4	7	4	2	10	2	4	9
2	1	6	9	2	3	10	6	10	1	4	8	10	7	7	4	2	10	10	2	3
3		5	1	1	5	2	6		6	5	6		4	7		8		10	8	
4			2		3	10	3			2	10		7	3					2	
5			2			2	7				5			5					7	
6						4														
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	10	8	6	8	7	7	7	4	5	6	6	5	9	9	6	8	2	6	10
2	7	4	7	1	7	5	9	1	7	2	4	7	2	4	2	7	4	7	7	2
3		6			4	6	5	7	1	8		1	8		2			9	8	5
4		10			10	9	8	2	7	10					2			8	3	10
5		1			10	4	2	4	3	9					8				3	8
6					3		9		4										7	

✓ Tiempos de producción por operación por producto (Top)

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	4	3	10	4	8	9	1	7	8	2	3	6	8	8	7	8	6	8	6	7
2	9	1	3	1	7	10	4	3	6	8	7	1	4	10	2	2	2	4	2	3
3		3	2	3	7	1	8		1	10	10		4	10			2		5	7
4			9		8	9	8			7	7		10	8						5
5			9			4	6				7			1						6
6						4														
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	3	8	6	6	5	4	9	8	8	7	6	4	7	5	6	2	8	6	10	5
2	2	10	2	5	3	5	10	3	1	5	6	3	4	6	8	3	7	6	5	7
3		7			1	4	5	3	6	3		4	4		6			6	4	3
4		7			1	10	3	3	8	9					5			5	7	3
5		3			10	1	10	7	9	5					8				4	2
6					7		7		6										4	

- ✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máquina	Costo Unitario	Capacidad
1	1612	480
2	193	480
3	498	480
4	569	480
5	1638	480
6	1840	480
7	205	480
8	1054	480
9	1490	480
10	1724	480

ANEXO R: CONJUNTO DE DATOS PROBLEMA 4

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	40
Número de tipos de máquina	10
Número de celdas	5
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	10

- ✓ Demanda por producto (Dp)

Producto	Demanda		Producto	Demanda
1	12		21	11
2	16		22	16
3	19		23	10
4	15		24	11
5	23		25	16
6	10		26	19
7	18		27	15
8	19		28	24
9	25		29	17
10	12		30	22
11	18		31	18
12	10		32	24
13	22		33	10
14	16		34	20
15	23		35	15
16	21		36	18
17	21		37	11
18	14		38	25
19	10		39	17
20	25		40	20

✓ Asignación de tipo de máquina por operación por producto (Mjp)

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	9	6	1	10	2	10	5	6	7	8	6	6	6	2	5	8	8	1	10	3
2	9	10	2	5	8	6	6	10	9	5	9	2	9	7	1	7	1	10	3	4
3	10	7		10	8				10	8	1	9	3	8		1			10	
4				4	9				7				6	1		2			5	
5				8									1	7					9	
6																				
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	5	5	9	1	5	3	2	5	5	9	2	9	6	1	2	10	6	5	4	5
2	1	9	3	3	9	6	6	9	3	8	3	2	9	7	4	9	6	1	1	7
3	1		9		6		7	1	7				5			4	7		8	8
4	8								9										6	
5	3																		4	
6																			4	

✓ Tiempos de producción por operación por producto (Top)

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	6	7	8	10	8	7	5	5	4	4	9	7	6	7	2	7	6	8	7	8
2	9	7	4	5	2	9	10	10	10	8	1	2	1	6	1	4	5	7	7	10
3	3	3		9	10				7	4	3	2	8	7		3			6	
4				2	6				6				5	1		4			9	
5				5									8	6					7	
6																				
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	2	2	2	5	1	3	4	2	6	7	3	6	3	9	6	1	1	3	5	8
2	6	4	10	7	9	10	6	9	7	10	1	10	6	6	4	5	6	7	10	9
3	5		10		5		3	3	7				2			8	10		2	10
4	1								1										6	
5	2																		4	
6																			5	

- ✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máquina	Costo Unitario	Capacidad
1	671	480
2	1269	480
3	761	480
4	1428	480
5	1462	480
6	1969	480
7	537	480
8	264	480
9	1950	480
10	104	480

ANEXO S: CONJUNTO DE DATOS PROBLEMA 5

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	50
Número de tipos de máquina	20
Número de celdas	6
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	20

- ✓ Demanda por producto (Dp)

Producto	Demanda	Producto	Demanda	Producto	Demanda
1	14	18	16	35	25
2	18	19	23	36	25
3	16	20	23	37	24
4	13	21	10	38	24
5	19	22	16	39	15
6	18	23	13	40	15
7	19	24	20	41	24
8	23	25	20	42	22
9	24	26	22	43	16
10	23	27	20	44	18
11	21	28	14	45	23
12	15	29	24	46	18
13	12	30	10	47	23
14	11	31	16	48	19
15	16	32	17	49	18
16	19	33	14	50	15
17	14	34	15		

✓ Asignación de tipo de máquina por operación por producto (Mjp)

Op	PRODUCTO																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25					
1	19	9	12	3	12	10	6	4	7	6	14	3	6	6	15	3	3	15	5	4	3	5	2	12	12					
2	13	3	8	18	16	4	11	7	18	9	7	1	7	5	5	1	17	2	5	6	8	14	13	16	14					
3	14		8	10	9	20	7	6		13	1	8	9	2		2	5			1	20	16		15	2					
4				17			10	9			2			18		10				3	8	5								
5											17			20								17								
6																						11								
Op	PRODUCTO																													
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50					
1	2	4	12	10	12	3	7	1	7	12	16	4	17	7	2	12	5	11	11	19	2	1	13	2	10					
2	1	6	4	14	11	15	17	5	19	6	9	7	5	5	8	3	6	10	17	5	1	2	6	9	15					
3		9			5	1	8	12		9	10	10			1	1	9			5	16	3		5	18					
4		1			4	10				4	6					10	12			6				1						
5		9				4					20					9	1			9				20						
6		4				10					2					4	2			4										

✓ Tiempos de producción por operación por producto (Top)

Op	PRODUCTO																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25					
1	9	6	9	2	10	4	4	8	9	10	4	6	1	9	2	9	9	10	4	1	6	6	5	1	10					
2	4	3	5	1	6	2	8	10	8	4	6	8	10	6	4	8	2	6	4	1	1	3	10	3	6					
3	1		3	6	8	7	2	2		7	4	7	2	6		6	7			3	6	7		7	8					
4				4			1	4			7			4		7				9	6	10								
5											10			7								3								
6																						3								
Op	PRODUCTO																													
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50					
1	4	1	1	3	5	3	8	1	3	4	8	8	5	3	4	7	2	7	6	1	9	9	10	8	2					
2	7	7	8	2	3	1	2	1	10	7	6	6	10	10	2	1	7	9	4	7	7	1	8	10	5					
3		8			7	8	1	8		10	2	4			1	7	9			3	6	5		5	9					
4		4			9	10				2	10					8	4			4				7						
5		1				8					6					10	6			7				1						
6		8				8					3					3	10			4										

- ✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máq	Costo Unitario	Capacidad		Tipo de Máq	Costo Unitario	Capacidad
1	1194	480		11	1987	480
2	1272	480		12	768	480
3	1713	480		13	1217	480
4	909	480		14	1388	480
5	628	480		15	1083	480
6	1306	480		16	155	480
7	996	480		17	1359	480
8	1947	480		18	1234	480
9	155	480		19	1953	480
10	347	480		20	1077	480

ANEXO T: CONJUNTO DE DATOS PROBLEMA 6

- ✓ Configuración característica del problema

Atributo	Valor
Número de productos	50
Número de tipos de máquina	20
Número de celdas	6
Costo Transferencias Intercelulares	1
Tamaño mínimo de celda	2
Tamaño máximo de celda	20

- ✓ Demanda por producto (Dp)

Producto	Demanda	Producto	Demanda	Producto	Demanda
1	25	18	22	35	25
2	18	19	22	36	22
3	25	20	21	37	18
4	13	21	10	38	25
5	20	22	20	39	12
6	25	23	11	40	21
7	11	24	14	41	13
8	18	25	15	42	11
9	23	26	24	43	14
10	14	27	15	44	10
11	10	28	23	45	24
12	11	29	19	46	23
13	10	30	10	47	22
14	25	31	25	48	24
15	22	32	16	49	10
16	14	33	12	50	15
17	16	34	22		

✓ Asignación de tipo de máquina por operación por producto (Mjp)

Op	PRODUCTO																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	14	20	4	7	16	15	9	16	13	18	10	8	14	8	18	12	15	9	10	11	13	14	1	11	5
2	5	5	6	16	9	2	3	19	20	2	14	10	6	3	10	20	2	20	18	11	10	17	18	17	4
3		8	1	8	12	2	13			10	12	13		14		5		5	3	19	11		14	11	12
4				16		2	20			2	19	5		4									4	19	12
5				8		3				3	5			2										19	2
6				3		16						8													16
Op	PRODUCTO																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	14	20	11	1	2	2	6	17	16	7	5	18	8	4	15	5	17	1	3	5	5	16	20	11	1
2	13	4	2	3	13	11	12	6	6	18	12	7	16	12	6	10	10	15	8	14	20	20	8	1	14
3	16	6	18	18		5	19	16	8	5	5	9	13	16		14		19	16	19	16	11	3	9	2
4						11	11	15		15		11		13		13			12		18	9	13		15
5						6		3		9				4					18			1			10
6						5		7														12			5

✓ Tiempos de producción por operación por producto (Top)

Op	PRODUCTO																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	8	3	3	9	9	10	9	6	1	2	10	6	7	1	10	2	7	3	3	8	7	3	5	5
2	9	7	6	5	4	9	3	3	6	10	1	8	9	6	5	7	1	9	8	4	8	5	8	6	3
3		6	8	8	5	2	4			2	1	9		7		1		9	2	9	9		1	2	3
4				1		3	5			9	2	8		9									7	6	6
5				5		10				4	3			8									9	4	
6				8		4						7											5		
Op	PRODUCTO																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	3	5	3	6	2	1	6	9	4	9	6	3	3	7	1	10	6	4	3	7	6	2	2	6	3
2	5	7	8	4	3	7	8	10	8	5	7	8	10	1	6	5	1	2	8	8	6	1	8	2	7
3	10	10	10	1		8	4	1	5	7	3	4	1	9		8		7	5	2	5	6	2	1	4
4						10	9	3		3		3		8		3			5		8	4	4		7
5						9		5		6				10					7			6			9
6						3		2														7			6

- ✓ Costo unitario y capacidad por tipo de máquina (CMm y CAPm)

Tipo de Máq	Costo Unitario	Capacidad		Tipo de Máq	Costo Unitario	Capacidad
1	149	480		11	259	480
2	896	480		12	854	480
3	1454	480		13	1585	480
4	1577	480		14	175	480
5	956	480		15	448	480
6	1560	480		16	1028	480
7	1824	480		17	712	480
8	491	480		18	1398	480
9	1743	480		19	333	480
10	947	480		20	1894	480

ANEXO U: RESULTADOS COMPLETOS EXPERIENCIA COMPUTACIONAL

✓ Resultados de costos

Problema	Réplica	PSO	A.G.	Prom PSO	Desv PSO	Prom A.G.	Desv A.G.
1	1	21599	21677	21612,6	39,253025	21693,4	166,73722
	2	21585	21544				
	3	21621	21586				
	4	21677	21971				
	5	21581	21689				
2	1	33732	33794	33761,8	24,004166	33757,8	35,646879
	2	33740	33699				
	3	33782	33757				
	4	33773	33765				
	5	33782	33774				
3	1	36252	35980	36303,2	161,00838	36000,2	75,087948
	2	36115	35948				
	3	36558	35917				
	4	36313	36064				
	5	36278	36092				
4	1	31590	31502	31490,4	92,381275	31458,2	29,003448
	2	31427	31436				
	3	31392	31438				
	4	31587	31474				
	5	31456	31441				
5	1	47957	48328	47236,2	627,95796	48268,4	138,45685
	2	46889	48290				
	3	46689	48025				
	4	46765	48361				
	5	47881	48338				
6	1	48996	49236	48873,4	154,68775	49053	142,59734
	2	48901	48948				
	3	49003	49174				
	4	48624	48919				
	5	48843	48988				

✓ Resultados de tiempo de ejecución

Problema	Réplica	PSO	A.G.	Prom PSO	Desv PSO	Prom A.G.	Desv A.G.
1	1	2373,4	493,8	2034,84	392,23617	638,24	114,14448
	2	2341,2	728,1				
	3	1591,3	681,0				
	4	2242,3	747,3				
	5	1626,0	541,0				
2	1	1721,1	695,9	1748,58	60,767401	605,46	66,229661
	2	1741,3	652,6				
	3	1853,6	581,8				
	4	1698,2	555,4				
	5	1728,7	541,6				
3	1	2343,3	680,5	2303,98	89,90332	734,94	61,624938
	2	2186,7	720,7				
	3	2333,8	679,3				
	4	2414,6	772,6				
	5	2241,5	821,6				
4	1	2445,7	927,7	2187,48	148,4531	867,24	72,599573
	2	2080,5	857,3				
	3	2104,3	808,9				
	4	2173,4	787,8				
	5	2133,5	954,5				
5	1	6019,2	1366,3	5979,42	286,32968	1369,48	46,979112
	2	5801,3	1420,4				
	3	5781,2	1400,8				
	4	5832,5	1297,4				
	5	6462,9	1362,5				
6	1	4860,6	1011,6	4548,18	297,07003	1197,04	184,91506
	2	4207,4	1489,3				
	3	4852,6	1250,3				
	4	4345,0	1142,7				
	5	4475,3	1091,3				

ANEXO V: RESULTADOS ALGORITMO PSO PROBLEMA 2

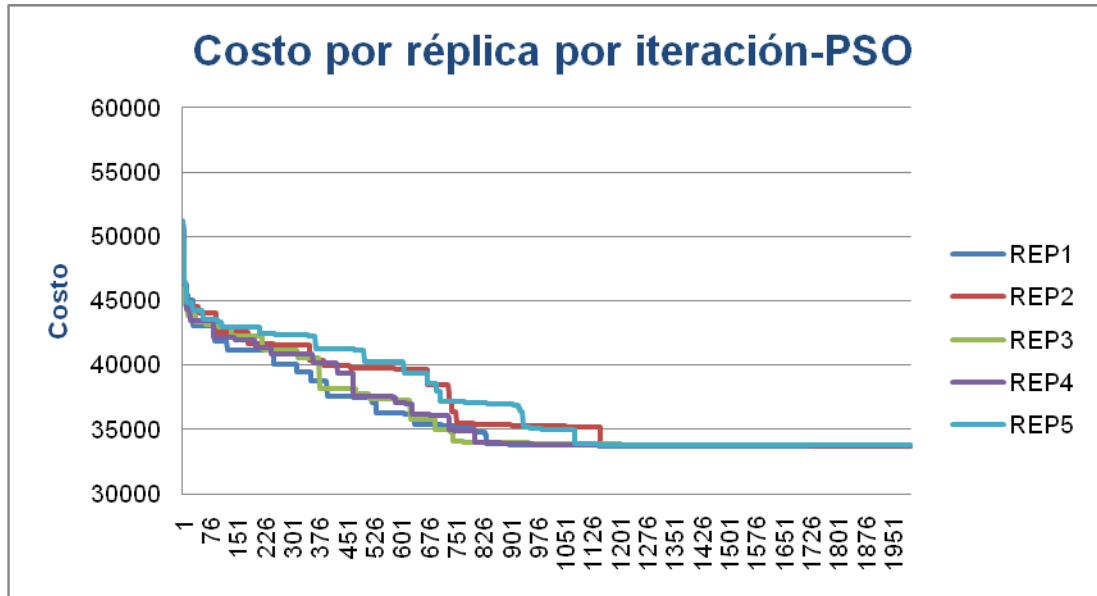
✓ Mejor configuración de celdas

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	2	3	2	2	2	4	4	2	4	2	4	4	3	2
2	4	2	3	4	2	2	4	4	4	4	4	4	4	4	2
3	2	4	3	4			2		4	3	4	4	4	4	
4		4	3	4						3	4	4	3		
5		3								2	1				
6		3								2	2				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	2	3	2	1	4	2	3	1	3	2	4	4	2	4
2	4	2	3	2	4	4	2	3	4	3	2	4	3	2	3
3	3		2	2		4	4	3	4	3	2		3		3
4	3			2			4		3				1		3
5							4		3						
6							2		3						

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	1	0	0	0	0	0	0	0	0	1	2
2	1	2	0	0	1	1	1	1	1	1	9
3	1	0	1	1	1	1	1	1	0	0	7
4	1	1	1	1	1	0	1	1	1	1	9
Total	4	3	2	2	3	2	3	3	2	3	

✓ Evolución por réplica por iteración



ANEXO W: RESULTADOS ALGORITMO PSO PROBLEMA 3

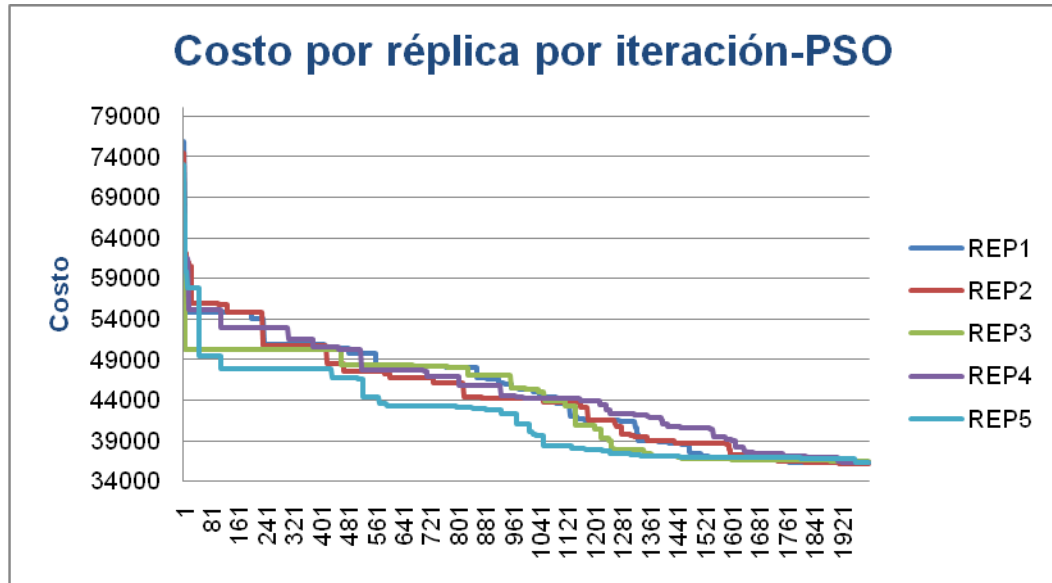
✓ Mejor configuración de celdas

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	5	3	4	4	4	5	2	1	5	5	1	5	4	3	4	3	4	5	1
2	5	5	1	2	4	3	5	2	5	3	4	5	5	4	5	4	5	5	4	5
3		5	2	2	5	5	3		5	1	2		5	4			5		4	5
4			2		5	4	4			3	2		3	4						2
5			2			1	4				5			1						2
6						4														
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	5	1	4	2	5	5	2	3	3	1	5	5	1	3	3	5	1	4	3	5
2	5	4	5	5	5	5	2	5	5	1	5	5	5	5	5	5	5	4	3	3
3		5			5	5	5	5	2	1		5	5		1			1	5	1
4		1			5	2	5	4	2	1					1			5	5	1
5		5			5	5	5	4	4	2					1				4	1
6					5		2		5										4	

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	0	1	0	0	1	0	0	2	1	1	6
2	1	1	0	0	0	1	1	0	1	1	6
3	0	1	0	1	0	1	1	0	1	1	6
4	0	1	2	1	0	0	2	1	0	1	8
5	1	1	1	2	1	1	1	1	0	1	10
Total	2	5	3	4	2	3	5	4	3	5	

- ✓ Evolución por réplica por iteración



ANEXO X: RESULTADOS ALGORITMO PSO PROBLEMA 4

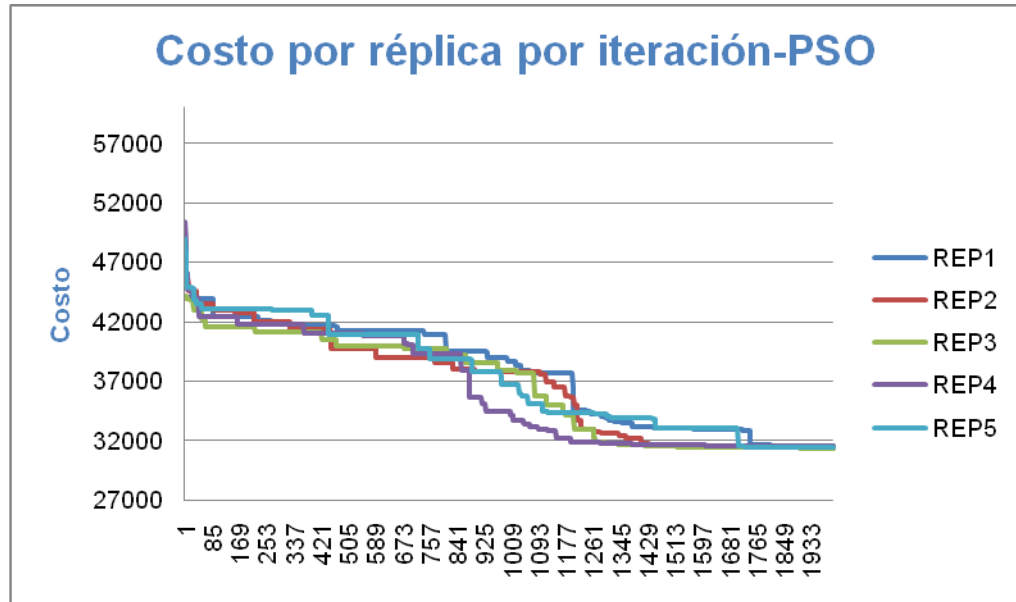
✓ Mejor configuración de celdas

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	3	2	4	5	5	4	2	2	5	1	5	3	5	5	4	4	2	4	4
2	5	4	1	4	1	2	5	2	2	5	5	5	3	5	5	5	4	2	4	4
3	5	4		5	1				2	5	5	5	5	4		5			5	
4				5	3				2				3	4		1			5	
5				5									3	2					5	
6																				
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	5	5	4	5	4	5	1	4	5	5	5	3	5	3	5	4	5	4	5	4
2	2	3	4	5	2	1	1	4	5	5	5	1	2	5	5	4	5	4	5	4
3	5		4		2		4	4	4				5			5	5		5	1
4	5								4											5
5	5																			5
6																				5

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	0	1	0	0	0	1	0	1	0	0	3
2	1	0	0	0	0	1	1	0	1	1	5
3	1	0	0	0	0	1	0	0	1	0	3
4	1	0	1	1	1	0	1	1	1	1	8
5	1	1	1	1	1	1	1	1	1	1	10
Total	4	2	2	2	2	3	3	3	4	3	

- ✓ Evolución por réplica por iteración



ANEXO Y: RESULTADOS ALGORITMO PSO PROBLEMA 5

✓ Mejor configuración de celdas

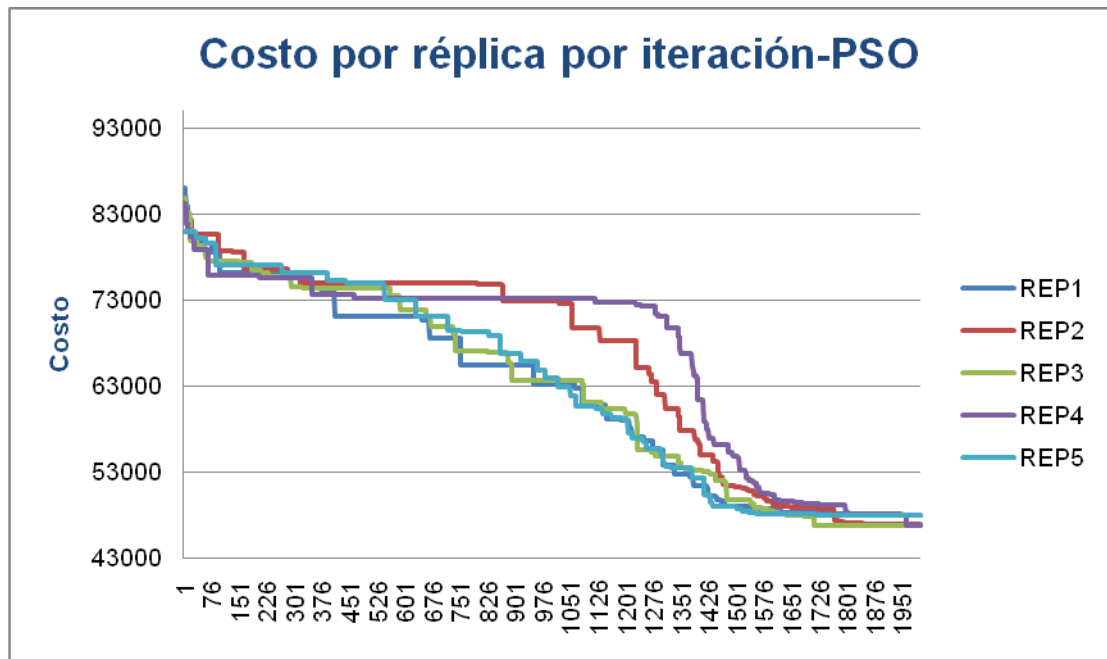
Op	PRODUCTO																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	6	1	6	1	6	1	5	6	4	2	5	6	1	6	5	6	6	2	6	6	2	6	6	2
2	6	6	6	2	1	6	2	6	2	6	6	6	6	1	6	6	6	5	6	5	6	2	6	6	2
3	2		6	2	6	6	6	6		6	5	6	6	5		6	6			5	6	1		6	3
4				6			6	6			5			2		4				5	6	1			
5											2			6							6				
6																					2				

Op	PRODUCTO																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	3	6	1	2	6	6	6	6	6	1	6	1	6	6	3	6	2	2	2	1	3	6	3	5	6
2	2	1	5	2	2	6	6	6	1	6	6	1	6	6	6	6	6	2	2	6	2	5	1	2	6
3		6			6	6	6	6		1	6	6			2	5	1			6	1	6		2	2
4		6			6	2				1	4					6	6			6				6	
5		6				6					6					2	6			6				6	
6		5				4					6					6	6			6					

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA																				Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	0	0	0	1	1	1	1	0	1	0	0	1	0	0	0	1	0	0	1	0	8
2	1	0	0	0	1	0	0	0	1	1	1	1	0	1	0	0	1	1	0	0	9
3	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2
4	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
5	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
6	2	1	1	1	2	1	2	1	2	1	0	1	1	0	1	1	1	1	0	0	20
Total	4	3	2	3	4	4	3	1	4	3	1	3	2	1	1	2	2	1	1	1	

- ✓ Evolución por réplica por iteración



ANEXO Z: RESULTADOS ALGORITMO PSO PROBLEMA 6

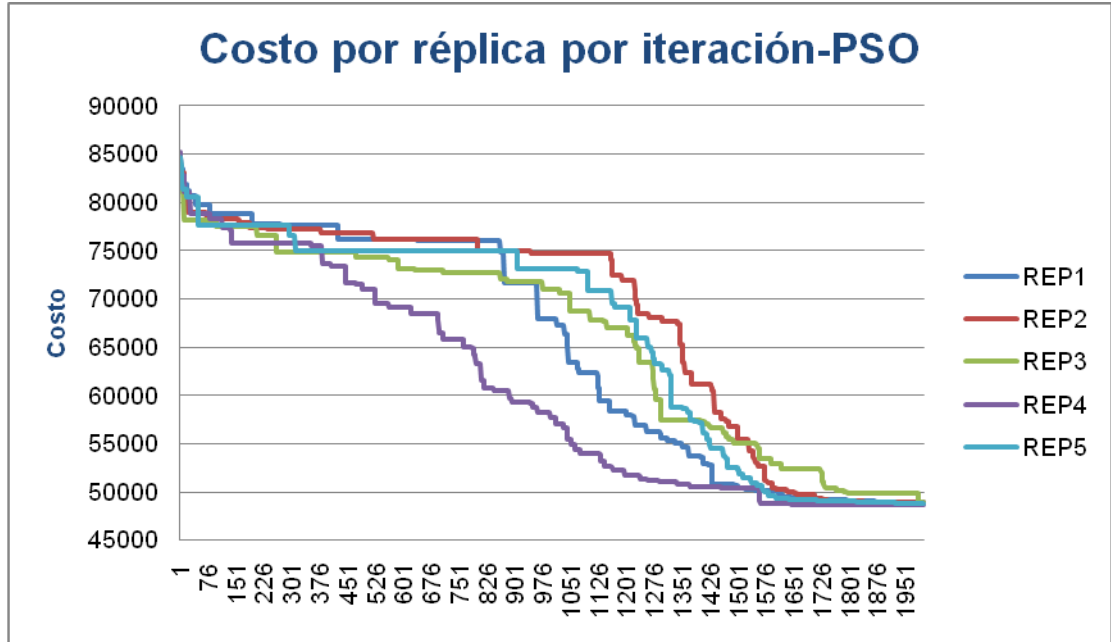
✓ Mejor configuración de celdas

Op	PRODUCTO																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	5	4	1	4	3	6	3	5	3	6	3	6	6	3	6	6	6	6	3	3	3	5	6	4	3	
2	5	6	6	6	6	6	1	6	6	6	6	3	6	1	3	6	6	4	3	3	3	6	2	6	3	
3		6	3	6	6	6	6			3	6	6		5		6		4	6	6	3		6	6	3	
4				3		6	4			6	6	6		1									3	3	3	
5				6		6				6	6			1									3	6		
6				1		3					6												3			
Op	PRODUCTO																									
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
1	6	4	6	6	6	6	6	6	6	4	6	3	3	3	6	3	6	6	1	6	3	6	6	6	6	
2	3	3	6	6	6	6	3	6	6	6	6	4	3	6	6	3	3	6	3	6	6	6	6	2	6	5
3	3	6	6	6		4	6	3	3	6	6	3	3	6		6		6	6	6	6	6	6	3	1	
4						4	4	3		6		3		6		6			6		3	6	6		3	
5						4		1		6				3					6			6			3	
6						6		4														3			3	

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA																				Total				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
1	0	0	0	1	1	1	1	0	1	0	0	1	0	0	0	1	0	0	1	0					8
2	1	0	0	0	1	0	0	0	1	1	1	1	0	1	0	0	1	1	0	0					9
3	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0					2
4	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0					2
5	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0					5
6	2	1	1	1	2	1	2	1	2	1	0	1	1	0	1	1	1	1	0	0	1				20
Total	4	3	2	3	4	4	3	1	4	3	1	3	2	1	1	2	2	1	1	1					

✓ Evolución por réplica por iteración



ANEXO. 1: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 2

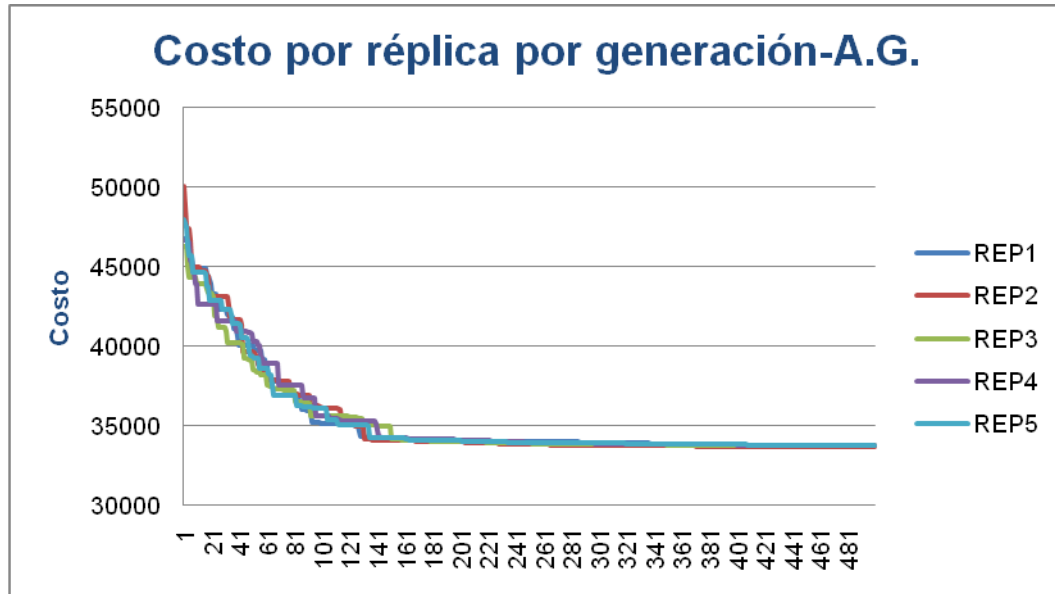
✓ Mejor configuración de celdas

Operación	PRODUCTO														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	1	1	2	1	4	4	1	1	2	2	3	3	3	1
2	2	1	1	2	1	4	4	1	4	1	2	3	3	3	1
3	2	3	3	2			4		4	1	2	3	4	3	
4		4	3	2						1	2	4	3		
5		1								1	2				
6		1								1	3				
Operación	PRODUCTO														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	3	1	1	3	1	2	4	3	2	2	1	1	2	4	4
2	2	1	1	2	1	2	4	3	2	2	3	1	2	4	4
3	3		2	2		2	3	3	1	2	3		1		3
4	3			2			3		1				1		3
5							3		1						
6							3		3						

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	1	1	0	1	1	1	1	1	0	1	8
2	1	1	1	0	1	1	1	1	1	1	9
3	1	1	1	1	0	1	0	1	0	1	7
4	1	0	0	0	1	0	1	0	1	0	4
Total	4	3	2	2	3	3	3	3	2	3	

- ✓ Evolución por réplica por iteración



ANEXO. 2: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 3

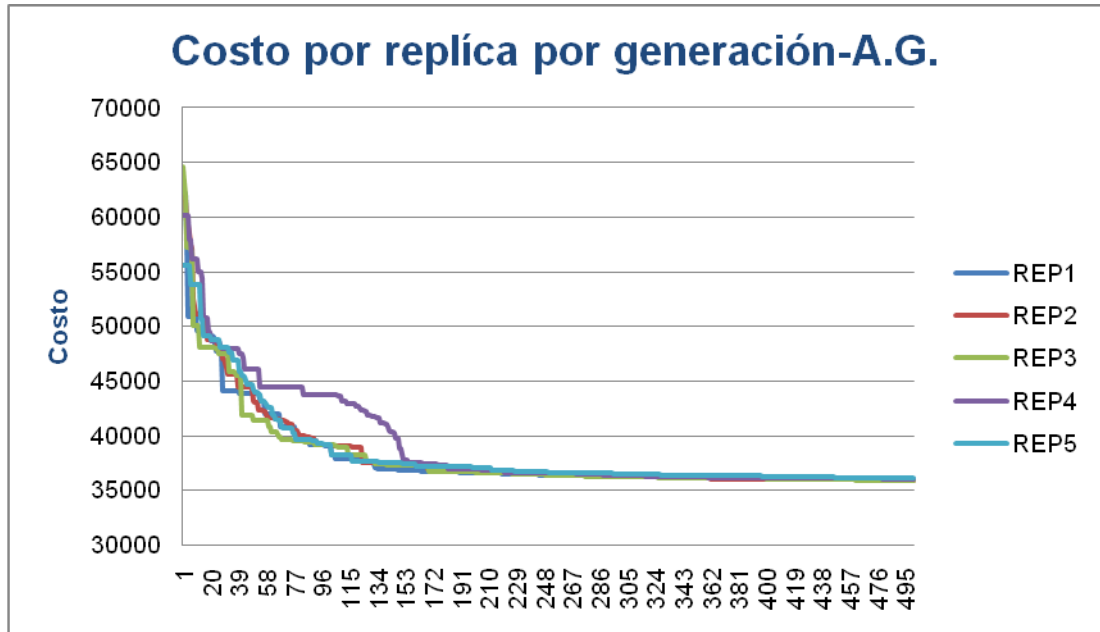
✓ Mejor configuración de celdas

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	3	3	3	1	4	2	1	3	5	4	4	2	3	2	2	5	1	3	2
2	5	3	3	1	4	4	2	3	3	3	4	5	5	4	4	2	5	2	2	2
3		5	5	3	1	4	2		3	1	3		3	2			1		2	1
4			4		1	2	1			1	3		1	2						1
5			4			2	2				5			1						1
6						2														
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	4	3	4	2	1	1	4	3	4	5	1	2	5	2	4	3	4	5	3	1
2	4	1	4	3	1	1	4	3	4	4	1	5	2	2	4	3	4	5	5	1
3		1			1	1	5	1	5	4		5	2		5			4	2	1
4		5			1	2	4	1	5	4					5			4	2	2
5		5			1	2	4	1	4	2					1				2	2
6					1		4		3										2	

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	0	1	1	1	1	1	1	1	0	1	8
2	0	1	1	1	0	1	1	1	1	1	8
3	1	0	0	1	0	1	1	0	1	1	6
4	0	2	1	1	0	0	1	2	1	1	9
5	1	1	0	0	1	0	1	0	0	1	5
Total	2	5	3	4	2	3	5	4	3	5	

✓ Evolución por réplica por iteración



ANEXO. 3: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 4

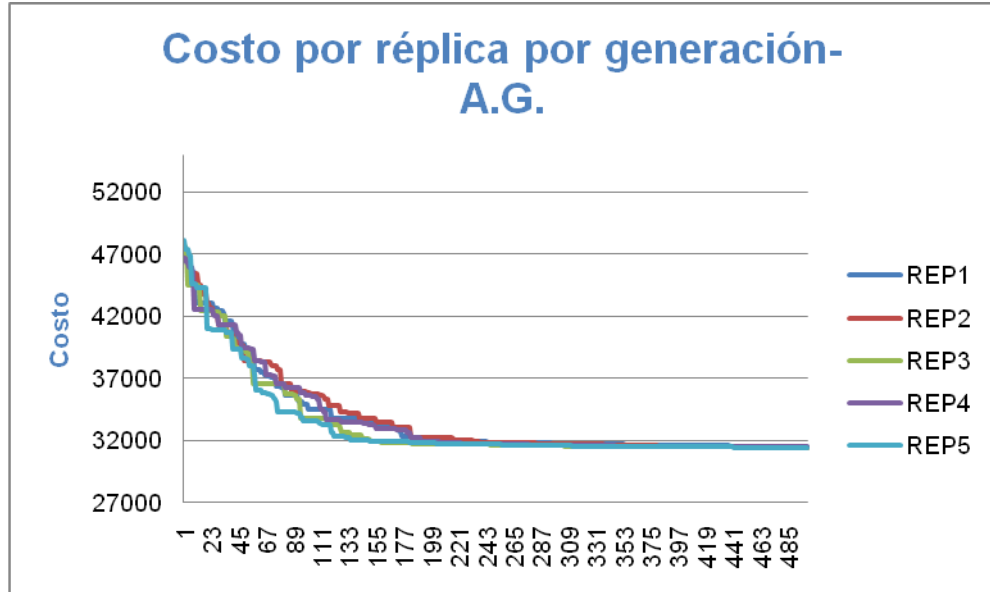
✓ Mejor configuración de celdas

Operación	PRODUCTO																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	4	5	3	2	4	3	4	2	5	3	5	1	2	3	1	5	3	4	1
2	2	4	5	2	2	1	3	4	1	3	3	2	1	2	3	2	5	3	2	4
3	3	4		4	1				4	1	3	2	1	2		2			3	
4				4	5				4				3	5		2			3	
5				5									3	5					3	
6																				
Operación	PRODUCTO																			
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	3	2	2	4	2	2	2	3	2	5	5	5	4	4	5	3	5	2	1	2
2	5	2	2	1	1	5	5	3	2	2	2	5	2	4	1	3	5	2	2	5
3	5		2		1		5	3	2				2			1	5		2	5
4	5								2											1
5	1																			1
6																				1

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA										Total
	1	2	3	4	5	6	7	8	9	10	
1	1	0	1	1	0	1	0	1	1	0	6
2	1	1	1	0	1	0	1	1	1	0	7
3	1	0	0	0	1	1	0	0	1	1	5
4	1	0	0	1	0	1	1	0	0	2	6
5	1	1	0	0	0	1	1	1	1	0	6
Total	5	2	2	2	2	4	3	3	4	3	

✓ Evolución por réplica por iteración



ANEXO. 4: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 5

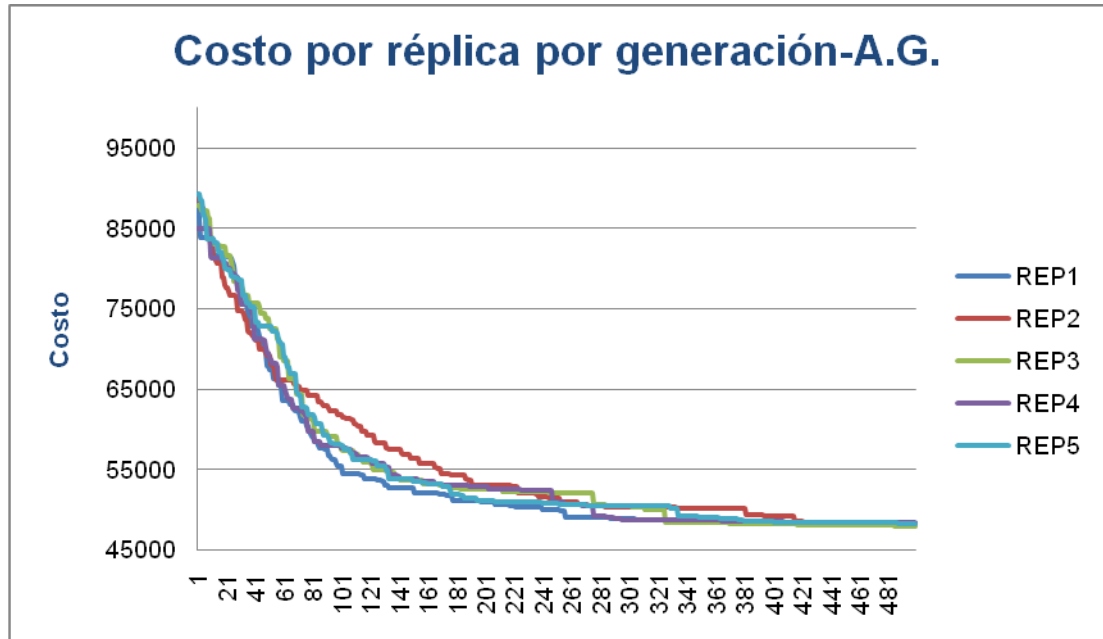
✓ Mejor configuración de celdas

Op	PRODUCTO																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	4	1	2	6	1	4	6	3	2	1	2	1	2	6	1	2	6	3	2	1	6	1	6	1
2	2	1	2	1	6	2	5	3	1	4	1	2	1	6	4	5	1	1	3	2	2	1	2	6	1
3	1		2	1	6	2	5	3		4	1	2	1	6		5	4			2	2	6		6	6
4				1			3	3			1			1		6				2	2	6			
5											1			2							2				
6																					5				
Op	PRODUCTO																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	6	1	1	1	2	1	5	5	1	2	3	2	1	5	2	2	3	5	5	1	2	4	4	2	6
2	5	1	2	1	5	6	1	5	1	1	3	1	5	5	2	2	3	6	2	4	1	1	3	3	6
3		1			6	4	2	1		1	3	1			2	1	3			4	6	1		3	1
4		1			6	1				1	4					3	2			4				2	
5		1				1					2					3	2			6				2	
6		1				3					2					1	5			6					

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA																				Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	1	1	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	13
2	1	1	1	1	0	1	0	1	0	0	0	1	1	0	0	1	1	0	0	1	11
3	0	0	0	0	1	1	1	0	2	1	0	0	0	0	0	0	0	0	0	0	6
4	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5
5	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	5
6	0	1	0	1	1	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	8
Total	4	4	2	3	4	4	3	1	5	3	1	3	2	1	1	2	2	1	1	1	

- ✓ Evolución por réplica por iteración



ANEXO. 5: RESULTADOS ALGORITMO GENÉTICO PROBLEMA 6

✓ Mejor configuración de celdas

Op	PRODUCTO																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	3	5	2	2	4	6	4	1	4	2	6	1	3	6	6	3	6	4	3	1	6	2	3	3
2	1	1	6	2	2	1	6	1	3	4	1	4	1	6	2	6	4	6	4	3	4	1	6	1	3
3		3	6	6	2	1	3			4	6	1		1		6		3	6	6	6		6	2	6
4				4		1	6			6	6	1		5									5	6	6
5				6		2				2	3			4									6	6	
6				6		2					3												3		

Op	PRODUCTO																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	6	6	6	6	1	6	1	1	5	2	2	2	3	3	4	2	1	2	2	3	5	3	6	2	6
2	3	3	6	2	1	6	2	5	6	2	6	2	3	6	5	2	2	3	2	1	3	3	6	6	6
3	5	6	4	2		6	6	5	2	2	6	6	3	3		6		1	4	1	4	3	6	6	6
4						2	6	3		3		3		3		3			2		6	2	3		4
5						1		2		2				3					2			2			4
6						5		2														2			5

✓ Necesidades de máquina

Celda	TIPO DE MÁQUINA																				Total				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
1	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0					7
2	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0	1	0	0					11
3	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	1	0	0	0	1					8
4	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0					6
5	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0					4
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	1				13
Total	2	3	2	2	5	3	1	3	2	2	3	2	2	3	2	4	1	3	2	2					

✓ Evolución por réplica por iteración

