

UNIVERSIDAD
NACIONAL
DE COLOMBIA

**UN ALGORITMO DE ESTIMACIÓN DE
DISTRIBUCIÓN PARA SOLUCIONAR PROBLEMAS
DE PROGRAMACIÓN EN AMBIENTE FLOWSHOP
CON BLOQUEO Y CON MÚLTIPLES OBJETIVOS**

John Andersson Valencia Palacio

Universidad Nacional de Colombia
Facultad de Minas-Ingeniería de la Organización
Medellín, Colombia

2017

UN ALGORITMO DE ESTIMACIÓN DE DISTRIBUCIÓN PARA SOLUCIONAR PROBLEMAS DE PROGRAMACIÓN EN AMBIENTE FLOWSHOP CON BLOQUEO Y CON MÚLTIPLES OBJETIVOS

John Andersson Valencia Palacio

Trabajo de investigación presentado como requisito parcial para optar al título de:
Magister en Ingeniería Industrial

Director:

Ph.D. Elkin Rodríguez Velásquez

Línea de Investigación:

Programación de la producción

Universidad Nacional de Colombia
Facultad de Minas-Ingeniería de la Organización
Medellín, Colombia
2017

A mis padres que con su esfuerzo, apoyo y comprensión han encaminado mis proyectos de vida

Agradecimientos

Agradezco a mi director de tesis, Elkin Rodríguez Velásquez, quien me ha acompañado desde los cursos de pregrado en los cuales fomentó la pasión por la investigación en los temas de programación de producción y aportó en mi proceso formativo como profesional, docente y persona.

A mis profesores de pregrado y posgrado, en especial a Gloria Inés Jiménez, Gloria Patricia Jaramillo, Alexander Correa Espinal por su aporte a mi formación profesional y por ser ejemplo de dedicación, trabajo y amor por la enseñanza.

A mi alma máter, la Universidad Nacional de Colombia por generar tan valiosa formación profesional y humana; y por permitirme el cumplimiento de muchos sueños y proyectos de vida.

A mis padres, familiares y amigos por el apoyo, la confianza y la comprensión en este proceso formativo.

A Dios, por todas sus bendiciones.

Resumen

La programación de producción tiene un impacto relevante sobre el uso eficiente de los recursos, reducción de costos y cumplimiento de los objetivos como servicio al cliente, entregas oportunas y satisfacción de la demanda. En un entorno cada vez más competitivo, las organizaciones se ven en la necesidad de aplicar herramientas, procedimientos y estrategias que les permitan estar a la vanguardia. En ese sentido, el uso de las metaheurísticas para resolver problemas de programación y secuenciación de trabajos va en aumento, ya que se han demostrado sus fortalezas para la búsqueda de soluciones eficientes, oportunas, rápidas y de buena calidad. Adicionalmente, las organizaciones buscan satisfacer o cumplir varios objetivos o metas de manera simultánea como entregar a tiempo al mínimo costo, entre otros. Así, se propone y desarrolla un metaheurístico de estimación de distribución para un ambiente de programación tipo flowshop con restricciones de bloqueo y con múltiples objetivos. A partir de la experimentación, se evidencia un adecuado rendimiento del algoritmo en cuanto a las soluciones encontradas y al rendimiento, que no se ve afectado por el número de trabajos, ni de máquinas a considerar en el problema.

Palabras clave: Algoritmo de Estimación de Distribución, Flowshop con bloqueo, problema de programación multiobjetivo, tardanza, makespan

Abstract

The production scheduling has a relevant impact on the efficient use of resources, reduction of costs and fulfilment of the objectives such as customer service, timely deliveries and demand satisfaction. In an increasingly competitive environment, organizations are in the need for tools, procedures and strategies that allow them to be at the forefront. In this sense, the use of metaheuristics for solving problems of scheduling and sequencing of jobs is increasing, since their strengths aiming to pursuit fast, timely, efficient and of good quality solutions, have been shown. In addition, organizations seek to meet several objectives or goals simultaneously, such as on time and the minimum cost deliveries, among others. Thus, an estimation of distribution metaheuristic for a flowshop scheduling problem with blocking and multiple objectives is proposed and developed. As a result of the experimentation, there is evidence of an appropriate performance of the algorithm in terms of the solutions found and the performance, which is not affected by the number of jobs or machines to be considered in the problem.

Keywords: Estimation Distribution Algorithm, Flowshop with blocking, multiobjective scheduling problem, Tardiness, makespan

Contenido

	Pág.
Resumen	IX
Lista de figuras	XIV
Lista de tablas	XVI
Introducción	1
Objetivos	8
1. Capítulo Marco Teórico	11
1.1 ADMINISTRACIÓN DE LA PRODUCCIÓN	11
1.1.1 Nivel Estratégico	11
1.1.2 Nivel Táctico	11
1.1.3 Nivel operativo	12
1.2 CONFIGURACIONES PRODUCTIVAS	12
1.2.1 Máquina simple o individual	12
1.2.2 Máquinas paralelas o uniformes.....	13
1.2.3 Flow Shop	14
1.2.4 Flexible Flow Shop	15
1.2.5 Job shop	15
1.2.6 Open Shop.....	17
1.2.7 Flexible Job shop	17
1.2.8 Supply Chain.....	17
1.3 RESTRICCIONES EN EL SISTEMA PRODUCTIVO	18
1.3.1 Interrupciones (No preemptions):	18
1.3.2 Precedencias:	18
1.3.3 Tiempos de preparación secuencialmente dependientes:	18
1.3.4 Familia de trabajos:	19
1.3.5 Procesamiento por Lotes:	19
1.3.6 Ruptura (Breakdowns):	19
1.3.7 Permutación:	19
1.3.8 Bloqueo:.....	20
1.3.9 Sin Esperas (No wait):.....	20
1.3.10 Recirculación:	20
1.4 FUNCIONES OBJETIVO.	20
1.5 METAHEURÍSTICAS APLICADAS EN LA GESTIÓN DE PRODUCCIÓN.	22
1.5.1 Métodos Constructivos.....	22
1.5.2 Búsqueda local iterada.....	22

1.5.3	Recocido Simulado:.....	23
1.5.4	Búsqueda de entorno variable (Variable Neighbourhood Search, VNS)	23
1.5.5	Búsqueda Tabú	23
1.5.6	Grasp (Greedy Randomized Adaptive Search Procedure).....	24
1.5.7	Colonia de hormigas.....	24
1.5.8	Algoritmos Genéticos.....	24
1.6	Algoritmos Multiobjetivo.....	25
1.6.1	Algoritmo VEGA.....	25
1.6.2	Algoritmo MOGA.....	25
1.6.3	SPEA.....	26
1.6.4	NSGA	26
1.6.5	SPEA2.....	27
1.6.6	NSGA-II	27
1.6.7	NSGA-III	28
1.7	Métricas de Desempeño.....	29
1.8	ANTECEDENTES	30
2.	Capítulo 2. Metodología multiobjetivo para programar un sistema flowshop con bloqueo	35
2.1	Descripción del problema	35
2.1.1	Modelo matemático	36
2.2	Descripción del Algoritmo de Estimación de Distribución	38
2.2.1	Seudo código.....	38
2.2.2	Representación de una solución.	39
2.2.3	Selección de individuos	40
2.2.4	Estimación de la función de distribución de probabilidad.	41
2.2.5	Generación de nuevos individuos y actualización de la población.	43
2.3	Descripción del Algoritmo Genético Multiobjetivo	43
2.3.1	Seudo código.....	44
2.3.2	Representación del cromosoma	45
2.3.3	Selección de Individuos	45
2.3.4	Operador de Cruce	46
2.3.5	Reparación de los hijos.....	47
2.3.6	Operador de Mutación.	48
3.	Capítulo 3. Experimentación y Resultados	49
3.1	Diseño del plan experimental	49
3.2	Resultados Obtenidos.....	51
3.2.1	Experimentos 50 trabajos-20 máquinas.....	51
3.2.2	Experimentos con 50 trabajos y 40 máquinas.....	54
3.2.3	Experimentos con 100 trabajos y 20 máquinas.....	57
3.2.4	Experimentos con 100 trabajos y 40 máquinas.....	60
3.3	Análisis del Algoritmo de Estimación de Distribución.....	63
4.	Conclusiones y recomendaciones	69
4.1	Conclusiones.....	69
4.2	Trabajos futuros	71
	A. Anexo: Resultados obtenidos con el Algoritmo de Estimación de Distribución.....	73

B. Anexo: Resultados obtenidos con el Algoritmo Genético.....	87
Bibliografía	101

Lista de figuras

	Pág.
Ilustración 1.1. Sistema productivo de máquina simple	13
Ilustración 1.2. Sistema productivo máquina en paralelo.....	14
Ilustración 1.3. Sistema productivo flowshop con m máquinas.....	14
Ilustración 1.4- Sistema productivo Flexible flowshop con m máquinas	15
Ilustración 1.5 Sistema productivo tipo Job shop con m máquinas.....	16
Ilustración 2.1. Una solución para 3 trabajos y 4 máquinas en BFSP.....	35
Ilustración 3.1 Diagrama de Pareto n=50, m=20	53
Ilustración 3.2 Gráfico de efectos principales.n=50, m=20	53
Ilustración 3.3. Gráficos de interacción entre factores n=50, m=20	54
Ilustración 3.4. Diagrama de Pareto estandarizado n=50, m=40	56
Ilustración 3.5. Gráfico de efectos principales n=50, m=40	56
Ilustración 3.6. Gráfico de interacciones entre factores n=50, m=40	57
Ilustración 3.7. Diagrama de Pareto Estandarizada para n=100, m=20.....	59
Ilustración 3.8 Gráfica de efectos principales para n=100, m=20	59
Ilustración 3.9. Gráfico de interacciones para n=100, m=20.....	60
Ilustración 3.10 Diagrama de Pareto Estandarizado n=100, m=40.....	61
Ilustración 3.11. Gráfico de efectos principales n=100, m=40	62
Ilustración 3.12 Gráfico de interacción de efectos n=100, m=40	63
Ilustración 3.13. Hipervolumen vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 50 individuos.....	64
Ilustración 3.14 Tardanza Total vs Generaciones con EDA, 40 trabajos y 6 máquinas ...	65
Ilustración 3.15 Makespan vs Generaciones con EDA, 40 trabajos y 6 máquinas.....	65
Ilustración 3.16 Hipervolumen vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos.....	66
Ilustración 3.17 Tardanza Total vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos.....	66
Ilustración 3.18 Cmax vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos.....	67

Lista de tablas

	Pág.
Tabla 1-1. Ranking de métricas más citadas para problemas multiobjetivo.....	30
Tabla 1-2 EDA para secuenciamiento.....	33
Tabla 2-1. Clasificación por frentes de dominancia.....	40
Tabla 2-2. Matriz Ortogonal.....	46
Tabla 3-1. Factores y niveles diseño experimental.....	49
Tabla 3-2 Tabla ANOVA n=50, m=20.....	52
Tabla 3-3. ANOVA para n=50, m=40.....	55
Tabla 3-4. ANOVA para n=100 y m=20.....	58
Tabla 3-5 ANOVA para n=100, m=40.....	61

Introducción

La dirección de operaciones es la actividad de administrar los recursos con los cuales se producen y se entregan productos y servicios. La función de operaciones es la parte de la organización que es responsable de esta actividad. Cada organización tiene una función de operaciones dado que cada organización produce algún tipo de bien y/o servicio (Slack, 2010).

Los administradores de operaciones son los responsables del suministro de una capacidad suficiente para satisfacer las necesidades de sus empresas. Deben tomarse decisiones de capacidad en función de los pronósticos de demanda y de los planes de desarrollo con el fin de proporcionar la capacidad adecuada para el largo, mediano y corto plazo (Schroeder et al, 2011).

Las decisiones de capacidad deben alinearse con la estrategia de operaciones de una empresa. La estrategia de operaciones proporciona un mapa que se usa al tomar decisiones de la cadena de suministros encaminadas a crear una red de organizaciones cuyo trabajo y producto final se apliquen a la satisfacción de las necesidades de productos y servicios de los clientes. Las decisiones de capacidad se basan en estimaciones pronosticadas de la demanda futura (Schroeder et al, 2011).

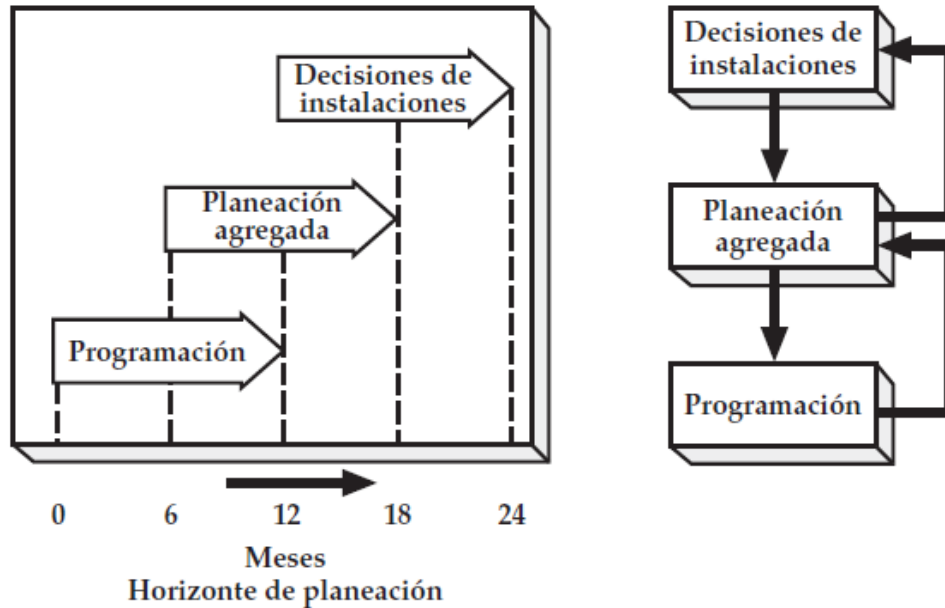


Ilustración 1. Horizonte de planeación para cada etapa de administración.

Fuente: (Schroeder, 2011).

En la ilustración 1, se muestra que las instalaciones, la planeación agregada y la programación de la producción integran una jerarquía de decisiones de capacidad respecto de una planeación de operaciones que puede extenderse de un largo plazo, hasta un mediano o uno corto. Primero, las decisiones de planeación de las instalaciones son de carácter a largo plazo y se toman para obtener la capacidad física que debe planearse, desarrollarse y estructurarse antes del uso que se pretende. Posteriormente, la planeación agregada determina el nivel de la fuerza laboral y el de la producción final para un mediano plazo dentro de la capacidad disponible. Por último, la programación de operaciones consiste en decisiones a corto plazo que se restringen por la planeación agregada y aplica la capacidad disponible asignándola a actividades específicas (Schroeder Et Al, 2011).

La programación de producción

La programación es un proceso de toma de decisiones que es usado como una base regular de muchas industrias manufactureras y empresas de servicios. Se trata de la asignación de recursos para algunas tareas dado un periodo de tiempo y su meta es optimizar uno o más objetivos (Pinedo, 2012).

Los recursos y las tareas en una organización pueden tomar diferentes formas. Los recursos pueden ser máquinas en un taller, pistas en un aeropuerto, el personal en una construcción, unidades de procesamiento en un entorno informático. Así, las tareas pueden ser las operaciones en un proceso de producción, etapas en un proyecto de construcción, la ejecución de programas de computador, entre otros. Los objetivos pueden tomar diferentes formas; un objetivo puede ser la minimización de los tiempos de terminación de la última tarea y otro puede ser la minimización del número de tareas terminadas después de sus respectivas fechas previstas de entrega (Pinedo, 2012).

Problemas de programación tipo Flowshop

La programación en ambientes flowshop es un área de investigación atractiva en manufactura. No solo en el campo del estudio teórico, pues también se ha generado interés en su aplicación en la industria y en muchas otras situaciones de la vida real (Yenisey Et Al, 2014). En la configuración flowshop, los trabajos fluyen desde una máquina inicial, a través de unas máquinas intermedias y posteriormente a una máquina final, la cual termina los trabajos (Baker, 2010).

El problema flowshop es definido por un conjunto $N= 1,2,3,\dots,n$ trabajos que tienen que ser procesados en un conjunto $M=1,2,3,\dots,m$ máquinas. El tiempo de procesamiento de cada trabajo $i \in N$ en cada máquina $j \in M$ es conocido y denotado como P_{ij} . Todos los N trabajos visitan las máquinas en el mismo orden. El objetivo es encontrar una secuencia de procesamiento de los trabajos que optimice un criterio dado (Minella, 2007).

El Problema multiobjetivo.

Un problema de múltiples objetivos se puede representar matemáticamente mediante un vector de p dimensiones, en donde cada dimensión representa una función objetivo, que a su vez es una función de n variables x , así:

$$Z(x)=[Z_1(x), Z_2(x), \dots, Z_p(x)]$$

Los métodos de análisis multiobjetivo tratan de identificar la mejor o las mejores soluciones considerando múltiples objetivos simultáneamente (Smith Et. Al.,2000).

Los problemas de la vida real naturalmente incorporan problemas multiobjetivo, el problema de programación en entornos flowshop con múltiples objetivos ha sido estudiado en los recientes años. Dos o más objetivos como la suma de los tiempos de terminación de los trabajos, su tardanza, su tiempo de flujo y diferentes combinaciones son consideradas en esos estudios (Yenisey, 2014).

Los enfoques para resolver los problemas multiobjetivo son generalmente divididos en tres clases dependiendo del rol del tomador de decisión en el proceso de solución (Arroyo Et. Al, 2011):

1. Un enfoque a priori: el tomador de decisión aporta toda la información necesaria para iniciar el proceso de toma de decisión. Esto se puede hacer de dos maneras: la primera en relación a la minimización de funciones con objetivos ponderados (Chou Et. Al.;1999); la segunda forma, a partir de la optimización jerárquica de los objetivos. Es decir, la optimización del primer objetivo, la optimización del segundo objetivo teniendo en cuenta el valor óptimo del primer objetivo y así sucesivamente (Gupta, 2002).

2. Un enfoque a posteriori: Inicialmente, un conjunto de soluciones eficientes es desarrollado. Entonces, el tomador de decisiones elige una de las soluciones de ese conjunto (Arroyo Et. Al, 2004).

3. Enfoque interactivo: el tomador de decisiones introduce un conjunto de preferencias durante el proceso de solución. En cada paso del procedimiento, el tomador de decisión expresa cual es la solución preferente. El proceso determina un compromiso de satisfacción entre los objetivos considerados por el tomador de decisiones (Allouche, 2009).

Algoritmos de estimación de distribución (EDA).

EDA es un algoritmo evolucionario propuesto por Mühlenbein y PaaB en 1996, el cual genera nuevos individuos con respecto a un modelo probabilístico, generado a partir de la población anterior.

En general, el procedimiento seguido por un EDA básico puede ser presentado así: Comenzar con una población inicial generada aleatoriamente, selección de una

subpoblación de M individuos de la población a través de un método de selección asociado con la medida de desempeño. Después, se estima una distribución de probabilidad de los individuos seleccionados a partir de un modelo probabilístico. Entonces, se genera una nueva población de acuerdo con la distribución de probabilidad estimada. Finalmente, algunos individuos en la población actual son reemplazados con la nueva población generada (Muhlenbein y PaaB, 1996).

Teniendo en cuenta que el uso de los algoritmos de estimación de distribución en los problemas de programación es relativamente nuevo; que actualmente los problemas de producción reales son casi todos multiobjetivo (Pelikan, 2007); y que además no se evidencia el uso de esta herramienta para resolver problemas de programación de producción en ambientes flowshop con múltiples objetivos (Yenisey, 2014); **¿Es posible desarrollar un algoritmo de estimación de distribución que genere soluciones a problemas de programación de producción tipo flowshop con múltiples objetivos y con restricción de bloqueo entre máquinas?**

JUSTIFICACIÓN

La programación es un campo de estudio que consiste en la utilización de recursos limitados en trabajos a ser procesados en alguna secuencia con el fin de optimizar una o más medidas de desempeño. Flowshop es una configuración de producción en el cual se tiene el mismo orden de procesamiento de trabajos a través de un conjunto de máquinas. En las últimas décadas, se ha puesto gran atención a la búsqueda de soluciones en problemas flowshop. Diferentes enfoques se han presentado para resolver este problema. En problemas de pocos trabajos, se utiliza programación entera mixta y algoritmos de ramal y límite; mientras que, en problemas con muchos trabajos, se utilizan heurísticos y metaheurísticos (Karimi Et al.,2014).

Al hacer la revisión de literatura, se evidencia que la mayor proporción de trabajos e investigaciones realizadas sobre problemas de programación flowshop están enfocadas a la optimización de un solo objetivo, entre los que se encuentra makespan, tardanza, flujo máximo (Karimi Et al.,2014). Sin embargo, algunos autores citan la importancia que tienen los problemas de programación con enfoque multiobjetivo con base en que la mayoría de los problemas reales de la industria tienen múltiples criterios de optimización y en donde

dichos criterios se encuentran en conflicto; es decir, la solución que genera el mejor desempeño bajo un criterio seleccionado, simultáneamente implica un nivel inferior al deseado en otro criterio (Arroyo Et al, 2005).

Por otra parte, en los últimos años se destaca un incremento en el interés de los investigadores en resolver problemas de programación flowshop con restricción de bloqueo, teniendo en cuenta su aplicación en la industria química, metalúrgica, entre otras. (Pan y Wang, 2012). Este problema con bloqueo ha sido trabajado por múltiples autores con el fin de optimizar una sola medida de desempeño (Toumi,S et al, 2013); (Han et al, 2012). Sin embargo, en lo que se refiere al problema con restricción de bloqueo y múltiples medidas de desempeño, en la revisión de literatura no se evidencia trabajos ni investigaciones en este campo específico.

En lo que respecta a las técnicas de solución aplicadas para los problemas de programación se evidencia el uso de métodos exactos como ramal y límite, métodos heurísticos y metaheurísticos como algoritmos genéticos, colonia de hormigas, búsqueda tabú, entre otros. En ese sentido, los algoritmos evolutivos que utilizan el aprendizaje de máquinas han tenido un interés investigativo que va en aumento durante los últimos años. (Echegoyen, 2015). Dentro de este grupo de algoritmos, se encuentran los Algoritmos de Estimación de distribución (EDA). Los EDA han sido empleados para resolver problemas de optimización combinatoria. Algunas aplicaciones exitosas se dieron en el problema de asignación cuadrática, (Zhang et al., 2006), el problema de la mochila (Hui Li et al., 2004), el problema de las n-reinas (Paul TK and Iba H, 2002), problema del agente viajero (Robles et al., 2006). En trabajos recientes, el EDA ha sido utilizado para resolver problemas de optimización multiobjetivo (Zhang et al. 2008, Hui Li et al., 2004), (Jarboui, 2009).

De acuerdo con la revisión realizada, y teniendo en cuenta que se presenta interés permanente en la investigación enfocada en problemas de programación flowshop, que la restricción de bloqueo tiene tanto aplicación académica como industrial, que dicha configuración con bloqueo no se ha estudiado ampliamente cuando se trata de optimizar múltiples criterios de desempeño y dado el creciente interés y los buenos resultados que ha presentado la utilización de los algoritmos de estimación de distribución; se evidencia una oportunidad de investigación y generación del conocimiento en el desarrollo de algoritmos de estimación de distribución como herramienta de solución para problemas de

programación de producción flowshop con restricciones de bloqueo y con múltiples objetivos.

Objetivos

Objetivo General

Desarrollar un algoritmo de estimación de distribución que permita resolver problemas de programación tipo Flowshop con bloqueo y con múltiples objetivos.

Objetivos Específicos

Revisar el estado de investigación actual en problemas de programación de producción tipo Flow Shop con bloqueo y con múltiples objetivos.

Revisar el estado del arte y de la investigación de los algoritmos de estimación de distribución y el uso que se les ha dado para resolver problemas de programación de producción.

Identificar en la literatura, los algoritmos genéticos que se utilizan para resolver problemas de programación de producción tipo Flow Shop con múltiples objetivos.

Identificar las diferentes metodologías que se utilizan en los problemas con múltiples objetivos para proponer una solución a los problemas de programación de producción

Definir el problema de programación de programación tipo Flow Shop con bloqueo y con múltiples objetivos.

Proponer un Algoritmo de Estimación de Distribución para encontrar la solución de un problema de programación tipo Flow Shop con bloqueo y con múltiples objetivos.

Elaborar un algoritmo genético que genere soluciones al problema de programación tipo Flow Shop con bloqueo y con múltiples objetivos.

Implementar el algoritmo genético y el algoritmo de estimación de distribución en un lenguaje de programación y validar en cuánto a factibilidad de las soluciones dadas.

Llevar a cabo un diseño experimental que permita comparar los resultados obtenidos por el algoritmo de estimación de distribución y el algoritmo genético.

Analizar los resultados obtenidos en el diseño experimental.

1. Capítulo Marco Teórico

1.1 ADMINISTRACIÓN DE LA PRODUCCIÓN

Distintos enfoques se han propuesto para caracterizar la función de producción. Entre estos están la producción como sistema, la producción como función organizacional, y la producción como conjunto jerárquico de decisiones (Becerra, 2008). El enfoque jerárquico de la producción establece 3 niveles de decisión: el nivel estratégico, el nivel táctico y el nivel operativo (Machuca, 1995).

1.1.1 Nivel Estratégico

El proceso de gestión de la producción debe iniciarse con el establecimiento de la estrategia de producción, la cual establece los criterios de competitividad sobre los cuales serán invertidos todos los esfuerzos del departamento de producción a través de la toma de decisiones estratégicas en los términos del largo plazo (Gaither, 2000), (Chase y Aquilano, 2000).

1.1.2 Nivel Táctico

A partir de la estrategia de producción se analizan los aspectos tácticos y operativos. De manera amplia, el nivel táctico de la producción se encarga de definir las necesidades de entregar un volumen de producción (en términos agregados) dadas unas capacidades estimadas para este periodo de tiempo (Machuca, 1995).

1.1.3 Nivel operativo

Por último, en el nivel operativo se llevan a cabo las asignaciones de los recursos en el corto plazo con el fin de dar cumplimiento a los presupuestos de producción y se ejecutan los programas de producción (Pinedo, 2005). La programación de producción o scheduling, se puede describir como la asignación de un conjunto de recursos en un período de tiempo, para llevar a cabo un conjunto de tareas (Phanden, 2012).

1.2 CONFIGURACIONES PRODUCTIVAS

En la investigación se encuentra la clasificación de los talleres de trabajo, la cual depende de la secuencia de operación, el número y tipo de máquinas. (Pinedo,2005), (Zandieh et al. ,2006) y (Allahverdi et al.,2008) presentan configuraciones similares con criterios de clasificación de problemas de programación secuencial basada en entornos de recursos. A continuación, se presenta la clasificación general encontrada (Ruiz, 2015).

1.2.1 Máquina simple o individual

Es un sistema en el cual solo existe una máquina (Pinedo, 2012; Allahverdi et al., 2008; Castrillón et al., 2010). En la ilustración 1.1 se ilustra el sistema productivo para una máquina o puesto de trabajo con n pedidos y m clientes; el proceso depende de varios factores, entre ellos: la programación y de las prioridades del operario o administrador; tiempo de llegada de los pedidos; tiempos de liberación de pedidos y tiempos de entrega.

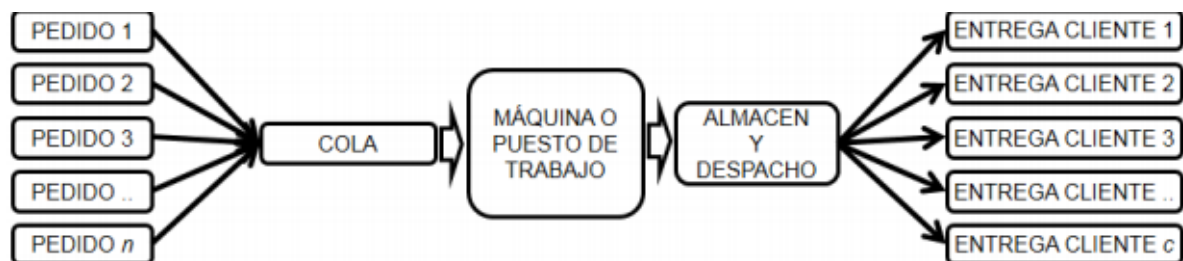


Ilustración 1.1. Sistema productivo de máquina simple

Fuente: (Ruiz, 2015)

La cola se refiere a la espera de los pedidos que serán procesados, esta condición propicia la generación de un cuello de botella. El procesamiento se puede programar con métodos tradicionales, la literatura establece los tres más utilizados así: el primero es “el tiempo de procesamiento más corto”, the shortest processing time (SPT), presentado por Smith (1956), donde el pedido que tiene el menor tiempo de procesamiento se ejecuta primero. El segundo es el “primero en entrar primero en salir”, First-Input-First-Output (FIFO) expuesto por Blackstone et al. (1982), los pedidos se elaboran en la medida en que llegan. El tercero, explicado por Chen y Vestjens (1997), es “el tiempo de procesamiento más largo”, the longest processing time (LPT) (Ruiz, 2015).

1.2.2 Máquinas paralelas o uniformes

En este sistema varias máquinas cumplen la misma función, aunque puede diferir en la capacidad de proceso, la calidad del trabajo y los tiempos de procesamiento, un operario puede trabajar sobre una o varias máquinas (Pinedo, 2012; Allahverdi et al., 2008; Castrillón et al. 2010). En la ilustración 1.2 se ilustra el sistema productivo para m máquinas en paralelo con n pedidos y c clientes, el proceso depende de las condiciones y factores mencionados en el ítem anterior. La asignación de cada pedido puede hacerse en la medida que se desocupe una máquina, si dos o tres máquinas están libres se asigna el pedido a la máquina con mayor velocidad de procesamiento. (Ruiz, 2015)

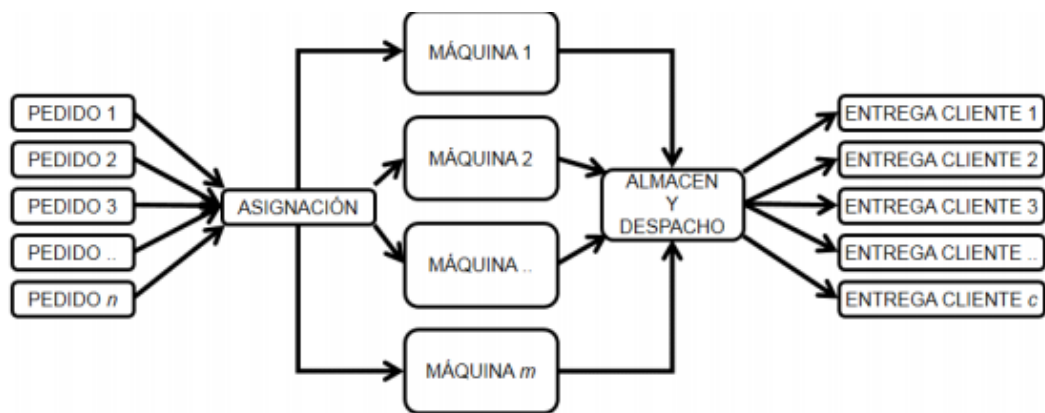


Ilustración 1.2. Sistema productivo máquina en paralelo

Fuente: (Ruiz, 2015)

1.2.3 Flow Shop

Los pedidos siguen la misma secuencia de operación, el patrón de flujo entre máquinas es el mismo. Todos los pedidos pasan por todas las máquinas (Pinedo, 2012; Allahverdi et al., 2008; Pan et al., 2011). En la ilustración 1.3 se ilustra el sistema productivo para m máquinas con configuración flowshop con n pedidos y c clientes. El tiempo de procesamiento, makespan, dependerá de la asignación de los pedidos, pueden generarse demoras o colas entre máquinas, un pedido no será procesado en la máquina hasta que ella esté libre.

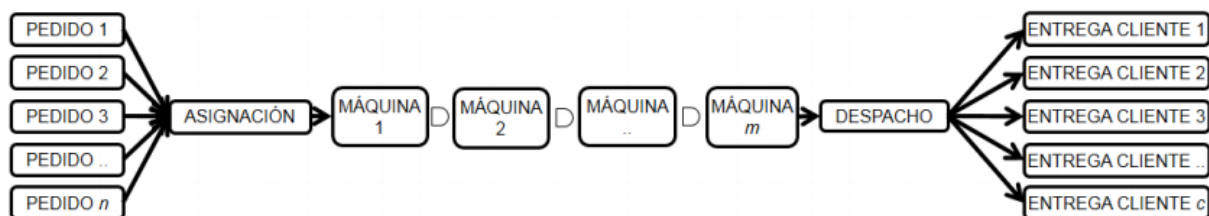


Ilustración 1.3. Sistema productivo flowshop con m máquinas

Fuente: (Ruiz, 2015)

1.2.4 Flexible Flow Shop

Es una variante del Flow Shop, algunos puestos de trabajo pueden tener máquinas en paralelo, para este caso un pedido pasa por una de las máquinas. En ciertos escenarios un pedido podrá omitir una o varias operaciones del sistema (Pinedo, 2012; Allahverdi et al., 2008). En la Ilustración 1.4 se ilustra el sistema productivo para m máquinas, n pedidos y c clientes; cada estación tiene 1 máquina excepto la estación 3, la cual tiene 2 máquinas en paralelo. El pedido 1 pasa por todas las estaciones, en la estación 3 el pedido 1 solo se procesa en la máquina 3.1, el pedido 2 pasa por las máquinas 1, 2, 3.2 y m . El pedido n solo se procesa en las máquinas 1, 2 y m . Igual que en la anterior configuración, el tiempo de procesamiento dependerá de la asignación de los pedidos, se generan demoras o colas entre máquinas y un pedido no será procesado en la máquina hasta que ella este libre. (Ruiz, 2015)

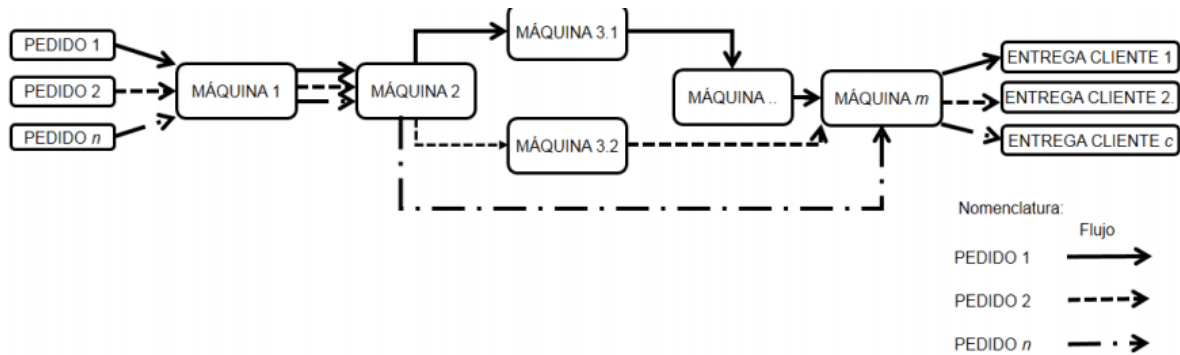


Ilustración 1.4- Sistema productivo Flexible flowshop con m máquinas

Fuente: (Ruiz, 2015)

1.2.5 Job shop

Según Domínguez Machuca et al. (1995) los sistemas Job Shop son sistemas de manufactura que, en algunos casos, tienen aplicación en empresas de servicios donde se elaboran productos de diferentes referencias, en lotes que van desde una a pocas unidades. Dichos lotes de producción tienen diferente secuencia de fabricación por centro de trabajo y/o máquina. Los lotes de producción son de cantidad variable en un solo producto, pueden diferir en términos de tiempo de procesamiento y requerimientos de material, y generalmente estas empresas u organizaciones suelen producir bajo pedido (make to order) (Ruiz, 2015).

Moraglio et al. (2005) y Kammer et al. (2011) presentan grafos de la secuencia de pedidos en la estructura Job Shop. En la ilustración 1.5. se muestra el sistema productivo para m máquinas y n pedidos: el pedido 1 se procesa en las máquinas 1, 2 y m ; el pedido 2 en las máquinas 2, m y 1; el pedido n en las máquinas m , 1 y 2. Igual que en la anterior configuración el tiempo de procesamiento dependerá de la asignación de los pedidos, se generan demoras o colas entre máquinas, un pedido no será procesado en la máquina hasta que ella esté libre. (Ruiz, 2015)

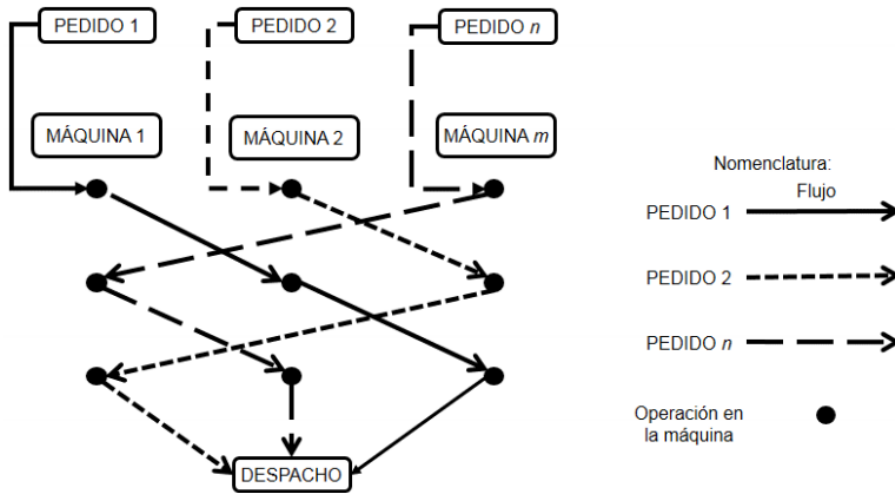


Ilustración 1.5 Sistema productivo tipo Job shop con m máquinas

Fuente: (Ruiz, 2015)

1.2.6 Open Shop

Cada pedido debe ser procesado en alguna o todas las máquinas y no existe una secuencia de operación estandarizada para cada trabajo. (Allahverdi et al., 2008), (Panahi y Tavkkoli, 2011), (Liaw, 2013), (Kononov, 2015), (Gupta y Singh, 2015), (Koulamas y Kyparisis, 2015) abordaron esta configuración, afirman que, a diferencia del flowshop, las operaciones de cada puesto de trabajo se pueden procesar en cualquier orden. El concepto que prevalece es el azar, la secuencia es arbitraria y la define el administrador antes de hacer la programación de la producción; esta labor puede ser asumida por un software que bajo condiciones de probabilidad elige aleatoriamente la secuenciación de los pedidos (Ruiz, 2015).

1.2.7 Flexible Job shop

Es la variante más completa del Job Shop, pues combina todas las configuraciones anteriores excepto el Open Shop. En este sistema, un pedido puede tener una o varias operaciones, con similar o diferente secuencia; algunos trabajos pueden repetir máquina, las máquinas pueden estar en paralelo y una operación se puede hacer en varias máquinas. (Roshanaei, 2012) presenta esta configuración con dos variaciones: la primera es el Parcial Flexible Job Shop, donde una operación no se puede realizar en todas las máquinas; la segunda es el Total Flexible Job Shop, donde una operación se puede realizar en todas las máquinas. Desde otra perspectiva (Frutos et al, 2010), (Medina et al., 2011) y (Zhang et al., 2012) plantean esta configuración, la cual es considerada la más compleja en cómputo y tiempo de procesamiento. Por su parte (Huang et al.,2014) afirma que es la configuración más aplicada en casos reales (Ruiz, 2015).

1.2.8 Supply Chain

Se denomina cadena de suministros, es la configuración más general, por que contempla todas las configuraciones anteriores, sin embargo, no es la más utilizada dentro de un taller

de trabajo. Establece una red entre sistemas con estructura Flexible Job Shop o Flexible Flow Shop. La planificación de la cadena de suministros se centra en la producción, transporte y almacenamiento entre instalaciones ubicadas en una red. (Pinedo, 2005) explica esta la importancia de la planeación y la programación de cada red en la cadena de suministros. Por su parte (Kazanas y Lerwick, 2003) mediante la descripción de proceso de moldeo, fundición y talleres de trabajo presentan la relación entre proveedores de la red productiva (Ruiz, 2015).

1.3 RESTRICCIONES EN EL SISTEMA PRODUCTIVO

A continuación, se describen las restricciones más utilizadas en las configuraciones productivas (Pinedo, 2012).

1.3.1 Interrupciones (No preemptions):

Se refiere a que en el momento que un trabajo es iniciado en una máquina, debe ser terminado completamente. No se permite la interrupción de la operación para adelantar otro trabajo.

1.3.2 Precedencias:

Las restricciones de precedencia pueden aparecer en aquellas configuraciones donde el comienzo de una operación o tarea depende de la ejecución exitosa de unas tareas previas.

1.3.3 Tiempos de preparación secuencialmente dependientes:

Se refiere a que los tiempos de limpieza, ajuste y puesta a punto de una máquina o conjunto de máquinas son diferentes dependiendo del trabajo que estaba montado previamente en la máquina y aquel que va a ser montado para su procesamiento. Este tipo de restricción se presenta frecuentemente en la elaboración de pinturas.

1.3.4 Familia de trabajos:

Se refiere a un conjunto de trabajos que comparten operaciones y rutas similares de trabajo, aunque los tiempos de operación pueden ser diferentes entre cada trabajo de la familia.

1.3.5 Procesamiento por Lotes:

Una máquina puede procesar un conjunto de trabajos de manera simultánea. En este caso, los tiempos de operación de cada trabajo en el lote no necesariamente es el mismo, por lo que el tiempo de procesamiento del lote será definido por el tiempo individual más largo de los trabajos dentro del lote.

1.3.6 Ruptura (Breakdowns):

Una máquina puede no estar disponible todo el tiempo, esto puede deberse a programación de mantenimientos.

1.3.7 Permutación:

En configuraciones flowshop se refiere a que la secuencia de trabajos en la que son procesados los trabajos en la primera máquina, debe ser la misma para todas las otras máquinas; es decir se cumple la política FIFO (First in, first out).

1.3.8 Bloqueo:

Esta restricción está normalmente relacionada con configuraciones flowshop. En este caso, un trabajo que es completado en una máquina no podrá ser entregado a la siguiente máquina hasta que esta se encuentre libre o disponible, generando así el bloqueo de la máquina previa.

1.3.9 Sin Esperas (No wait):

Los trabajos no pueden esperar para ser procesados entre dos máquinas sucesivas. Bajo esta restricción, el tiempo de inicio de producción en la primera máquina debe garantizar que no se presenten esperas entre las máquinas siguientes.

1.3.10 Recirculación:

Se refiere al hecho de que un trabajo puede visitar una máquina o centro de trabajo en más de una ocasión

1.4 FUNCIONES OBJETIVO.

A partir de la búsqueda en la literatura se pueden listar las siguientes funciones objetivo:

- Con relación a los tiempos de terminación de los trabajos
 - ❖ Minimizar el máximo tiempo de terminación
 - ❖ Minimizar el tiempo de terminación total
 - ❖ Minimizar el tiempo de terminación ponderado

- Con relación con el flujo: Flujo se define como el tiempo que permanece un trabajo en el sistema
 - ❖ Minimizar el máximo flujo
 - ❖ Minimizar el flujo total
 - ❖ Minimizar el flujo total ponderado

- Con relación con la fecha prevista de entrega: En esta sección es importante definir el concepto de demora que se refiere a la diferencia entre el tiempo de terminación de un trabajo y su fecha prevista de entrega. Mientras que la tardanza se refiere a la parte positiva de la demora
 - ❖ Minimizar la máxima demora
 - ❖ Minimizar la demora total
 - ❖ Minimizar la demora total ponderada
 - ❖ Minimizar la máxima tardanza
 - ❖ Minimizar la tardanza total
 - ❖ Minimizar la tardanza total ponderada.
 - ❖ Minimizar el número de trabajos tardíos

Para una descripción detallada de las distintas funciones objetivo se remite el lector al texto de (Pinedo, 2012).

1.5 METAHEURÍSTICAS APLICADAS EN LA GESTIÓN DE PRODUCCIÓN.

Los problemas de programación de producción son en su mayoría NP-Complejos y requieren para su solución, procedimientos complejos y costosos en el tiempo (Tavares, 2014). En casos como este, las técnicas de optimización combinatoria surgen como alternativa interesante, pues tienen la capacidad de encontrar soluciones de muy buena calidad, en tiempos de cómputo bastante razonables (Gallego, 2006), (López, 2013) .

Los métodos de resolución existentes en la literatura pueden agruparse en dos grandes familias, estos son: los métodos exactos (como el modelo de programación lineal entera) y los métodos de aproximación (como las heurísticas y meta-heurísticas) (Yalaoui, 2012). La complejidad del modelamiento y la complejidad computacional que implica el problema de programación de la configuración flowshop, conlleva a que el problema de optimización combinatoria resulte muy difícil de resolver (Zhang, 2013), (López, 2013).

1.5.1 Métodos Constructivos

Son utilizados para definir soluciones sobre aquellos problemas de los cuales se desconoce cualquier aproximación inicial. Para este efecto, se establecen las reglas que permitirán obtener las mencionadas soluciones (Schrimpf et al, 1999).

En las siguientes secciones se explican métodos que, a diferencia de los constructivos parten de una o varias soluciones iniciales para realizar la búsqueda.

1.5.2 Búsqueda local iterada.

Permite rastrear la solución a un problema de optimización combinatoria. Se diseña, por tanto, un algoritmo de búsqueda que una vez identificado un óptimo local, trata de buscar otra solución óptima perturbando la solución actual sin aleatorizar (Martin et al, 2003)

1.5.3 Recocido Simulado:

En la literatura se encuentra con los nombres de “simulación de templado”, “recocido simulado” y “Simulated Annealing” (SA). Planteado por (Kirkpatrick et al.,1983) y (Kirkpatrick, 1984) es una técnica empleada para solucionar diversos problemas complejos de optimización, con variables continuas o discretas, como son el diseño de bases de datos, filtros digitales, multiprocesador, electrostáticos y análisis de imágenes. SA ejecuta una búsqueda haciendo uso de la analogía con el proceso físico de templado de metales. (Toro et al., 2008) y (Loranca et al., 2009) presentan varias aplicaciones del SA (Ruiz, 2015).

1.5.4 Búsqueda de entorno variable (Variable Neighbourhood Search, VNS)

La idea fundamental de VNS, se orienta a cambiar sistemáticamente de estructura de entornos en una búsqueda local, lo cual mejora la detección de soluciones satisfactorias. (Hansen et al, 2003)

1.5.5 Búsqueda Tabú

La búsqueda tabú (TS) es un método meta-heurístico propuesto por Glover en la década de los 80 (Saricicek, 2011). El algoritmo TS se considera una técnica de búsqueda local, ya que realiza una exploración a través de toda la vecindad estudiando adecuadamente los óptimos locales (Gallego, 2006). El procedimiento de la búsqueda tabú, inicia con una solución básica inicial y se busca en su vecindario una solución que presente mejor rendimiento. Cuando lo hace, la búsqueda se mueve hacia la mejor y repite el proceso hasta que alguna condición de parada se satisfaga. La clave está, en evitar quedar atrapado en un óptimo local mediante la introducción de un mecanismo de la lista tabú,

que define algunos movimientos prohibidos para aplicase en un momento dado (Jin, 2011), (López, 2013).

1.5.6 Grasp (Greedy Randomized Adaptive Search Procedure).

Esta metodología en cada iteración recorre dos fases, la primera de construcción y la siguiente de búsqueda local. En la fase de construcción, se obtiene una solución factible mientras que, en la segunda, se detecta un posible valor óptimo. El valor óptimo se guarda. Posteriormente, se continúan recorriendo las fases nombradas durante varias iteraciones. Se selecciona entonces como solución, el mejor de los valores óptimos que se guardaron (Rabello et al, 2008).

1.5.7 Colonia de hormigas

En la literatura se encuentra con el nombre de “ant colony optimization” (ACO), desarrollada por (Dorigo et al., 1996). Consiste en simular el comportamiento de la colonia cuando se traslada desde el hormiguero hasta el alimento; las hormigas se comunican entre ellas para optimizar el trayecto y reducir el camino. La simulación sirve para solucionar problemas de transporte y logística. (García et al., 2009) desarrollan un algoritmo basado en ACO para solucionar problemas en un contexto militar (Ruiz, 2015).

1.5.8 Algoritmos Genéticos

Los algoritmos genéticos (GA), desarrollados en la década de los 70 por Holland (Gómez, 2007), (Mahdavi, 2008), son técnicas de búsqueda heurística que toman la analogía de los conceptos de la selección natural, empleando una población de soluciones candidatas y combinándolas en formas específicas con el fin de obtener mejores soluciones (Feng,

2009); estos algoritmos se han convertido en una metodología muy popular para solucionar una gran variedad de problemas complejos (Bertel, 2004),(Chen, 2012). El algoritmo genético, siendo una técnica de búsqueda estocástica, se ha logrado aplicar a varias áreas, incluyéndose los problemas de programación de máquinas (Hsu, 2009), (López, 2013).

1.6 Algoritmos Multiobjetivo

1.6.1 Algoritmo VEGA.

Con el propósito de aprovechar la potencialidad de los algoritmos genéticos para realizar optimización multiobjetivo, en el año de 1985, se realiza una propuesta en este sentido. La propuesta consiste en readaptar la aptitud del vector valorado (término que se refiere al conjunto de objetivos que se desean alcanzar), lo cual recibió el nombre de algoritmo genético para evaluación del vector (Vector evaluated genetic algorithm: VEGA). Se modifica, por tanto, el paso correspondiente a la selección. De esta forma, en cada generación se obtienen cierto número de subpoblaciones, teniendo como criterio el desempeño de la función objetivo evaluada en su momento. Con base en lo expuesto, para “m” objetivos, “m” generaciones con tamaño N / m serían generadas, asumiendo que la población es de tamaño “N”. Posteriormente, las subpoblaciones se reúnen, con el propósito de continuar con las funciones usuales de cruzamiento y mutación (Toscano, 2001), (Coca, 2014).

1.6.2 Algoritmo MOGA.

Inicialmente en la programación multiobjetivo, se lograba convergencia hacia cierto tipo de soluciones que reciben el nombre de frontera de Pareto. El riesgo de este enfoque consiste en su poca capacidad de evitar que un óptimo local sea identificado como solución. Es así como en 1993, se diseña la aplicación denominada: algoritmo genético multiobjetivo (Multiobjective genetic algorithm: MOGA), la cual permite, por medio de la modificación de la función de adaptación, ampliar la búsqueda hacia áreas de la frontera de Pareto con

pocos puntos solución. De esta forma, se extiende el espacio de búsqueda, evitando los óptimos locales (Baesler et al, 2006), (Coca, 2014).

1.6.3 SPEA

Los investigadores Zitzler y Thiele (1999), crean un algoritmo llamado *Strength Pareto Evolutionary Algorithm* (SPEA) el cual utiliza un archivo externo para retener las soluciones no dominadas globales previamente encontradas; en cada generación se copia en este archivo las soluciones no dominadas y se borran las que pasan a ser dominadas (Cuartas, 2009)

Si el número de individuos no dominados en el archivo externo, supera el número dado permitido N' entonces, se reducen los individuos mediante *clúster*. SPEA lo hace con el método de vinculación promedio (*The average linkage method*). El cual consiste en calcular la distancia promedio entre las soluciones y formas clústers entre las soluciones más cercanas, seleccionando una de las que pertenece al clúster, normalmente se utiliza el centroide para esta selección. Posteriormente se asigna una función de aptitud a cada solución de acuerdo a la fuerza o dominancia que presente con los otros individuos de la población. Finalmente, la reproducción se hace bajo torneo binario en función de esa aptitud y los operadores de cruce y mutación funcionan de la misma manera que un Algoritmo genético convencional.

1.6.4 NSGA

Los investigadores (Deb y Srinivas, 1994) crean el *Nondominated Sorting Genetic Algorithm* (NSGA). Estos toman la idea propuesta por Goldberg, que consiste en clasificar los individuos por frentes dependiendo de la dominancia (Cuartas, 2009).

Este se diferencia de un algoritmo genético convencional por la manera de trabajar con el operador de selección. Para determinar cuáles individuos son seleccionados se busca en la población actual los individuos no dominados y se clasifican en el primer frente de dominancia con la misma función de aptitud

Este procedimiento se repite hasta que se clasifican a todos los individuos de la población en diferentes frentes de dominancia. Para conservar la diversidad se utiliza un operador de distribución que trata de seleccionar aquellas soluciones que se encuentren dispersas dentro del frente de dominancia. La selección se hace con base en la función de aptitud que es proporcional al frente de dominancia al que pertenece el individuo.

Este algoritmo no fue muy eficiente porque la frontera de Pareto debe ser reparada muchas veces, por lo tanto resultó ser un algoritmo muy lento, poco efectivo e implicaba altos costos en su implementación (Coello, 2002), (Cuartas, 2009).

1.6.5 SPEA2

En el año 2001 Zitzler, Laumanns y Thiele hacen unas correcciones al algoritmo SPEA, al que llaman *SPEA2* (Zitzler et al., 2001), el cual se diferencia del primero por la asignación de la aptitud, la selección de los padres, el operador de truncamiento y en fijar el tamaño del archivo externo para todas sus generaciones (Cuartas, 2009).

El operador de selección de individuos considera tanto a los individuos dominados como a los no dominados, así como un indicador de densidad, que se refiere a la cantidad de soluciones cercanas; y a partir de estos genera la función de aptitud de cada individuo. La selección de los padres sólo se ejecuta con los individuos seleccionados y ubicados en el archivo externo.

1.6.6 NSGA-II

En el año 2000 los investigadores Deb, Pratap, Agarwal y Meyarivan desarrollan un algoritmo llamado *Nondominated Sorting Genetic Algorithm-II* (NSGA-II) descrito en Deb et al. (2002) el cual usa un procedimiento rápido para organizar la población por no dominancia, un enfoque para preservar el elitismo y un operador, diferente al nicho, para dispersar los individuos en la frontera de Pareto (Cuartas, 2009).

Los individuos son clasificados por frentes de dominancia y su aptitud es definida de manera proporcional al frente de dominancia. A partir de esta aptitud, y por medio de torneo

binario se seleccionan los padres y se generan hijos del mismo tamaño de la población de los padres. Finalmente, se seleccionan del grupo conformado por padres e hijos un tamaño de población indicado de acuerdo con su aptitud. En caso de que se obtenga un frente de dominancia dado con más individuos que los que se pueden seleccionar para la nueva población, se utiliza un operador de apilamiento que determina la distancia existente entre las soluciones del último frente de dominancia.

1.6.7 NSGA-III

Cuando se desea trabajar con problemas de optimización con más de 3 objetivos con los algoritmos desarrollados en problemas bi o tri-objetivo, los operadores de conservación de la diversidad generan un costo computacional muy alto y por tanto un tiempo de respuesta considerable a la hora de entregar un conjunto solución. Por otra parte, en múltiples dimensiones la frontera de Pareto puede ser muy extensa, por lo que un conjunto muy grande de soluciones pueden ser no dominadas en el espacio de búsqueda, lo que finalmente implica complejidad adicional en el decisor a la hora de seleccionar el conjunto solución más adecuado. Es así como Deb & Jain describen un algoritmo llamado “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach”, En el cual proponen un algoritmo basado en NSGA II adaptado para la solución de problemas de optimización con más de 3 objetivos. La diferencia principal se encuentra en la forma de seleccionar los individuos del último frente de dominancia que es elegido para crear la nueva población; pues en este caso el operador de apilamiento es reemplazado por un operador basado en punto de referencia suministrados bien sea por el decisor o conocedor de las características del problema o bien generados por los resultados de búsqueda del mismo algoritmo.

Así, El algoritmo propuesto inicia generando una Población de padres P y una población de hijos Q; los cuales se clasifican en frentes no dominados de acuerdo a su función de aptitud multiobjetivo. Luego de esto, se seleccionan los individuos desde el primer frente de dominancia hasta aquel que complete el tamaño de la nueva población. En caso que, al agregar los individuos de un frente de dominancia dado, se sobrepase al tamaño de la población establecida, se debe aplicar el operador basado en puntos de referencia para seleccionar a aquellos individuos que generen mayor diversidad en el conjunto de soluciones. (Deb & Jain, 2014).

1.7 Métricas de Desempeño

Existen diversos tipos de clasificaciones para categorizar las métricas de desempeño multiobjetivo. Un criterio de clasificación considera las siguientes categorías:

- **Métricas de Cardinalidad:** La cardinalidad de un conjunto A se refiere al número de soluciones que se encuentran en A . Intuitivamente, un número mayor de cardinalidad es mejor.
- **Métricas de Exactitud:** Este aspecto se refiere directamente a la convergencia de las soluciones en el conjunto A . Indica la distancia entre las soluciones encontradas y la frontera de Pareto real del problema. En caso de no conocerse la frontera real, se utilizan valores de referencia para cada objetivo.
- **Métricas de Diversidad:** Mide la distribución y la extensión de las soluciones pertenecientes al conjunto A ; donde la primera se refiere a la distancia entre las soluciones en el conjunto A ; mientras que la segunda se asocia al rango de valores cubiertos por las soluciones en el conjunto A .

Existen algunas métricas utilizadas en el campo científico que pueden clasificarse en una o varias categorías de las mencionadas anteriormente. En la tabla 1.1 se muestra un ranking de las métricas más utilizadas en las publicaciones de investigaciones con problemas multiobjetivos de acuerdo con el informe presentado en la conferencia de computación de América Latina en 2015

Ranking	Citaciones	Métrica	Características
1	91	Hipervolumen	Exactitud y Diversidad
2	26	Distancia Generacional	Exactitud
3	23	Familia Epsilon	Exactitud, Diversidad y Cardinalidad
4	17	Distancia Generacional Inversa	Diversidad y exactitud

Tabla 1-1. Ranking de métricas más citadas para problemas multiobjetivo

Fuente: (Riquelme, Et Al, 2015)

Así, la métrica de desempeño más utilizada por los investigadores es el hipervolumen, también conocido como métrica S, o Hiper Área, entre otros; El hipervolumen mide la región del espacio factible que es dominado o cubierto por un conjunto de soluciones dado; para lo cual se requiere establecer un punto de referencia que puede ser el origen o bien algunos valores estimados por el decisor.

El Hipervolumen es la única métrica no binaria que considera los tres aspectos citados anteriormente: Cardinalidad, exactitud y diversidad. Razón por la cual a partir del Hipervolumen se puede saber si un conjunto de soluciones domina completamente a otro, pues esto indicará que el hipervolumen del primer conjunto será mayor que el asociado al segundo conjunto. Por estas características, esta es la métrica seleccionada para valorar los conjuntos de soluciones entregadas por los algoritmos propuestos.

1.8 ANTECEDENTES

El primer artículo que se refiere a los problemas de programación flowshop fue publicado hace poco más de 50 años (Gupta Et Al, 2006). Desde entonces, diferentes autores se han enfocado en diferentes aspectos de este problema. (Bellman, Et al; 1954) y (Bellman Et al, 1982) presentan el problema clásico de flowshop con dos máquinas. (Carlier Et Al, 1996) y (Ronconi, 2005) trabajan el problema con múltiples máquinas. (Lyer Et al, 2004) y (Siarry, 2007) han propuesto algunos heurísticos y metaheurísticos para problemas de gran tamaño en este tipo de configuraciones flowshop (Trabelsi Et al, 2012).

La mayoría de la literatura para el problema de producción flowshop está concentrado en un criterio de optimización individual (Pan et al, 2002), (Fink and Vob, 2003), (Bulfin and M'Hallah,2003); (Blazewicz et al.,2005), (Choi et al.2007), (Grabowski and Pempera, 2005) y (Wang et al., 2006) han hecho propuestas con objetivos como trabajos tardíos, máximo tiempo de terminación, makespan, entre otros. (Karimi Et al.,2014).

En la práctica, existen procesos de producción donde no existe un espacio intermedio entre máquinas debido a requerimientos técnicos o la característica del proceso. En consecuencia, un trabajo que ha sido procesado completamente en una máquina tendrá que permanecer en esta máquina hasta que la siguiente máquina se encuentre disponible para su procesamiento. Este modelo es conocido como el problema de programación de producción flowshop con bloqueo, el cual tiene importantes aplicaciones en diversos sistemas industriales. (Pan y Wang, 2012). (Grabowski y Pempera, 2000) describen un ejemplo aplicado en la fabricación de bloques de concreto; (Ronconi, 2004), (Merchan, 2016) describen otro ejemplo en la industria química donde se procesan los trabajos parcialmente. (Gong et al.2010) consideran un problema de programación flowshop con bloqueo encontrado en la industria del hierro y el metal. (Koren Et Al, 2017) describen la aplicación de este ambiente de trabajo en proceso de manufactura de productos en serie.

Por décadas, heurísticos y metaheurísticos han sido desarrollados para resolver el problema de programación flowshop con bloqueo. (Pan y Wang, 2012). (Pan et al. 2013), proponen un algoritmo memético para resolver este problema con el objetivo de minimizar el makespan. (Toumi et al. 2013) proponen un algoritmo de Ramal y límite para minimizar la tardanza total. (Han et al, 2012) han propuesto el algoritmo de colonia de hormigas para determinar soluciones que minimicen el tiempo total de flujo y (Han et Al, 2014) proponen el mismo metaheurístico para minimizar el makespan. (Jarboui et al. 2009) presenta un algoritmo de estimación de distribución con el fin de encontrar soluciones que minimicen el makespan. (Caraffa et al, 2001) usan algoritmos genéticos para minimizar el makespan y (Januario et al, 2009) usan algoritmos genéticos para minimizar la tardanza total. Por otra parte, (Duan Et Al, 2017) proponen un algoritmo memético probabilístico basado en EDA para solucionar un problema de programación en un entorno flowshop con bloqueo y con tiempos de preparación secuencialmente dependientes. Adicionalmente, (Han Et Al, 2016) muestran un MFFO (Modified Fruit Fly Optimization Algorithm) con el objetivo de minimizar el makespan en un ambiente de programación flowshop con bloqueo

Según (Karimi, 2014) considerar un solo objetivo es poco práctico y poco calificado para los problemas de la vida real. Así, muchos investigadores se han concentrado en considerar problemas con más de un objetivo a la vez. (Murata et al, 1996) propone un algoritmo genético multiobjetivo que utiliza un enfoque ponderado para resolver el

problema de programación flowshop multiobjetivo. Un algoritmo genético híbrido con enfriamiento simulado y búsqueda local es utilizado por este mismo autor para mejorar las soluciones entregadas por su trabajo anterior. (Murata et al, 1996-2)

(Toktas et al, 2004), proponen un procedimiento de ramal y límite y un heurístico para encontrar soluciones cercanas al óptimo de un problema de programación flowshop con dos máquinas y con los objetivos de minimizar el makespan y la máxima demora. (Arroyo y Armentano, 2005) proponen un algoritmo genético de búsqueda local con elitismo y el uso de una búsqueda local multiobjetivo paralela para intensificar la búsqueda en distintas regiones del problema de programación flowshop. (Rahimi-Vahed and Mirghorbani, 2007) desarrollan un algoritmo de nube de partículas para optimización multiobjetivo en problemas de programación flowshop. (Karimi, 2014)

Por otra parte, (Jaszkiewicz, 2002), (Ishibuchi et al.2003), (Reeves, 1995), y (Chen et al, 1995) son los pioneros en la presentación de trabajos que aplican algoritmos genéticos para resolver el problema de programación flowshop multiobjetivo. (Reeves and Yamada, 1998), (Wang and Zhang, 2003), (Wang et al., 2006), (Chang et al. 2007), proponen híbridos de otras técnicas con los algoritmos genéticos para mejorar el espacio de búsqueda de soluciones. Numerosos algoritmos multiobjetivos son propuestos como el que combina la población de un algoritmo genético cuando la diversidad de la población decrece (Affenzeller, 2001), (Chang, 2008).

Recientemente, viene aumentando el interés en la aplicación de técnicas de aprendizaje de máquinas en optimización combinatoria y estocástica. El uso de aprendizaje de máquinas en la optimización provee una mayor eficiencia en la exploración y representación de la información obtenida sobre el espacio de búsqueda. Un ejemplo consolidado de los métodos que incorporan aprendizaje de máquinas para optimización son los algoritmos de estimación de distribución (EDAs). (Bosman, 2010), (Hauschild et al, 2012) y (Larrañaga et al, 2012) muestran que una evidencia fuerte de su popularidad es el desarrollo de nuevos y más complejos EDAs, (Armañanzas et al, 2008), (Santana, 2008) y (Browniee et al, 2008), muestran su aplicación en problemas reales y académicos, mientras que (Zhang, 2004), (Shapiro, 2005) y (Echegoyen et al, 2011) se enfocan en el estudio de las preguntas fundamentales con el fin de comprender mejor cómo funciona el algoritmo.(Echegoyen et al, 2015). Por otra parte, (Zhou Et Al, 2016) utiliza un EDA para

minimizar el makespan en un ambiente de programación flowshop con dos máquinas de lote, no espera y tiempos de preparación secuencialmente dependiente para la segunda máquina de lote.

Elementos	Algoritmos						AEDS
	Chen <i>et al.</i> (2010a) EA/G	Chen <i>et al.</i> (2012b) GA - autoguiado	Chen <i>et al.</i> (2012c) eACGA	Jarboui <i>et al.</i> (2009) JEDA	Pan y Ruiz (2012)	Wang <i>et al.</i> (2012) BEDA	
Configuración	Máquina simple	PFSP	PFSP	PFSP	PFSP por lotes	FJSP	FJSP
Tiempos Proceso	Fijo	Fijo	Fijo	Fijo	Fijo	Fijo	Variable
Supuestos	Requeridos	Requeridos	Requeridos	Requeridos	Requeridos parcialmente	Requeridos	No Requeridos
Modelo Probabilístico	Univariado	Univariado	Uni/Bivariado	Univariado	Uni/Bivariado	Bivariado	Bivariado
Tipo de EDA	Discreto	Discreto	Discreto	Discreto	Discreto	Discreto	Discreto y Continuo
Hibridación	AED-AG	AED-AG	AED-AG	AED-VNS	EDA-VNS	EDA-Búsqueda Local	EDA
Población	Una	Una	Una	Una	Una	Dos	Una
Objetivo	Costo total promedio de tardanza	Makespan	Makespan	Tiempo total de flujo	Makespan	Makespan	Makespan
Enfoque	Académico	Académico	Académico	Académico	Industrial, secuencia dependiente del tiempo de preparación	Académico	Industrial, secuencia dependiente de las horas fuera de servicio

PFSP por sus siglas en inglés Permutation Flowshop Scheduling Problem
 FJSP por sus siglas en inglés Flexible Jobshop Scheduling Problem
 VNS por sus siglas en inglés Variable Neighborhood Search

Tabla 1-2 EDA para secuenciamiento.

Fuente: (Pérez, 2015)

En la tabla 1.2, se resumen y comparan los enfoques en los cuales se ha utilizado el algoritmo de estimación de distribución en los años recientes. En esta podemos observar que para todos los algoritmos se utiliza un solo criterio de optimización, se proponen diseños del algoritmo para variables discretas y con un enfoque principalmente académico; es decir orientados a la investigación científica más que a la aplicación industrial (Pérez et al, 2015).

De acuerdo con la revisión de la literatura, se puede llegar a las siguientes conclusiones:

- Se evidencia interés de investigación por los problemas de programación flowshop

- Aunque la mayor proporción de la investigación se ha enfocado en los problemas con un solo criterio de optimización, los autores citan la importancia de desarrollar problemas multicriterio que se acerquen a los problemas reales de la industria.
- La restricción de bloqueo entre máquinas en un problema de producción flowshop tiene aplicación tanto en la academia como en el mundo industrial
- Los algoritmos de estimación de distribución han tenido un buen desempeño en comparación con otras herramientas y técnicas de optimización.
- No se evidencia aplicación de algoritmos de estimación de distribución en problemas de programación flowshop con restricciones de bloqueo y en la cual se quieran alcanzar múltiples objetivos.

2. Capítulo 2. Metodología multiobjetivo para programar un sistema flowshop con bloqueo

2.1 Descripción del problema

En el problema $F_m/block/C_{max}-T$, se tiene un conjunto de n trabajos disponibles en el tiempo cero; los cuales deben ser procesados a través de m máquinas siguiendo el mismo patrón de flujo en ellas. No se tiene capacidad de almacenamiento intermedio entre máquinas sucesivas. Las operaciones no pueden ser interrumpidas una vez que se inicien. No se presentan tiempos de preparación secuencialmente dependientes entre los trabajos. Cada máquina puede procesar sólo un trabajo a la vez y cada trabajo sólo puede ser procesado por una máquina a la vez. Cada trabajo tiene un tiempo de procesamiento en cada máquina y una fecha prevista de entrega. El objetivo es encontrar un programa factible que minimice el máximo tiempo de terminación de los trabajos y que a la vez minimice la tardanza total.

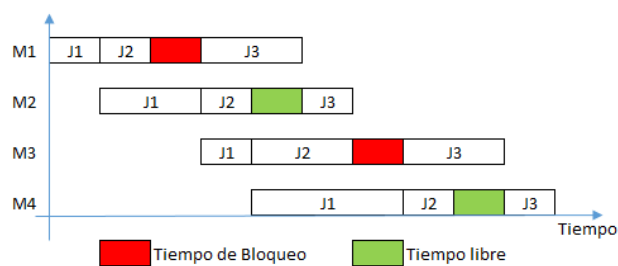


Ilustración 2.1. Una solución para 3 trabajos y 4 máquinas en BFSP.

Fuente: Elaboración propia

La ilustración 2.1 muestra una solución de un problema de programación en ambiente flowshop con bloqueo BFSP (Blocking Flowshop Scheduling Problem) con 3 trabajos y cuatro máquinas

2.1.1 Modelo matemático

A continuación, se presenta la formulación matemática propuesta por (Trabelsi, Sauver y Sauer, 2011). Con el fin de definir formalmente el problema presentado:

Parámetros:

n : número de Trabajos

m : número de máquinas

P_{ij} : tiempo de procesamiento del trabajo i en la máquina j

d_i : fecha prevista de entrega del trabajo i

Variables:

Skj : Tiempo de inicio de ejecución de un trabajo en la posición k en una secuencia G sobre una máquina j

Ckj : Tiempo de terminación de un trabajo en la posición k sobre la máquina j en una secuencia G

$Cmax$: Tiempo de terminación del trabajo ubicado en la última posición en la secuencia G sobre la última máquina m

Tk : Tardanza presentada por el trabajo ubicado en la posición k en la secuencia G .

Función objetivo:

$$\text{Min } \sum_{i=1}^n T_i, Cmax \quad (1)$$

Restricciones:

$$S_{kj} \geq S_{k,j-1} + \sum_{i=1}^n P_{ij} - 1 * G_{ik} \quad \forall k \in \{1,2 \dots n\}, \forall j \in \{2, \dots m\} \quad (2)$$

$$S_{kj} \geq S_{k-1,j+1} \quad \forall j \in \{1, \dots m - 2\} \quad (3)$$

$$S_{k,m-1} \geq S_{k-1,m} \quad \forall k \in \{2, \dots n\} \quad (4)$$

$$S_{k,m} \geq C_{k-1,m} \quad \forall k \in \{2, \dots n\} \quad (5)$$

$$C_{kj} \geq S_{k,j} + \sum_{i=1}^n P_{ij} * G_{ik} \quad \forall k \in \{1,2 \dots n\}, \forall j \in \{1, \dots m\} \quad (6)$$

$$\sum_{i=1}^n G_{ik} = 1, \forall k \in \{1,2 \dots n\} \quad (7)$$

$$\sum_{k=1}^n G_{ik} = 1, \forall i \in \{1,2 \dots n\} \quad (8)$$

$$G_{ik} \in \{0,1\} \quad \forall i \in \{1,2 \dots n\}, \forall k \in \{1,2 \dots n\}, \quad (9)$$

$$C_{kj} \geq 0, \quad \forall k \in \{1,2 \dots n\}, \quad \forall j \in \{1,2 \dots m\} \quad (10)$$

$$S_{kj} \geq 0, \quad \forall k \in \{1,2 \dots n\}, \quad \forall j \in \{1,2 \dots m\} \quad (11)$$

$$T_i \geq 0 \quad \forall i \in \{1,2 \dots n\} \quad (12)$$

Donde:

Ec. (1): Función objetivo: minimizar la tardanza total y el tiempo máximo de terminación de los trabajos

Ec. (2): Representa la restricción de precedencia entre operaciones sucesivas de un mismo trabajo.

Ec. (3): Esta ecuación modela las restricciones de bloqueo

Ec. (4): Este es un caso especial de la ecuación 3 la cual solo relaciona a la penúltima máquina

Ec. (5): Este es un caso especial de la ecuación 3, solo está relacionada con la última máquina, la cual será la única que no presente bloqueo

Ec. (6): Esta restricción calcula el tiempo de terminación de cada trabajo.

Ec. (7): Cada trabajo i es ubicado sólo en una posición k sobre la secuencia G

Ec. (8): A cada posición k sobre la secuencia G sólo es asignado un trabajo i

Ec. (9): G_{ik} en una variable binaria. Es igual a 1 si el trabajo i se encuentra en la posición k sobre la secuencia G , cero en otro caso.

Ec. (10): El tiempo de inicio de los trabajos no puede ser negativo

Ec. (11): El tiempo de terminación de los trabajos no puede ser negativo

Ec. (12): La tardanza de cada trabajo no puede ser negativa.

2.2 Descripción del Algoritmo de Estimación de Distribución

El algoritmo de estimación de distribución inicia generando una población de individuos que serán evaluados en la función bi-objetivo definida. En este punto, se organizan los individuos en orden descendente de frentes de dominancia y se seleccionan una fracción de los mejores. A partir de los individuos seleccionados, se genera una función de distribución de probabilidad de cada trabajo en cada posición dentro de una secuencia dada y se genera la nueva población de individuos a partir de esta estimación. El algoritmo continúa hasta que se cumpla un determinado criterio de parada. A continuación, se muestra el pseudocódigo del algoritmo propuesto:

2.2.1 Seudo código

Leer P_{ij} , m , n , d_i

For $i = 1$ a Pop

 Generar un individuo de manera aleatoria

 Evaluar C_{max} y Tardanza del individuo

Next i

Calcular los frentes de dominancia de cada individuo

For j=1 to Gen

 Almacenar el o los individuos del primer frente de dominancia

 Ordenar los individuos en orden descendente de frentes de dominancia

 Seleccionar los mejores P individuos de acuerdo a frentes de dominancia

 Estimar función de distribución de probabilidad individual y conjunta

 For i= 1 a Pop

 Generar un individuo a partir de la función de distribución de probabilidad

 Evaluar Cmax y Tardanza del individuo

 Next i

Calcular los frentes de dominancia de cada individuo

Next j

Reportar la o las mejores soluciones, Cmax y T

2.2.2 Representación de una solución.

Teniendo en cuenta que el algoritmo busca encontrar una secuencia de trabajos que minimice el Cmax y T; y que ambas dependen del orden de los trabajos dentro de una secuencia dada; una solución factible será representada por una permutación de trabajos, de tal manera que dicha permutación se replicará en el orden estricto de cada una de las máquinas del problema. Sólo los programas de permutación en una configuración flowshop con bloqueo son factibles (Pinedo, 2012), (Ribas Et Al, 2011). Por ejemplo, en el caso de tener un problema con 5 trabajos y m máquinas, una posible representación de una solución es la permutación 3-4-1-2-5; la cual indica que en la primera posición se encuentra el trabajo 3, en la segunda el trabajo 4, y así hasta llegar a la posición final.

2.2.3 Selección de individuos

Cada individuo generado inicialmente, es clasificado y ordenado a partir del número de individuos restantes que son dominados por él, de acuerdo con los valores obtenidos en las medidas de desempeño C_{max} y T

En la tabla 2-1 se muestra un ejemplo de dicha clasificación por frentes de dominancia para un conjunto de 5 individuos:

Individuo	C_{max}	T	Soluciones Dominadas
1	10	15	1
2	3	4	4
3	12	13	1
4	20	18	0
5	5	8	3

Tabla 2-1. Clasificación por frentes de dominancia

Fuente: Elaboración propia

Así pues, por ejemplo, el individuo 2 domina a los cuatro individuos restantes dado que sus valores de C_{max} y T son menores que en los demás individuos; mientras que el individuo 4 no domina a ningún otro individuo dado sus pobres valores en los objetivos definidos. En caso de presentarse empates en la medida de desempeño de uno de los objetivos entre varios individuos, la clasificación será determinada por el valor de la otra medida de desempeño. En caso que dos o más individuos presenten empate en ambos valores de los objetivos propuestos, la clasificación entre ellos se realiza de manera aleatoria.

Finalmente, se ordenan los individuos en orden descendente de soluciones dominadas y se selecciona los primeros P individuos dentro de ese ordenamiento para ser utilizados como muestra para estimar la función de distribución de probabilidad.

2.2.4 Estimación de la función de distribución de probabilidad.

A partir de los individuos seleccionados, se desarrolla una estimación de la función de distribución de probabilidad asociada para cada trabajo a programar para cada posición de la permutación de trabajos. Dentro de la literatura se encuentran algunos enfoques para esta estimación como funciones discretas, redes bayesianas, probabilidades condicionales; redes de Markov, entre otros (Larrañaga, 2000).

Para la estimación de la función de distribución de probabilidad se utilizará el modelo probabilístico con funciones discretas citado por (Pan and Ruiz, 2012); teniendo en cuenta que considera aspectos como: el método de definición de probabilidades se adapta a la representación de soluciones con permutaciones de trabajos, define la probabilidad individual de cada trabajo en cada posición dentro de una secuencia y la probabilidad de encontrar bloques de dos trabajos conjuntos en cualquier posición de la secuencia de trabajos. Por otra parte, como lo evidencian los autores, el uso de esta función de distribución generó buenos resultados al aplicarse en problemas de programación con configuración flowshop.

A partir de esta función de distribución de probabilidad, los parámetros que se deben calcular son los siguientes:

$$\xi_{ij} = \frac{P_{ij}}{\sum_{l \in \Omega(i)} P_{il}} \quad \text{Para } i = 1$$
$$\xi_{ij} = \frac{\frac{P_{ij}}{\sum_{l \in \Omega(i)} P_{il}} + \frac{\lambda'_{jj}}{\sum_{l \in \Omega(i)} \lambda'_{lj}}}{2} \quad \text{Para } i > 1$$

Donde:

ξ_{ij} : Probabilidad de ubicar el trabajo j en la posición i dentro de la secuencia

P_{ij} : Número de veces que se encuentra el trabajo j antes o en la posición i en los individuos seleccionados.

$\lambda_{j'j}$: número de veces que aparece el trabajo j inmediatamente después del trabajo j' en los individuos seleccionados

$\Omega(i)$: conjunto de trabajos disponibles para ser programados en la posición i

A continuación, se relaciona un ejemplo que ilustra el procedimiento establecido anteriormente. Suponga que en el proceso de selección se eligieron los siguientes individuos: {1-3-4-2}, {3-2-1-4}, {1-3-2-4}; a partir de los cuales se establecen los valores de P_{ij} y $\lambda_{j'j}$

$$P_{ij} \begin{vmatrix} 2 & 0 & 1 & 0 \\ 2 & 1 & 3 & 0 \\ 3 & 2 & 3 & 1 \\ 3 & 3 & 3 & 3 \end{vmatrix}$$

$$\lambda_{j'j} \begin{vmatrix} - & 0 & 2 & 1 \\ 1 & - & 0 & 1 \\ 0 & 1 & - & 1 \\ 0 & 1 & 0 & - \end{vmatrix}$$

De acuerdo con esto, se pueden calcular las probabilidades de asignar cada trabajo j a la posición 1 así

$$\xi_{11} = \frac{2}{2+1} = 0.67$$

$$\xi_{12} = \frac{0}{2+1} = 0$$

$$\xi_{13} = \frac{1}{2+1} = 0.33$$

$$\xi_{14} = \frac{0}{2+1} = 0$$

Suponga que al generar un número aleatorio se selecciona el trabajo 1 en la posición 1; entonces:

$$\Omega(2)=\{2,3,4\}$$

$$\xi_{22} = \frac{\frac{1}{1+3} + \frac{0}{2+1}}{2} = 0.125$$

$$\xi_{23} = \frac{\frac{3}{1+3} + \frac{2}{2+1}}{2} = 0.71$$

$$\xi_{24} = \frac{\frac{0}{1+3} + \frac{1}{2+1}}{2} = 0.165$$

Así, a partir de estas probabilidades y bajo la generación de un número aleatorio se selecciona aquel trabajo que se ubicará en la posición 2. Este método se repite hasta que se programe el último trabajo en la última posición dentro de la secuencia.

2.2.5 Generación de nuevos individuos y actualización de la población.

Para la generación de los nuevos individuos, se generan números aleatorios que se utilizarán para hacer la selección, dada la distribución probabilística anterior, del trabajo que será ubicado en la posición i a partir del mecanismo de ruleta.

2.3 Descripción del Algoritmo Genético Multiobjetivo

Basado en los principios de la evolución y la genética, el algoritmo genético comienza con la creación de una población inicial de individuos que representan soluciones factibles del

problema abordado, a través de un proceso evolutivo, cada población es sometida a operadores genéticos de selección, cruce y mutación con el objetivo de mejorar la medida de desempeño deseada. El proceso continúa hasta que un criterio de parada determinado sea alcanzado.

A continuación, se muestra el pseudocódigo del algoritmo propuesto.

2.3.1 Seudo código

Leer P_{ij} , m , n , d_i , Tasa de Cruce Cr , Tasa de Mutación Tm

For $i= 1$ a Pop

 Generar un individuo de manera aleatoria

 Evaluar C_{max} y Tardanza del individuo

Next i

Calcular los frentes de dominancia de cada individuo

For $j=1$ to Gen

 Almacenar el o los individuos del primer frente de dominancia

 Ordenar los individuos en orden descendente de frentes de dominancia

 Seleccionar los mejores P individuos de acuerdo a frentes de dominancia

 If $BestGen > BestH$ then

$BestH = BestGen$

 End if

 For $c=1$ to $Tc * Pop$

 Seleccionar aleatoriamente dos padres del subconjunto P

 Crear un nuevo individuo a partir del cruce de los padres seleccionados

 Aplicar procesos de reparación genética en caso que se requiera

 Evaluar C_{max} y Tardanza del individuo

Next c

For m= Tc* POP + 1 to Pop

 Generar aleatoriamente un nuevo individuo para completar la población

 Evaluar Cmax y Tardanza del individuo

Next m

 Calcular los frentes de dominancia de cada individuo

Next j

Reportar la o las mejores soluciones, Cmax y T

2.3.2 Representación del cromosoma

En el algoritmo genético propuesto, un cromosoma factible es representado a partir de una permutación de trabajos, de tal manera que la longitud del cromosoma dependerá únicamente del número de trabajos que considere el problema de producción que se desea resolver. Esta representación viene siendo la misma que se utiliza para el desarrollo del algoritmo de estimación de distribución.

2.3.3 Selección de Individuos

Al igual que EDA, los individuos de la población son ordenados y clasificados en orden descendente de número de individuos que son dominados por este dentro de la población; de tal manera que al final se seleccionan (Cuartas, 2009).

los P individuos que presentan mayor dominancia desde el punto de vista de Frontera de Pareto.

2.3.4 Operador de Cruce

A partir de los individuos seleccionados anteriormente, se eligen aleatoriamente dos de estos individuos, denominados padres, de los cuales se generan cuatro hijos a partir de dos puntos de cruce definidos. Para determinar si los elementos pertenecientes a cada punto de corte serán copiados del padre 1 o padre 2, se utiliza una matriz ortogonal que determina según el punto de corte a cuál de los dos padres pertenece esa parte del cromosoma. Así, según la ilustración 2.2. para el primer hijo, los elementos antes del primer corte pertenecerán al segundo padre (un cero está asociado al primer padre y un uno al segundo); los elementos entre el primer y segundo corte pertenecen al padre 1 y los restantes al padre 1. Se repite el mismo proceso para cada hijo generado.

$$A = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}$$

Tabla 2-2. Matriz Ortogonal.

Fuente: Elaboración propia

A continuación, se presenta un ejemplo sobre la generación de hijos a partir de los padres seleccionados:

Padre 1: **2 3 4 1 5 6 8 7**

Padre 2: **1 3 2 4 6 8 7 5**

Hijo 1: **2 3 4 4 6 8 8 7**

Hijo 2: 1 3 2 1 5 6 7 5

Hijo 3: 2 3 4 1 5 6 7 5

Hijo 4: 1 3 2 4 6 8 8 7

Fuente: Elaboración propia.

Así, el hijo 1 fue generado antes del primer punto de cruce, por los elementos del padre 1, los elementos entre el primer y segundo corte, por el padre 2 y los elementos finales por el padre 1.

2.3.5 Reparación de los hijos.

Como se presentó en el ejemplo anterior, es posible que los hijos generados no sean permutaciones factibles dada las repeticiones de algunos elementos dentro del cromosoma. En estos casos se activa un operador de reparación que consiste en ubicar en la posición donde se encuentra el trabajo repetido alguno de los trabajos que aún no han sido programados así:

Hijo 1 Reparado: 2 3 4 1 6 8 5 7

Hijo 2 Reparado: 1 3 2 4 5 6 7 8

Hijo 3 Reparado: 2 3 4 1 5 6 7 8

Hijo 4 Reparado: 1 3 2 4 6 8 5 7

Fuente: Elaboración propia

Finalmente, los hijos reparados son evaluados en las funciones de desempeño definidas y se elige como nuevo individuo de la población, a aquel individuo que presente un mejor frente de dominancia. En caso de empate, se selecciona uno de los individuos de manera aleatoria.

2.3.6 Operador de Mutación.

Con el fin de conservar la diversidad en la población; los individuos faltantes para completar el tamaño de la población, son generados de manera aleatoria.

3. Capítulo 3. Experimentación y Resultados

3.1 Diseño del plan experimental

Con la finalidad de revisar el rendimiento del algoritmo de estimación de distribución desarrollado, se plantea un diseño de experimentos que permita comparar el desempeño del EDA multiobjetivo con el algoritmo genético multiobjetivo propuesto; estos aplicados a un problema de secuenciación de trabajo en un ambiente flowshop con bloqueo.

Teniendo en cuenta que ambos algoritmos involucran tamaño de población inicial y número de generaciones dentro de su desarrollo, estos elementos son considerados factores dentro del diseño experimental y se definieron sus niveles como se muestra en la tabla 3.1:

Factor	Nivel Bajo	Nivel Alto
Población Inicial	50	150
Generaciones	30	60

Tabla 3-1. Factores y niveles diseño experimental

Fuente: Elaboración propia

Los valores de los niveles bajos de cada factor se asignaron con base en los trabajos presentados por (Zhang et al, 2014) y (Trabelsi et al, 2012); mientras que los niveles altos se definieron de acuerdo con el planteamiento descrito por (Montgomery, 2004) en la página 51 de este documento.

En el uso del algoritmo genético, se consideró una tasa de selección del 25% de los mejores individuos de la población, una tasa de cruce de 60% y una tasa de mutación del 1%; como lo recomienda De Jong en (Coello, 2004).

Adicionalmente, estos factores fueron probados con diferentes cantidades de órdenes de trabajos por programar y número de máquinas en el flujo productivo. En ese sentido, se

manejaron problemas con 50 y 100 órdenes de trabajo; 20 y 40 máquinas, como lo emplean (Jarboui Et Al, 2009), (Karimi, 2014) en sus diseños experimentales.

Los tiempos de procesamiento de cada orden en cada máquina, se generaron de manera aleatoria con valores entre 1 y 99; mientras que las fechas previstas de entrega de la orden de trabajo se generó como un factor de la suma de los tiempos de procesamiento de cada orden en cada máquina, tal cual se propone en (Rego Et Al, 2012), (Zangari Et Al, 2017).

$$P_{ij} \sim U(1,99)$$

$$d_i = 1.5 \sum_{j=1}^m P_{ij} \quad \forall i$$

Donde i equivale al número de la orden de trabajo y j representa a la máquina.

En lo referente a la variable respuesta del diseño de experimentos, dado que se trata de un problema con múltiples objetivos, se define la métrica hipervolumen, la cual, dadas las ventajas descritas anteriormente, permite hacer una buena comparación de los resultados obtenidos por cada algoritmo en relación con la convergencia y la diversidad de los frentes de Pareto encontrados.

Esta métrica requiere de la definición de un punto de referencia en el espacio de soluciones para calcular el área cubierta por el conjunto de soluciones encontradas por cada algoritmo. Para efectos de esto, se calcularon los límites superiores de los objetivos elegidos de la siguiente manera:

$$Max Cmax: \sum_{i=1}^n \sum_{j=1}^m P_{ij}$$

Max T: Peor tardanza conocida

Donde el máximo C_{max} se obtiene considerando que un trabajo i sólo inicia a ser procesado en la primera máquina cuando el trabajo anterior ha pasado por todos los procesos en todas las máquinas. Lo que finalmente se representa como la suma de todos los tiempos de procesamiento de todos los trabajos en todas las máquinas.

Respecto a la máxima tardanza total, se toma como referencia el valor máximo de tardanza total conocida durante las corridas realizadas para cada problema planteado.

Se realizaron 20 corridas experimentales por cada combinación de factores, se calculó el hipervolumen generado por cada corrida y se define el promedio aritmético de estos, como la variable respuesta asignada a cada tratamiento del experimento.

Para el análisis estadístico se utiliza un diseño de experimentos 2^k con 3 factores a saber: Tamaño de la población inicial, número de generaciones y tipo de algoritmo multiobjetivo; y una sola réplica; esto considerando que aún para un diseño de experimentos con un número moderado de factores, el número total de combinaciones en un modelo 2^k es grande. Adicionalmente, para asegurar que se obtienen estimaciones confiables de los efectos de los factores para un diseño no replicado, se definieron los valores lo suficientemente distantes desde el punto de vista cuantitativo, de los niveles de cada factor (Montgomery, 2004)

Las corridas de los algoritmos se realizaron en un computador con procesador Intel Celeron de 2 Ghz y los datos se analizaron en el paquete estadístico Statgraphics Centurion XVI.

3.2 Resultados Obtenidos.

3.2.1 Experimentos 50 trabajos-20 máquinas

Análisis de Varianza para Hipervolumen

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
A:Población inicial	4.00403E15	1	4.00403E15	28.84	0.1172
B:Generaciones	2.26987E13	1	2.26987E13	0.16	0.7554
C:Algoritmo	3.59976E17	1	3.59976E17	2592.66	0.0125
AB	3.86268E14	1	3.86268E14	2.78	0.3438
AC	1.04778E15	1	1.04778E15	7.55	0.2223
BC	2.39856E15	1	2.39856E15	17.28	0.1503
Error total	1.38844E14	1	1.38844E14		
Total (corr.)	3.67975E17	7			

Tabla 3-2 Tabla ANOVA n=50, m=20

R-cuadrada = 99.9623 porciento

R-cuadrada (ajustada por g.l.) = 99.7359 porciento

Error estándar del est. = 1.17832E7

Error absoluto medio = 4.166E6

Estadístico Durbin-Watson = 1.5

Autocorrelación residual de Lag 1 = 0.125

De acuerdo con la tabla ANOVA, y con un nivel de significancia del 95% se evidencia un efecto de factor Algoritmo en la variabilidad del hipervolumen. Adicionalmente, no se evidencian efectos significativos de los factores Población inicial y generaciones en la respuesta media de hipervolumen. En relación con el R cuadrado ajustado, el 99.96% de la variabilidad en el hipervolumen es explicada por cambios en el algoritmo multiobjetivo ejecutado.

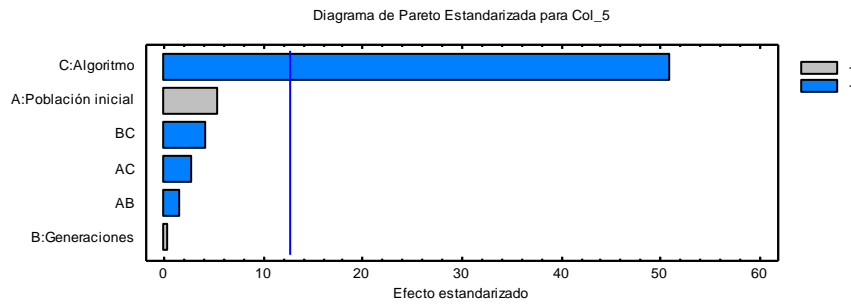


Ilustración 3.1 Diagrama de Pareto n=50, m=20

La ilustración 3.1 complementa el análisis previo realizado con la tabla Anova, en la cual se evidencia el efecto significativo del factor denominado Algoritmo, principalmente en su nivel inferior, que para el caso del diseño experimental está representado por el Algoritmo de Estimación de Distribución EDA.

Teniendo en cuenta que la tabla ANOVA evidenció que el modelo presentado es significativo en la respuesta media del hipervolumen, se presenta análisis gráfico de los efectos principales y sus interacciones, en la variable respuesta hipervolumen

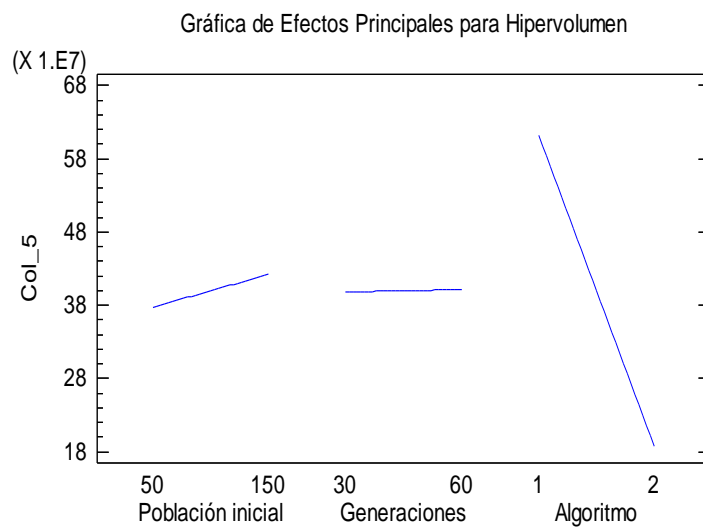


Ilustración 3.2 Gráfico de efectos principales.n=50, m=20

De acuerdo con la gráfica anterior, no se evidencia un cambio significativo en el hipervolumen con relación a los niveles de las generaciones establecidos. Por otra parte, se evidencia un incremento en el valor encontrado del hipervolumen a partir del aumento de la población inicial de individuos seleccionados. Finalmente, se observa un mejor rendimiento de la variable respuesta con el uso del algoritmo 1 (EDA) en comparación con el algoritmo 2 (AG)

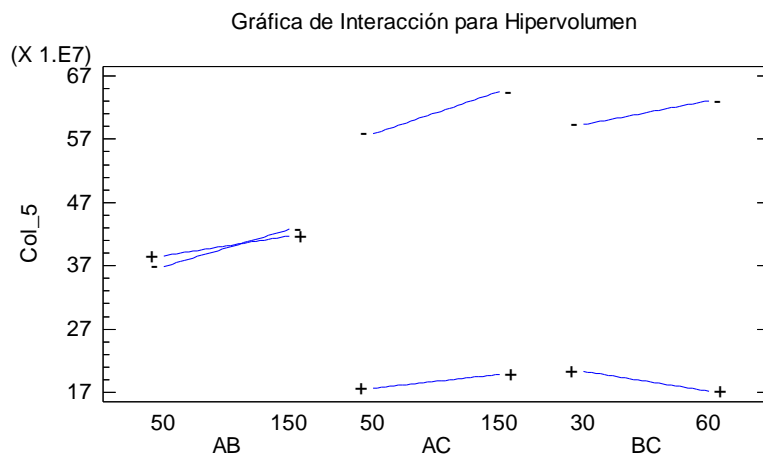


Ilustración 3.3. Gráficos de interacción entre factores $n=50$, $m=20$

El gráfico anterior muestra que existe una interacción entre la población inicial y las generaciones, de tal manera que una población inicial de 150 individuos y 30 generaciones, genera un mejor comportamiento de la variable respuesta que al realizar 60 generaciones, mientras que para una población de 50 individuos se presentan mejores resultados al interactuar con mayor número de generaciones. Por otra parte, no se evidencia efecto en la variable respuesta a partir de la interacción entre los algoritmos y los demás factores.

3.2.2 Experimentos con 50 trabajos y 40 máquinas

Análisis de Varianza para Col_5

<i>Fuente</i>	<i>Suma de Cuadrados</i>	<i>Gl</i>	<i>Cuadrado Medio</i>	<i>Razón-F</i>	<i>Valor-P</i>
A:Población inicial	7.46913E16	1	7.46913E16	289.07	0.0374
B:Generaciones	1.07977E15	1	1.07977E15	4.18	0.2896
C:Algoritmo	2.41511E18	1	2.41511E18	9346.88	0.0066
AB	7.23959E14	1	7.23959E14	2.80	0.3428
AC	6.69944E15	1	6.69944E15	25.93	0.1235
BC	1.3756E16	1	1.3756E16	53.24	0.0867
Error total	2.58387E14	1	2.58387E14		
Total (corr.)	2.51232E18	7			

Tabla 3-3. ANOVA para n=50, m=40

R-cuadrada = 99.9897 por ciento

R-cuadrada (ajustada por g.l.) = 99.928 por ciento

Error estándar del est. = 1.60744E7

Error absoluto medio = 5.68317E6

Estadístico Durbin-Watson = 3.5

De acuerdo con la tabla ANOVA, se evidencia que tanto la población inicial como el tipo de algoritmo ejecutado, tienen efecto significativo en la respuesta media del hipervolumen, con un nivel de confianza del 95%; dado que el valor p asociado a estos factores es menor que 0.05. Adicionalmente, se observa que el factor generaciones y las interacciones entre los factores no influyen en la variabilidad de la variable respuesta.

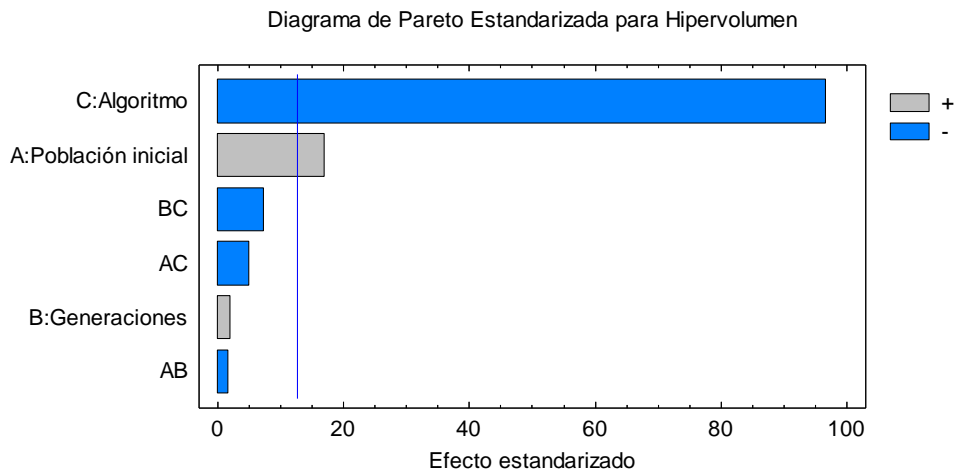


Ilustración 3.4. Diagrama de Pareto estandarizado n=50, m=40

De acuerdo con este gráfico, se confirma la información analizada previamente en la tabla ANOVA y se evidencia que el nivel inferior del factor algoritmo, es decir el EDA; y el nivel superior del factor población, en este caso 150 individuos, generan un aumento en el valor de la variable respuesta

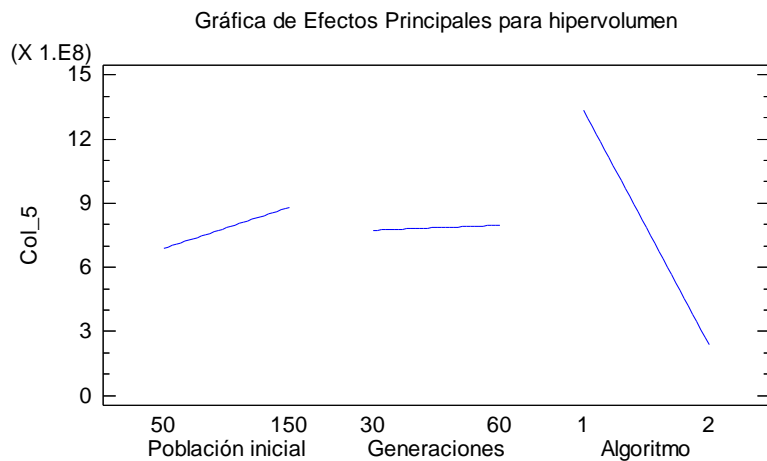


Ilustración 3.5. Gráfico de efectos principales n=50, m=40

Ahora, al hacer el análisis de los efectos de los factores principales para la variable respuesta, se evidencia un mejor comportamiento del hipervolumen cuando la población inicial de individuos es de 150 soluciones y utilizando el algoritmo de estimación de distribución EDA.

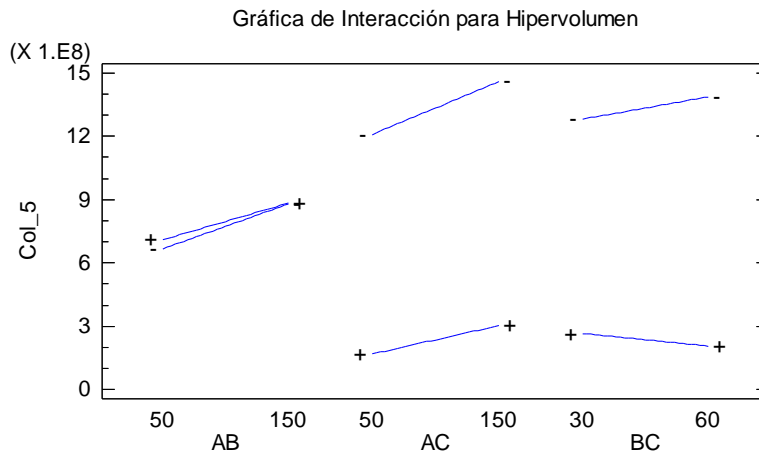


Ilustración 3.6. Gráfico de interacciones entre factores n=50, m=40

De acuerdo con el gráfico de efectos de la interacción de los factores en la respuesta media del hipervolumen, no se evidencian interacciones entre los factores. Sin embargo, se evidencia que el incremento en la población inicial interactuando con los algoritmos utilizados, mejora el desempeño en la respuesta media del hipervolumen.

3.2.3 Experimentos con 100 trabajos y 20 máquinas

Análisis de Varianza para Hipervolumen

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Población inicial	1.36057E17	1	1.36057E17	6.59	0.2365
B:Generaciones	2.85003E16	1	2.85003E16	1.38	0.4490

C:Algoritmo	9.14473E18	1	9.14473E18	442.77	0.0302
AB	2.10864E16	1	2.10864E16	1.02	0.4967
AC	1.08939E16	1	1.08939E16	0.53	0.6001
BC	5.26525E16	1	5.26525E16	2.55	0.3562
Error total	2.06535E16	1	2.06535E16		
Total (corr.)	9.41457E18	7			

Tabla 3-4. ANOVA para n=100 y m=20

R-cuadrada = 99.7806 por ciento

R-cuadrada (ajustada por g.l.) = 98.4644 por ciento

Error estándar del est. = 1.43713E8

Error absoluto medio = 5.08103E7

Estadístico Durbin-Watson = 1.5

De acuerdo con la tabla ANOVA y el diagrama de Pareto estandarizado, el tipo de algoritmo ejecutado en el experimento tiene efecto significativo en la respuesta media del hipervolumen, con un nivel de confianza del 95%; Mientras que los factores Población inicial, generaciones y las interacciones no presentan efecto significativo en la variable respuesta.

Adicionalmente, se observa que el 98.4 % de la variabilidad del hipervolumen es explicada por cambios en los niveles de los factores considerados.

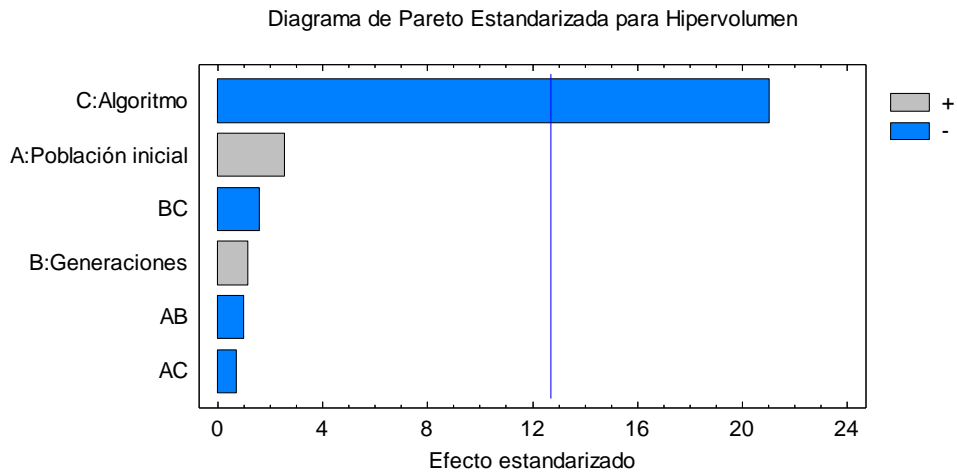


Ilustración 3.7. Diagrama de Pareto Estandarizada para n=100, m=20

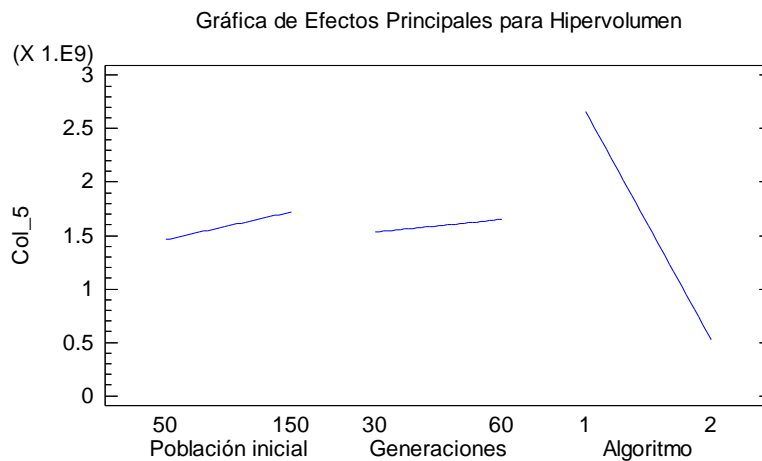


Ilustración 3.8 Gráfica de efectos principales para n=100, m=20

Con respecto al gráfico de efectos principales, se observa que el hipervolumen muestra un mejor desempeño con la ejecución del algoritmo 1 (EDA) en comparación con el algoritmo 2 (AG); Mientras que un incremento en la población inicial de individuos y las generaciones generan un mejoramiento en el hipervolumen, aunque no es significativo según la tabla ANOVA.

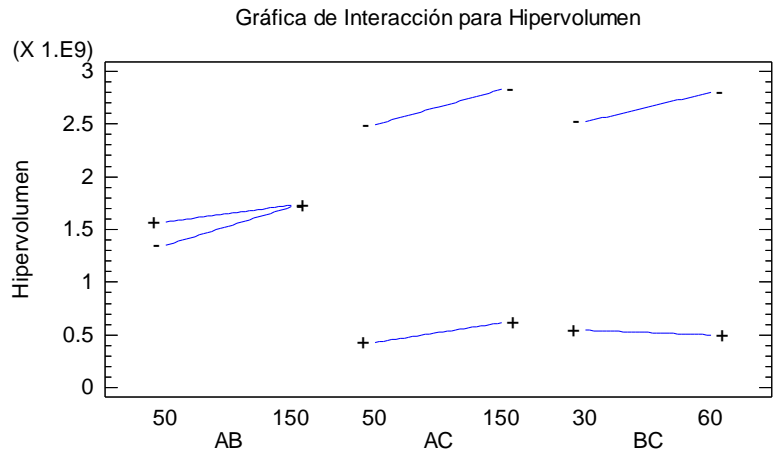


Ilustración 3.9. Gráfico de interacciones para n=100, m=20

Aunque no se evidencian interacciones significativas entre los factores principales del diseño, es importante destacar que la tasa de mejoramiento de la variable respuesta considerando la interacción entre población inicial y número de generaciones es mayor cuando se manejan menos generaciones, pero mayor número de individuos o soluciones iniciales.

3.2.4 Experimentos con 100 trabajos y 40 máquinas.

Análisis de Varianza para Hipervolumen

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
A:Población inicial	2.45813E17	1	2.45813E17	24.58	0.1267
B:Generaciones	1.33926E17	1	1.33926E17	13.39	0.1698
C:Algoritmo	5.05063E19	1	5.05063E19	5050.56	0.0090
AB	1.06862E16	1	1.06862E16	1.07	0.4894
AC	2.79146E16	1	2.79146E16	2.79	0.3434
BC	8.51908E16	1	8.51908E16	8.52	0.2101

Error total	1.00002E16	1	1.00002E16		
Total (corr.)	5.10199E19	7			

Tabla 3-5 ANOVA para n=100, m=40

R-cuadrada = 99.9804 por ciento

R-cuadrada (ajustada por g.l.) = 99.8628 por ciento

Error estándar del est. = 1.00001E8

Error absoluto medio = 3.53556E7

Estadístico Durbin-Watson = 1.0

De acuerdo con la tabla ANOVA anterior, el tipo de algoritmo presenta un efecto significativo en la respuesta media del hipervolumen, con un nivel de confianza del 95%; esto dado que el valor p asociado a este factor es menor a 0.05. Por otra parte, no se evidencia efecto significativo de los demás factores ni de sus interacciones.

En relación con el R cuadrado ajustado, el 99.86% de la variabilidad en la respuesta media es explicada por el comportamiento de los factores considerados en el diseño de experimentos.

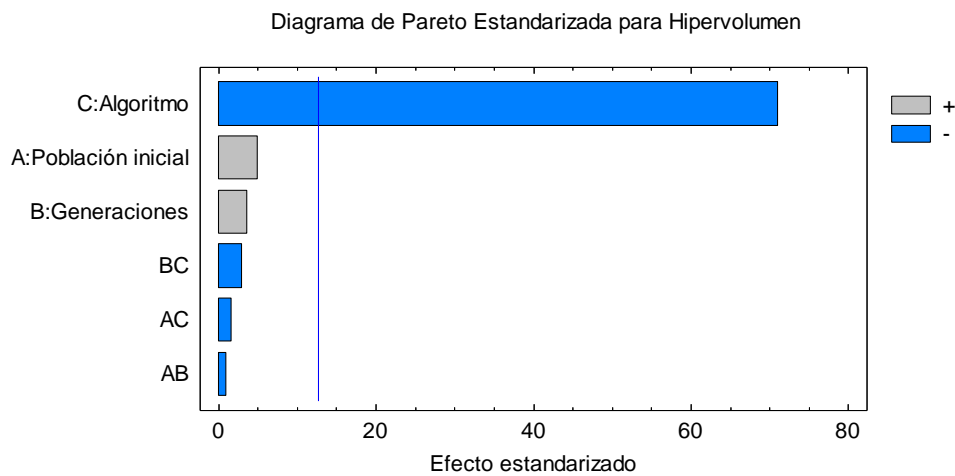


Ilustración 3.10 Diagrama de Pareto Estandarizado n=100, m=40

La gráfica de efectos principales muestra que se presenta un mejor comportamiento de la variable respuesta con el uso del algoritmo 1 (EDA) en comparación con el algoritmo 2 (AG). Por otra parte, el uso de un mayor número de individuos en la población inicial y de generaciones, incrementa el valor del hipervolumen, aunque este no es significativo para el modelo.

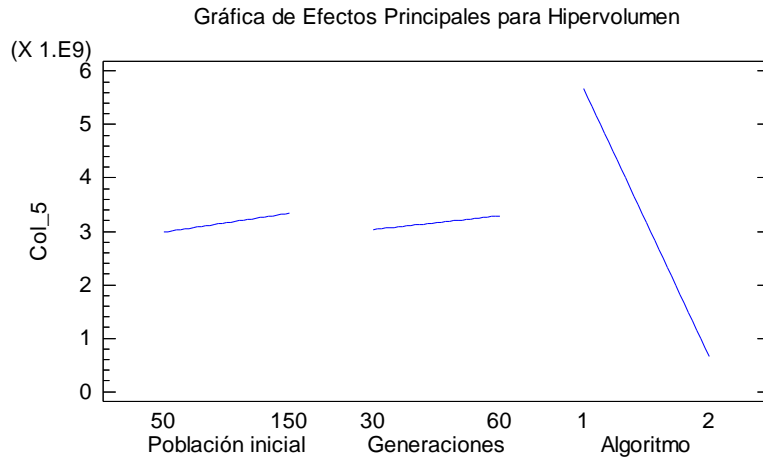


Ilustración 3.11. Gráfico de efectos principales n=100, m=40

Aunque no se evidencian aportes significativos de las interacciones entre los factores, en la respuesta media del hipervolumen, el incremento de la población inicial y el número de generaciones, incrementa levemente el valor de la variable respuesta, en mayor tasa cuando se utiliza el algoritmo 1 (EDA) en comparación con el algoritmo 2 (AG).

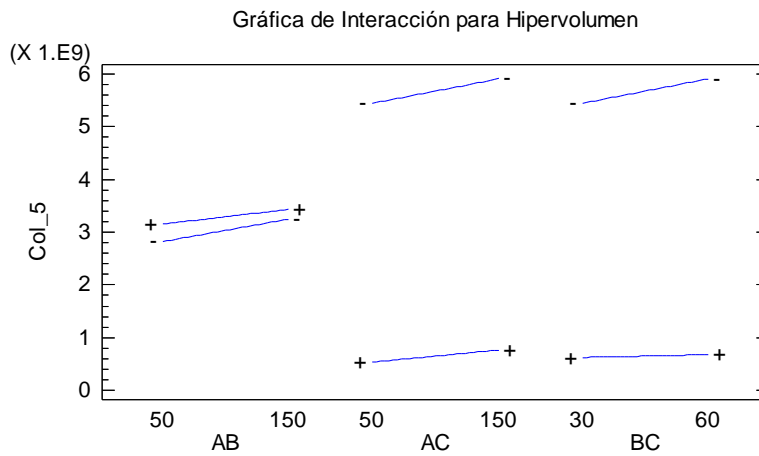


Ilustración 3.12 Gráfico de interacción de efectos n=100, m=40

3.3 Análisis del Algoritmo de Estimación de Distribución

A continuación, se presentan algunos gráficos que muestran el comportamiento del algoritmo de estimación de distribución en relación a las medidas de desempeño conforme pasan las generaciones establecidas.

En las ilustraciones 3.13, 3.14 y 3.14 se muestra el comportamiento de las mejores soluciones encontradas en cada generación de las corridas experimentales para un problema de 40 trabajos y 6 máquinas bajo el planteamiento de un problema flowshop con bloqueo y con mínima tardanza total y mínimo makespan como objetivos a alcanzar

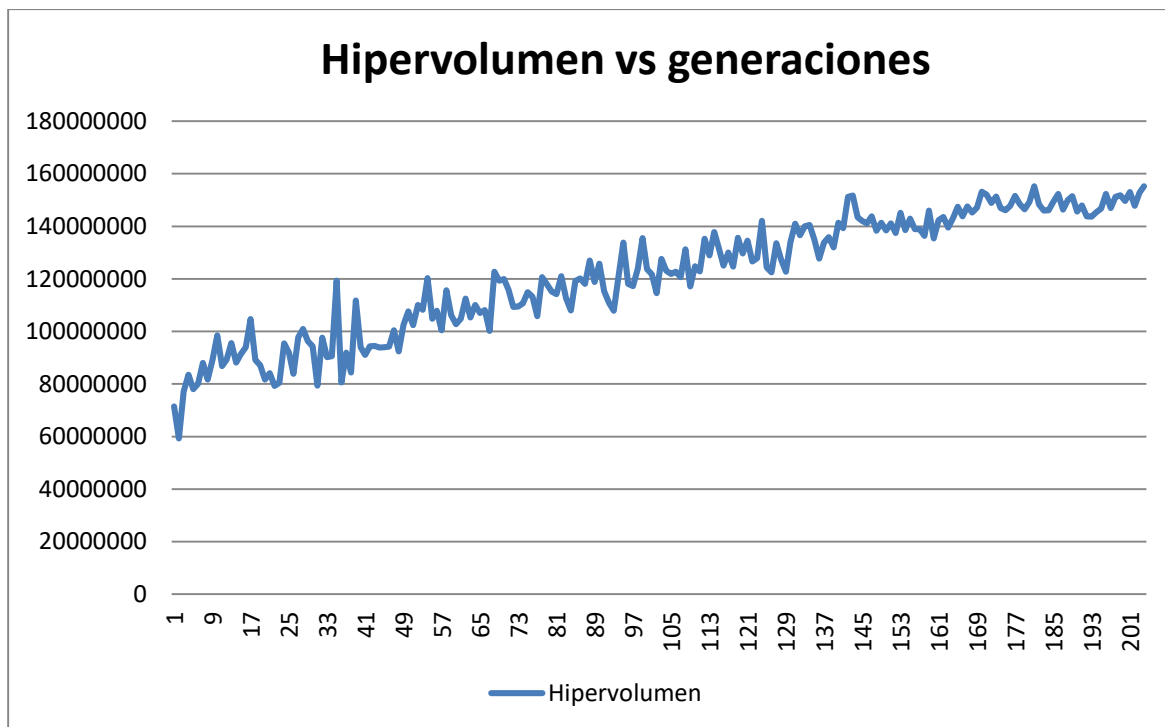


Ilustración 3.13. Hipervolumen vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 50 individuos

A partir de la ilustración anterior, se puede observar una tendencia positiva y ascendente en el comportamiento del hipervolumen encontrado en cada generación trabajada por el EDA, lo que indica un mejoramiento de la métrica de desempeño y de la aptitud de los individuos de la población en la medida que pasan las generaciones

En las ilustraciones 3.14. y 3.15 se observan los mejores valores de la tardanza total y el Cmax encontrados por el algoritmo EDA en cada generación. Teniendo en cuenta que se desea minimizar ambos objetivos, se observa un buen comportamiento de las soluciones a lo largo de las generaciones, a pesar que en algunas generaciones se pueden presentar empeoramientos en la medida de desempeño, la tendencia global es decreciente.

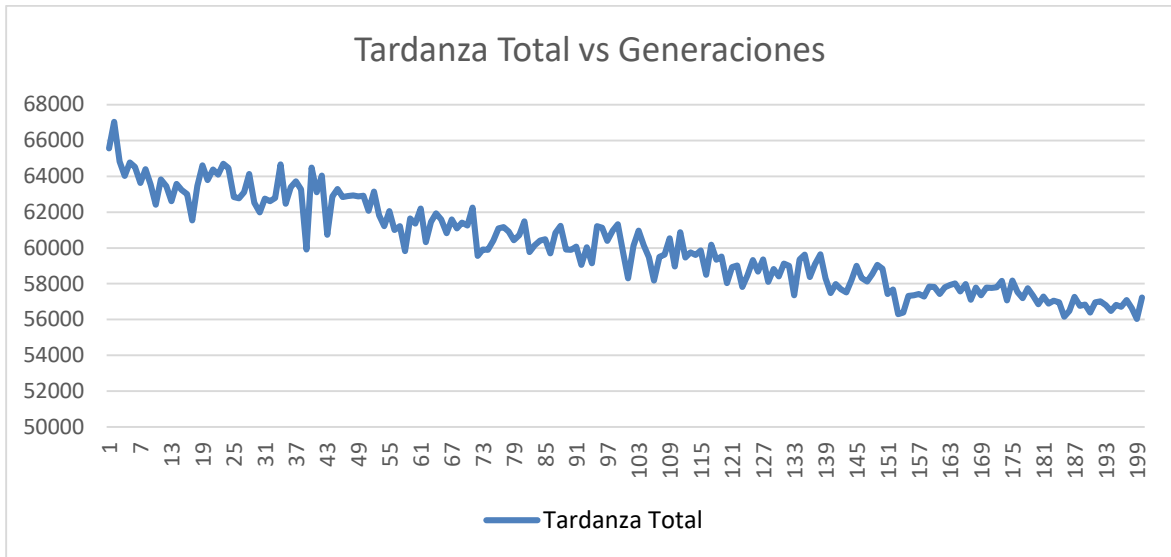


Ilustración 3.14 Tardanza Total vs Generaciones con EDA, 40 trabajos y 6 máquinas

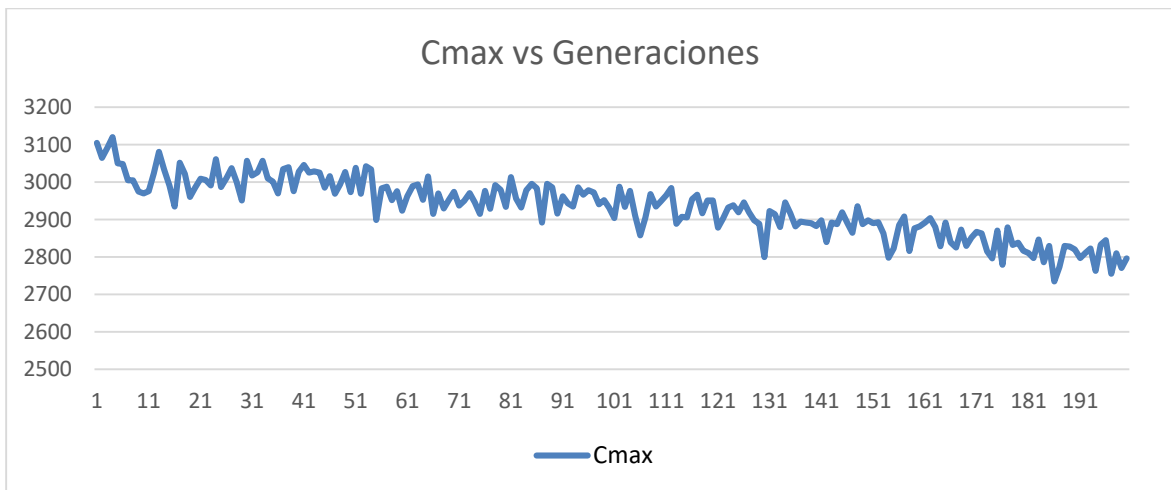


Ilustración 3.15 Makespan vs Generaciones con EDA, 40 trabajos y 6 máquinas

En las ilustraciones 3.16 a 3.18 se muestra el comportamiento del hipervolumen, makespan y tardanza total para un conjunto de 40 trabajos con 6 máquinas y una población inicial de 100 individuos para un problema flowshop con bloqueo.

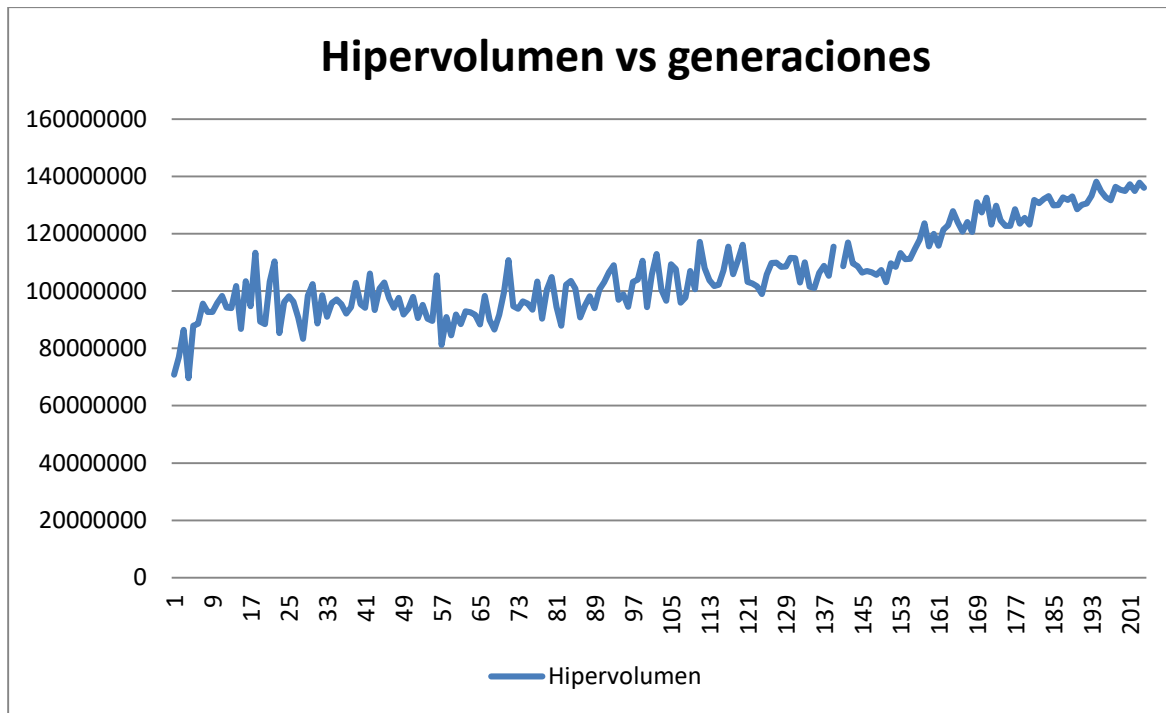


Ilustración 3.16 Hipervolumen vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos

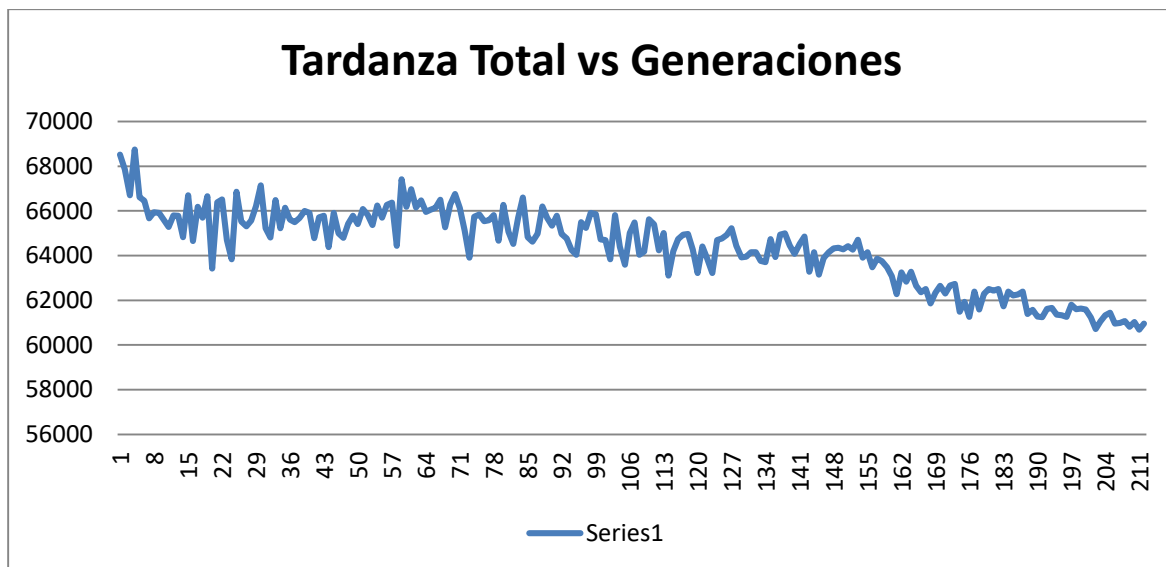


Ilustración 3.17 Tardanza Total vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos

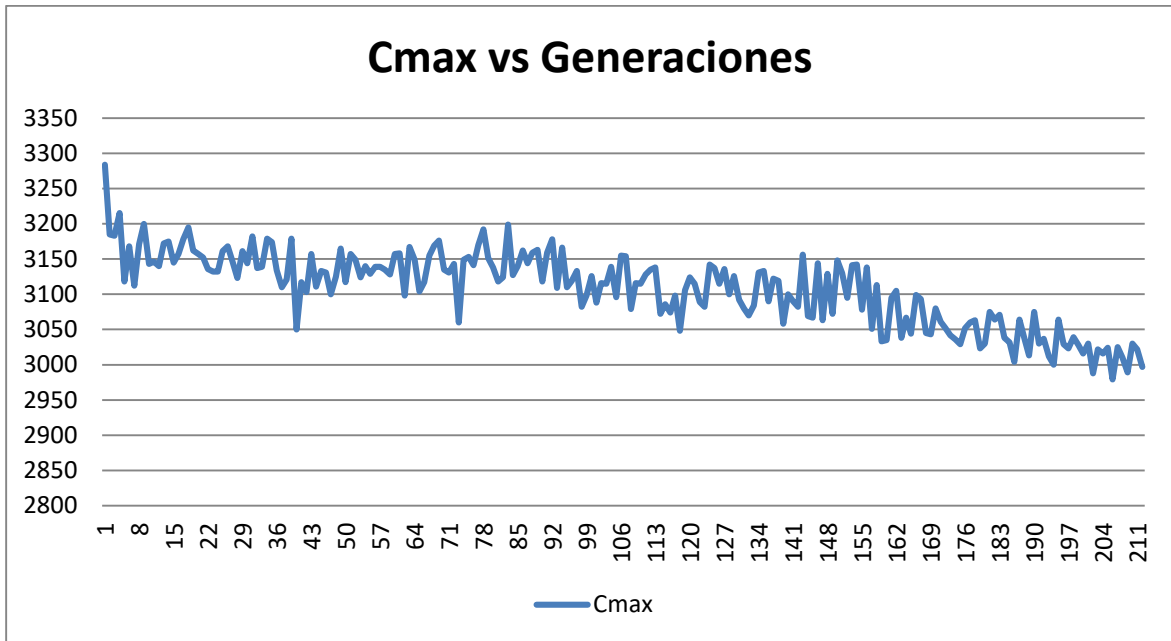


Ilustración 3.18 Cmax vs Generaciones con EDA, 40 trabajos, 6 máquinas, Pob. Inicial de 100 individuos

Al comparar el comportamiento de los objetivos y el hipervolumen para los casos con diferentes tamaños de población inicial de individuos, no se evidencia un cambio significativo en el comportamiento de cada objetivo y métrica a lo largo de las generaciones, por lo que se confirma la afirmación planteada en secciones previas sobre la poca significancia de esta variable en el comportamiento de búsqueda del algoritmo

4. Conclusiones y recomendaciones

4.1 Conclusiones

La presente tesis desarrolló en su totalidad tres capítulos: un marco teórico y referencial, un diseño y descripción de la metodología propuesta, y finalmente la experimentación con los resultados encontrados; lo cual permitió concluir que:

Al hacer la revisión de la literatura se evidencia una oportunidad de investigación en el área de programación de producción en ambientes flowshop con restricciones de bloqueo y con el fin de satisfacer múltiples objetivos.

Los algoritmos de estimación de distribución han tenido un creciente interés por parte de los investigadores para ser aplicados en diferentes ambientes asociados con la programación de producción, pero también en otras áreas de ciencias y salud.

El planteamiento matemático del problema de secuenciación en un ambiente flowshop con restricciones de bloqueo y múltiples objetivos ayudó a fortalecer el diseño del pseudocódigo de los algoritmos diseñados, de tal manera que procesaran y entregaran soluciones válidas y coherentes con el problema asociado.

Al comparar las soluciones encontradas por el algoritmo de estimación de distribución con las generadas por el algoritmo genético multiobjetivo, se evidenció que en todos los casos el EDA arrojó un mejor indicador de hipervolumen independiente del número de generaciones, población inicial, número de trabajos a secuenciar y número de máquinas en el flujo productivo

Al analizar el comportamiento del algoritmo de estimación de distribución se observa una tendencia de mejoramiento en la medida de desempeño, a pesar de que en algunas

generaciones puede empeorar su fitness en comparación con resultados anteriores, esta exploración ayuda a encontrar mejores valores de las funciones objetivo.

A partir de la experimentación realizada, se logra evidenciar la utilidad que tienen los algoritmos de estimación de distribución en la solución de problemas de secuenciación de trabajos en ambientes flowshop con restricciones de bloqueo y con múltiples objetivos

Es importante citar las bondades del algoritmo de estimación de distribución desarrollado, pues en su ejecución no requiere de la definición de parámetros; lo que reduce el sesgo relacionado con la estimación a priori de los niveles de los parámetros y su implicación en la ejecución y búsqueda de soluciones del metaheurístico

Al evidenciar el creciente interés de las investigaciones en problemas de optimización con múltiples objetivos, el desarrollo de Algoritmos de Estimación de Distribución contribuye como herramienta de búsqueda de soluciones que pueden satisfacer los resultados esperados por el decisor.

La métrica de desempeño denominada hipervolumen permite medir la convergencia y la diversidad de las soluciones no dominadas encontradas por los metaheurísticos para problemas multiobjetivo, siendo así ésta un buen indicador para comparar la eficiencia de los metaheurísticos respecto a las soluciones encontradas.

La aplicación del metaheurístico EDA es transversal a todas las áreas del conocimiento y los tipos de problemas de optimización que se desea resolver; desde programación entera, lineal, no lineal, difusa, dinámica; entre otros enfoques explorados en las investigaciones científicas.

4.2 Trabajos futuros

El desarrollo de este trabajo puede contribuir en la ejecución de futuras investigaciones en las siguientes temáticas:

Utilización de heurísticos y metaheurísticos en la generación de la población inicial de soluciones de tal forma que el EDA inicie su proceso de búsqueda de soluciones con base en unos individuos previamente seleccionados por su potencial en las medidas de desempeño.

Aplicación de la metaheurística EDA en otros ambientes de programación como job shop, open shop y supply chain pueden generar soluciones que sean competitivas con las metodologías actuales de solución de estos problemas.

A partir del algoritmo EDA desarrollado, se pueden generar algunos otros mecanismos y procedimientos para la estimación de funciones de probabilidad, de tal manera que se puedan hacer experimentos comparativos con el fin de revisar la eficiencia de cada mecanismo y el impacto en la calidad de las soluciones encontradas.

El planteamiento del problema puede considerar diferentes tipos de restricciones como no-wait, tiempos de preparación secuencialmente dependientes, no interrupciones, entre otras características que presentan los flujos productivos en el sector industrial y de servicios.

Es posible desarrollar el mismo problema planteado bajo el metaheurístico EDA, pero considerando una serie de objetivos diferentes como la demora, el número de trabajos tardíos, el flujo de producción, entre otros y comparar la eficiencia de las soluciones generadas en relación con los metaheurísticos existentes en la literatura.

El uso de hiperheurísticos puede mejorar el desempeño del EDA al combinar sus fortalezas con las correspondientes de otros metaheurísticos como algoritmos genéticos, colonias de hormigas, nubes de partículas, entre otros.

A. Anexo: Resultados obtenidos con el Algoritmo de Estimación de Distribución

A continuación, se relaciona los resultados obtenidos para las corridas experimentales del Algoritmo de estimación de distribución. En las tablas se presentan el makespan, la tardanza total y el hipervolumen obtenido en cada corrida para cada combinación de trabajos, máquinas, número de generaciones y número de individuos en la población inicial.

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	4918	149078	677112438
2	5126	153049	496277500
3	5058	149956	703222786
	5056	151463	
4	5046	153300	485912370
5	4987	149970	636032241
6	5046	151038	587317830
7	5101	151846	550419075
8	5061	153124	493637225
9	4964	151419	571280640
10	5039	153489	477514050
11	5068	155890	369621192
12	5005	150093	630258066
13	5125	150328	618487800
	5092	151056	
14	4967	149335	664832836
15	5121	153457	478072910

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
16	5008	150756	620470372
	5020	150618	
	5030	150374	
	5042	150310	
17	5144	154080	450550002
	5081	154761	
18	5044	152269	532155840
19	4996	152391	527250240
Mejor	4918	149078	703222786
Media	5040.8846	151635.6923	563682410
Desviación Estándar	55.768868	1776.930288	88588909.78

Tabla 1-A Resultados EDA con n=50, m=20, Po=50, Gen=30

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	5052	150698	602479384
2	4931	150148	628825495
3	5030	150682	603492622
4	5076	152955	501043200
5	4981	149669	649630650
6	5074	150457	612980964
7	4926	148402	707378150
8	4995	150249	623397090
9	4973	148034	723162815
10	5043	150538	609773633
11	5017	152620	516730821
12	4933	149197	671538306
13	4985	148329	709726710
14	5016	152116	539351780
15	4911	151450	570560885
16	4977	149256	668231817
17	4986	149614	652027250
18	4964	148193	716166752
19	4970	149560	654684574
Mejor	4911	148034	723162815
Media	4987.55	150010.05	629535942

Desviación Estándar	48.768519	1445.11643	199206727.7
----------------------------	-----------	------------	-------------

Tabla 2-A Resultados EDA con n=50, m=20, Po=150, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	4994	149261	667754396
2	5076	154340	438995200
3	4980	150556	609822368
4	5010	150523	610895456
5	5057	151693	557817274
6	4954	149236	669472566
7	4924	147223	760440962
	4922	147674	
8	4954	149955	637173648
9	4999	152230	534440193
10	5027	151332	574381143
11	5018	151289	576425300
12	4991	152171	537183680
13	5004	151702	563771554
	5006	151575	
14	4879	148823	720587112
	4936	148124	
15	5010	152246	533591338
16	4993	151850	551567187
17	5006	151345	574066780
18	5010	147743	735622936
19	5057	151998	544147479
Mejor	4879	147223	760440962
Media	4991.2273	150585.8636	599902977.5
Desviación Estándar	47.518349	1848.464982	81209093.67

Tabla 3-A Resultados EDA con n=50, m=20, Po=50, Gen=60

Número de Trabajos	50	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	4867	148218	716588289
2	4897	147150	764148231
3	5015	151756	555513763
4	4973	150542	610546091

5	4965	149453	659562946
Número de Trabajos	50	Generaciones	60
Número de máquinas	20	Población inicial	150
6	4971	148770	690144945
7	4954	150836	597597366
8	4981	149232	669249765
9	4986	151492	567723830
10	4897	150591	609375492
11	4982	147696	738192042
12	4970	147314	755543450
13	4925	148811	689008928
14	5018	150150	627518562
15	4971	149394	662124225
16	4963	149225	669832482
17	4997	150766	601261922
	4912	151256	
18	4953	149157	673036386
19	4990	148088	720465186
Mejor	4867	147150	764148231
Media	4959.35	149494.85	661970205.3
Desviación Estándar	40.39056	1389.921364	61382739.48

Tabla 4-A Resultados EDA con n=50, m=20, Po=150, Gen=60

Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	9173	502476	3454976822
	9109	505063	
2	9277	518807	1964817141
3	9201	507420	3002028430
4	9255	505796	3147849626
5	9214	513203	2475751110
6	9181	508021	2948019867
7	9278	516052	2215131348
8	9188	509443	2818453572
9	9160	510602	2713870752
10	9276	513780	2421632200
11	9318	518548	1987454532
12	9081	505683	3164166061
13	9218	512607	2628887848

	9269	511517	
Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
14	9184	513743	2427449520
15	9252	509205	2838102700
16	9198	511675	2615152230
17	9164	505973	3134897860
18	9176	501169	3571494648
19	9242	513391	2457899178
Mejor	9081	501169	3571494648
Media	9210.1905	510198.7619	2736212392
Desviación Estándar	59.040341	4856.7684	450432227.8

Tabla 5-A Resultados EDA con n=100, m=20, Po=50, Gen=60

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	9246	510877	2686308594
2	9294	516870	2140426000
3	9315	518768	1967537798
4	9225	513677	2432356007
5	9169	511133	2665294575
6	9303	514948	2314810362
7	9169	513879	2415477225
8	9235	510974	2677815504
9	9285	522261	1650817171
10	9289	517291	2102293845
11	9319	523751	1514870175
	9366	523511	
12	9325	521413	1727104923
13	9104	513372	2463360320
14	9283	513625	2435529105
15	9231	516651	2161820227
16	9275	516582	2167043912
17	9142	517947	2045997966
18	9263	508837	2871203433
19	9304	514942	2315329920
Mejor	9104	508837	2871203433

Media	9257.1	516065.45	2250284056
Desviación Estándar	67.542501	4296.369256	366934704.8

Tabla 6-A Resultados EDA con n=100, m=20, Po=50, Gen=30

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	9234	514110	2392751200
2	9158	513269	2471270746
3	9160	510118	2757907008
4	9149	508604	2896006870
5	9185	507037	3037393887
6	9179	513064	2489348190
7	9131	503721	3340996217
8	9257	512741	2516570143
9	9222	514426	2364335688
10	9205	507533	2991620283
11	9251	509563	2805594231
12	9235	505298	3193814988
13	9204	503071	3397427460
14	9102	507551	2993369918
15	9170	509839	2782985634
16	9317	512830	2506825200
17	9162	511055	2672596250
18	9149	510264	2744955170
19	9180	511416	2639229496
Mejor	9102	503071	723162815
Media	9192.1053	509763.6842	2789210452
Desviación Estándar	51.425129	3389.469871	888634019.4

Tabla 7-A Resultados EDA con n=100, m=20, Po=150, Gen=30

Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	9180	508621	2893473876
2	9229	512303	2557166205
3	9185	505025	3220403395
4	9206	507286	3014049072
5	9158	514293	2378101082
6	9117	510166	2754841128
7	9210	512544	2535785524

8	9141	507245	3019934555
Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
9	9258	513317	2464192118
10	9180	507566	2989440896
11	9110	510327	2740396502
12	9161	504363	3281483861
13	9199	516438	2234863806
	9231	515856	
14	9177	513470	2452470320
15	9117	506490	3089456380
16	9248	508358	2915216512
17	9112	506002	3134049696
18	9132	499140	3759429264
	9088	505344	
19	9183	507125	3029456105
Mejor	9088	499140	3759429264
Media	9172.4762	509108.5238	2866537384
Desviación Estándar	48.526919	4304.804079	373449121.5

Tabla 8-A Resultados EDA con n=100, m=20, Po=150, Gen=60

Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	6597	219155	1264746496
2	6546	224965	713132618
3	6594	223075	892343312
4	6513	215923	1573202048
5	6564	223175	967128376
	6574	222291	
6	6639	218997	1279525983
	6612	220098	
7	6514	219060	1274885037
8	6578	216550	1512544759
9	6492	221264	1065550939
10	6559	217198	1451257374
11	6637	220592	1127745000
12	6692	224224	782367859

13	6610	221243	1066223880
Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
14	6632	221130	1236066883
	6636	219452	
15	6576	223977	807746330
	6593	223967	
16	6545	220985	1091478920
17	6564	216673	1501077554
18	6556	223187	882054720
19	6568	220011	1183780872
Mejor	6492	215923	1573202048
Media	6582.2174	220747.4783	1140676787
Desviación Estándar	46.844963	2630.561915	259253748

Tabla 9-A Resultados EDA con n=50, m=40, Po=50, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	6394	216603	1510427304
2	6422	218463	1332942732
3	6614	222043	990186184
4	6488	216069	1559728566
5	6614	226204	594928633
6	6548	219219	1259315136
7	6644	221173	1072972123
	6601	221244	
8	6480	219561	1227683250
9	6545	220632	1125035100
10	6435	215865	1580012340
11	6498	215759	1589047756
12	6496	218616	1317354759
13	6632	220625	1124670266
14	6589	218117	1363479600
15	6514	219696	1214407161
16	6523	218545	1323731604
17	6501	218395	1338303488
18	6504	219970	1188477197
19	6488	216131	1554820872

	6423	217243	
Mejor	6394	215759	1589047756
Media	6521.5714	219055.8571	1277238109
Desviación Estándar	72.716966	2507.158816	243685108.8

Tabla 10-A Resultados EDA con n=50, m=40, Po=50, Gen=60

Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	6486	219148	1266889961
2	6485	218766	1303239120
3	6516	217749	1399519902
4	6486	215675	1597238248
5	6388	216739	1497572976
6	6496	219439	1239080052
7	6526	219108	1270160361
8	6465	217888	1387046060
9	6636	219004	1278567647
10	6514	216827	1487223240
11	6518	219213	1260283098
12	6526	216634	1505385807
13	6570	218946	1284968235
14	6566	219056	1490586851
15	6536	216788	1431954697
16	6528	217406	1446865552
17	6429	217265	1632878490
18	6460	215305	1820064348
19	6483	213333	1531646890
Mejor	6388	213333	723162815
Media	6506	217594.1579	1427956397
Desviación Estándar	54.061693	1637.785234	453814737.7

Tabla 11-A Resultados EDA con n=50, m=40, Po=150, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	6503	215738	1590961358
2	6476	217364	1436733287
3	6431	216688	1501750546
4	6396	216768	1494686091

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
5	6472	215995	1567030776
6	6439	217971	1379526336
7	6493	216266	1540909512
8	6381	217873	1389699056
9	6477	213579	1796777664
10	6510	217226	1449342895
11	6549	217701	1403596896
12	6449	216713	1499087624
13	6428	216954	1476480801
14	6493	217565	1417359024
15	6506	219052	1275753085
16	6517	217564	1417096464
17	6450	214577	1702322950
18	6474	216389	1529516218
19	6367	215742	1592855550
Mejor	6367	213579	723162815
Media	6463.7368	216722.3684	1497972954
Desviación Estándar	48.273345	1262.066084	464780382.5

Tabla 12-A Resultados EDA con n=50, m=40, Po=150, Gen=60

Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	10674	657971	6193058500
	10767	654839	
	10769	652663	
2	10786	664220	3971324764
3	10760	657674	5228681534
4	10761	656299	5492650176
5	10772	657295	5301117432
6	10804	670738	2719796226
7	10889	662012	4692909795
	10898	660448	
8	10757	652768	6170720139
9	10755	657189	5321939154

Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
10	10829	659828	4813387391
11	10765	653788	5974630167
12	10844	658757	5018551100
13	10782	648591	6971800416
14	10813	655803	5586367280
15	10778	661987	4400181600
16	10719	660288	4727808141
17	10707	657612	5242032045
18	10811	660867	4614405880
19	10811	658044	5156272261
Mejor	10674	648591	723162815
Media	10784.136	658167.3182	5136717579
Desviación Estándar	52.706099	4561.419583	917799423.3

Tabla 13-A Resultados EDA con n=100, m=40, Po=50, Gen=30

Número de Trabajos	100	Generaciones	60
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	10776	653944	5944338666
2	10906	658915	4986617184
3	10803	651581	6397092330
4	10798	658834	5004973080
5	10779	656138	5523043851
6	10804	657340	5291595918
7	10787	657261	5307230266
8	10709	647734	7139037477
9	10767	656806	5395139091
10	10879	663463	4114653276
11	10760	655398	5665668982
	10806	654278	
12	10831	660710	4644057619
13	10665	644820	7700432091
14	10816	661095	4570522904
	10809	662331	

Número de Trabajos	100	Generaciones	60
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
15	10802	656178	5514703924
	10798	656402	
16	10754	644190	7817826868
	10748	651491	
17	10721	650961	6518888002
18	10688	652488	6226717330
19	10762	656753	5405455384
Mejor	10665	644190	723162815
Media	10781.217	655178.7391	5745683908
Desviación Estándar	54.840228	5079.755383	1013118464

Tabla 14-A Resultados EDA con n=100, m=40, Po=50, Gen=60

Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	10695	652761	6174057198
2	10718	657852	5195642200
3	10762	657284	5303505508
4	10776	659675	4844089824
5	10727	654210	5894775607
6	10782	655759	5595716448
7	10747	651537	6407407070
8	10698	654388	5861479140
	10693	655512	
9	10779	664886	3843611559
10	10715	650079	6688473604
11	10760	651038	6502780262
	10738	653697	
12	10782	658591	5052040416
13	10683	650764	6558016725
14	10838	657488	5262254480
15	10772	652386	6243576706
16	10693	655569	5634802970
17	10738	652946	6137151220
18	10740	651305	6452188836
19	10798	655817	5584116400
Mejor	10683	650079	723162815

Media	10744.476	654930.6667	5749246641
Desviación Estándar	40.982459	3577.830297	722017037.8

Tabla 15-A Resultados EDA con n=100, m=40, Po=150, Gen=30

Número de Trabajos	100	Generaciones	60
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	10702	651479	6420047968
2	10841	662155	4366495584
3	10724	654080	5919832118
4	10677	649943	6715920084
5	10751	658048	5157116013
6	10619	652005	6321757378
7	10793	649933	6713783910
8	10740	652179	6284365104
9	10734	652130	6293970648
10	10739	648704	7056121105
	10741	648160	
11	10679	651347	6446171240
12	10810	656722	5410054380
13	10759	658353	5098341446
14	10746	652123	6294921408
15	10772	652707	6181949200
16	10722	655539	5639713248
17	10685	650213	6663780662
18	10756	654093	5916349628
19	10620	651386	6440657898
Mejor	10619	648160	723162815
Media	10730.5	653064.95	6070597317
Desviación Estándar	56.064343	3553.016028	671935740.6

Tabla 16-A Resultados EDA con n=100, m=40, Po=150, Gen=60

B. Anexo: Resultados obtenidos con el Algoritmo Genético

A continuación, se relaciona los resultados obtenidos para las corridas experimentales del Algoritmo genético multiobjetivo. En las tablas se presentan el makespan, la tardanza total y el hipervolumen obtenido en cada corrida para cada combinación de trabajos, máquinas, número de generaciones y número de individuos en la población inicial.

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	5305	164139	186382996
2	5414	160482	165684275
	5415	160338	
	5418	159947	
3	5397	160414	149485350
4	5426	160776	161835618
5	5330	160506	179984724
6	5248	160106	189702060
7	5345	159879	181839258
8	5373	160053	209696032
9	5364	159428	163091838
10	5230	160486	259349697
11	5383	158310	160522260
12	5410	160529	198136190
	5391	159685	
13	5219	156206	189282030
14	5253	161047	354263981
15	5410	159884	137974316
	5388	160539	

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
16	5334	160218	182130558
	5344	160050	
17	5202	159809	193438420
	5202	159809	
18	5287	160648	181430220
	5335	159522	
19	5386	160061	205813365
Mejor	5202	156206	354263981
Media	5338.8077	160110.4231	192107536.2
Desviación Estándar	73.000285	1242.420707	47178856.81

Tabla 1-B Resultados AG con n=50, m=20, Po=50, Gen=30

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	5192	159330	214885356
2	5201	159126	223955775
3	5331	158498	251278345
4	5355	158517	250297062
5	5344	157382	300902724
6	5196	161037	138597360
7	5187	158726	241901557
8	5200	157838	281503476
9	5107	157489	300041526
	5113	157437	
10	5249	159425	210371678
11	5235	158491	252132368
12	5309	159575	203403788
13	5367	159095	224503396
14	5258	160161	177490404
15	5233	158168	266563353
16	5254	159516	206287506
17	5303	159766	194917729
18	5324	162473	187413824
	5383	159929	
19	5301	160653	155388450

Número de Trabajos	50	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
20	5313	160184	176246665
Mejor	5107	157382	300902724
Media	4826.0417	159218.9091	202640114.6
Desviación Estándar	1477.5159	1257.473919	78471320.94

Tabla 2-B Resultados AG con n=50, m=20, Po=150, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	5285	161041	138142918
2	5460	161631	111395328
3	5416	160710	152453340
4	5380	160033	182700576
5	5333	159425	209975702
6	5434	161079	135992520
7	5385	159870	189932079
8	5194	160542	160721154
9	5299	161053	137564622
10	5379	159498	206510577
11	5398	160126	178490214
12	5413	159851	190657344
13	5327	160313	170444474
14	5389	160481	162733446
15	5389	160002	199566239
	5396	159653	
16	5254	162400	77597658
17	5195	159217	219919882
18	5458	161755	105892512
19	5273	158767	239607316
20	5334	157975	274556888
Mejor	5194	157975	274556888
Media	5351.9524	160258.1905	172242739.5
Desviación Estándar	77.749261	1038.684582	47453163

Tabla 3-B Resultados AG con n=50, m=20, Po=50, Gen=60

Número de Trabajos	50	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	5411	160249	172968850
2	5304	159009	228654360
3	5251	159778	194609625
4	5269	158570	248416383
5	5359	159289	215907450
6	5236	158762	240029280
7	5317	161346	124453287
8	5278	161088	136068498
9	5378	159105	224002932
10	5269	160802	148853559
11	5219	160436	165364871
12	5291	159894	189439643
	5245	160306	
13	5384	159529	104111024
	5364	161555	
14	5366	161518	116660710
15	5326	160945	142292700
16	5294	160843	146942272
17	5233	162839	58035900
18	5223	158530	250458677
19	5240	161847	102305712
20	5262	159033	227799084
Mejor	5219	158530	250458677
Media	5296.3182	160239.6818	171868740.9
Desviación Estándar	58.612843	1190.486671	55985422.48

Tabla 4-B Resultados AG con n=50, m=20, Po=150, Gen=60

Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	9522	537461	269056718
2	9445	535784	421387554
3	9524	537063	305117540
4	9361	530540	897843870
5	9429	535807	419375445
	9430	535777	

Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
6	9502	536325	372085410
7	9465	536376	367612666
8	9505	535049	487728459
9	9504	536601	347060560
10	9390	531793	783842298
11	9524	537063	305117540
12	9476	537474	299384698
	9479	537128	
13	9524	537063	305117540
14	9503	536803	328754907
15	9478	536109	391767786
16	9496	535383	457500456
17	9516	536609	346289588
18	9372	536215	382603980
19	9524	537063	305117540
20	9511	536415	363891495
Mejor	9361	530540	897843870
Media	9476.3636	535995.5	407832802.5
Desviación Estándar	50.91594	1702.69265	159495542.7

Tabla 5-B Resultados AG con n=100, m=20, Po=50, Gen=60

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	9490	535521	445020486
2	9495	535244	470105714
3	9494	536625	344923250
4	9522	532900	682383660
5	9454	536294	375093840
6	9524	537063	305117540
7	9476	534257	559693564
8	9524	537063	305117540
9	9459	536046	397563040
10	9465	535894	411319944
11	9472	534478	539679744
12	9524	537063	305117540

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	50
Corrida	Cmax	T	Hipervolumen
13	9524	537063	305117540
14	9495	536832	326155102
15	9485	535936	407421546
16	9467	535750	507964256
	9476	534828	
17	9478	535680	430663500
18	9515	536137	389070297
19	9409	527911	1135911465
20	9475	536574	349619664
	9474	536608	
Mejor	9409	527911	1135911465
Media	9486.2273	535534.8636	449652961.6
Desviación Estándar	28.755011	2004.764475	189807782.3

Tabla 6-B Resultados AG con n=100, m=20, Po=50, Gen=30

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	9503	535322	462994228
2	9464	536767	332160840
3	9375	527867	1140330947
4	9505	540430	0
5	9343	528078	1121573952
6	9385	532391	729611601
7	9470	533310	645598880
8	9538	533107	663507738
9	9443	535755	424027175
10	9301	530770	877543380
11	9487	535767	422733591
12	9402	527041	1214944638
13	9356	530943	861305756
14	9503	537149	297393121
15	9486	534836	537418286
	9493	534502	
16	9485	535917	409144067

Número de Trabajos	100	Generaciones	30
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
17	9393	534913	571730228
	9406	534131	
	9374	535175	
18	9347	536198	384252904
19	9433	531066	849417804
20	9405	531515	808938185
Mejor	9301	527041	1214944638
Media	9430.3043	533606.5217	637731366.1
Desviación Estándar	64.797892	3265.03774	314552763.6

Tabla 7-B Resultados AG con n=100, m=20, Po=150, Gen=30

Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	9523	534904	500771646
2	9496	535970	404290080
3	9473	534495	538132385
4	9439	536784	330710430
5	9475	534203	564595863
6	9465	533136	661412626
7	9482	536090	393473080
8	9479	534748	515158530
9	9426	533762	604907624
10	9431	533449	633267453
11	9497	535552	442300590
	9470	535818	
12	9450	536439	548424562
	9451	534383	
13	9518	536062	395854368
14	9446	533778	603323096
15	9440	532056	759555296
16	9451	534240	561535638
	9427	534348	
17	9443	531817	781207713
18	9398	525840	1323984140
19	9485	534358	550481448
20	9429	532025	762459575

Mejor	9398	525840	1323984140
Número de Trabajos	100	Generaciones	60
Número de máquinas	20	Población inicial	150
Corrida	Cmax	T	Hipervolumen
Media	9460.6087	534098.1304	593792307.2
Desviación Estándar	31.230433	2276.320342	212963686.3

Tabla 8-B Resultados AG con n=100, m=20, Po=150, Gen=60

Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	6881	231713	71421896
2	6818	231203	119810768
3	6819	230991	139904136
4	6848	231565	85470814
5	6815	230053	228823060
6	6838	230611	175887552
7	6757	230287	206779902
	6751	230590	
8	6776	231061	133329574
9	6846	231014	137684827
10	6715	231523	89576160
11	6818	227070	534028536
	6824	226833	
12	6820	230782	159712725
13	6841	231804	62828532
14	6826	229815	251353908
15	6828	226680	548474499
16	6868	230545	182084514
17	6782	230613	175801842
18	6742	230343	201489012
19	6798	230479	188476316
20	6861	230500	186361448
	6809	231134	
Mejor	6715	226680	548474499
Media	6812.2174	230313.4348	193965001.1
Desviación Estándar	41.839473	1460.488269	129422958

Tabla 9-B Resultados AG con n=50, m=40, Po=50, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	6823	229742	258280950
2	6847	230810	157014006
3	6812	231118	127875757
4	6800	230531	183542480
5	6796	229455	285564708
6	6830	229763	261105023
	6832	229712	
7	6804	229688	263451979
8	6848	231553	86607898
9	6813	231101	129485872
10	6824	230230	212031163
	6821	230445	
11	6875	231936	50301630
12	6807	231713	71477692
13	6788	232361	10050602
14	6825	231137	126075550
	6803	231642	
15	6799	231862	57357630
16	6864	231861	57413046
17	6871	231690	73608318
18	6773	230322	203414640
19	6858	232467	0
20	6828	230105	223863274
Mejor	6773	229455	285564708
Media	6823.5217	230923.6522	141926110.9
Desviación Estándar	27.026703	935.9943963	90037649.37

Tabla 10-B Resultados AG con n=50, m=40, Po=50, Gen=60

Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	6807	229691	263213126
	6785	230018	
2	6710	227122	507213775
3	6802	230343	201361572
4	6697	228068	417500292
5	6770	229668	265443165

Número de Trabajos	50	Generaciones	30
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
6	6780	230182	216675125
7	6628	228708	357018543
8	6654	227201	500011966
9	6810	228468	379179074
	6767	230284	
10	6741	227758	446714576
11	6784	229670	265214337
12	6803	229688	263454758
13	6758	226243	590349328
	6754	227067	
14	6779	229120	317382622
15	6739	229396	291333486
16	6754	229307	299729160
17	6829	230634	173724408
18	6796	229044	324531207
19	6775	229287	301559400
20	6754	228689	358347078
Mejor	6628	226243	590349328
Media	6759.8261	228941.5652	336997849.9
Desviación Estándar	48.969984	1199.43001	109260011.2

Tabla 11-B Resultados AG con n=50, m=40, Po=150, Gen=30

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	6836	230858	152483321
2	6785	230141	220551320
3	6758	231637	78723010
4	6810	231173	189013530
	6821	230473	
5	6812	228420	383627271
6	6724	229291	301342056
7	6737	231211	119154208
8	6805	230272	208086000
9	6826	229724	259981456
	6825	229808	
10	6698	226746	542962947

Número de Trabajos	50	Generaciones	60
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
11	6655	228503	376381800
12	6788	229748	257807423
13	6818	230151	307287416
	6831	229225	
14	6754	227654	456517863
15	6751	230090	225467958
16	6769	228371	388448256
17	6810	229812	251680725
18	6810	229150	314516015
	6785	229227	
19	6790	230875	150945480
20	6838	230274	207824031
Mejor	6655	226746	542962947
Media	6784.8333	229701.4167	269640104.3
Desviación Estándar	46.54373	1168.916924	116642519.6

Tabla 12-B Resultados AG con n=50, m=40, Po=50, Gen=30

Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	10955	681666	645032757
	10961	681544	
2	10913	681415	669922740
3	10989	682294	501092397
4	11025	684826	15530373
5	11012	679828	973877934
6	10936	683087	349116040
7	10967	683245	318756642
8	11072	676295	1650799832
9	10980	682444	472349214
10	11025	684826	15530373
11	10941	681944	568353771
12	11003	683324	303548165
13	11009	682008	555880351
14	11025	684826	15530373
15	11120	680039	932893784
16	10950	684030	168215616

17	10962	681730	609335892
Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
18	10986	682978	369928188
19	11077	682046	548399341
20	10958	682561	449962800
Mejor	10913	676295	1650799832
Media	10993.619	682236	506702829.2
Desviación Estándar	51.465014	1933.901781	379473935.3

Tabla 13-B Resultados AG con n=100, m=40, Po=50, Gen=30

Número de Trabajos	100	Generaciones	60
Número de máquinas	40	Población inicial	50
Corrida	Cmax	T	Hipervolumen
1	10944	681147	721220640
2	10971	684034	167430051
3	11039	682509	459828508
	11000	682693	
4	10951	682470	467433659
5	10857	678857	1161001050
6	10940	668602	3127592490
7	10916	680393	865974788
8	10944	682495	462655368
9	10998	683781	215921760
10	10957	683700	231503807
11	10985	682387	483267960
12	10981	682796	404841247
13	11016	684907	0
14	10988	684619	55229760
15	10988	683656	239904270
16	11037	682075	542953872
17	10946	681621	630294232
18	11023	684481	81679110
	10988	685401	
19	11007	683934	186573723
20	10977	681131	724165056
Mejor	10857	668602	3127592490
Media	10975.136	682167.6818	561473567.6
Desviación Estándar	42.008271	3413.441084	672114834.5

Tabla 14-B Resultados AG con n=100, m=40, Po=50, Gen=60

Número de Trabajos	100	Generaciones	30
Número de máquinas	40	Población inicial	150
Corrida	Cmax	T	Hipervolumen
1	10972	682041	549658676
2	10951	681752	605151085
3	11042	678055	1419269691
	11049	677504	
4	10981	682800	404074139
5	10981	678345	1258440674
6	10957	680065	928700442
7	10914	679896	961330284
8	11025	684826	340258315
	11065	683132	
9	11009	680378	868431221
10	11054	679661	1005679184
11	11025	684826	15530373
12	11009	677639	1393631732
13	10944	683288	310546866
14	11039	683904	192294157
15	10961	682865	391649474
16	10960	680838	780426062
17	10978	683194	328545286
	10961	683369	
18	10987	681650	624598147
19	10972	680119	918271368
20	10928	676910	1534064510
Mejor	10914	676910	1534064510
Media	10989.739	681176.3913	741527584.3
Desviación Estándar	42.302179	2402.971281	437293494.1

Tabla 15-B Resultados AG con n=100, m=40, Po=150, Gen=30

Bibliografía

Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3), 985-1032.

Allouche, M. A., Aouni, B., Martel, J. M., Loukil, T., & Rebaï, A. (2009). Solving multi-criteria scheduling flow shop problem through compromise programming and satisfaction functions. *European journal of operational research*, 192(2), 460-467.

Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., ... & Larrañaga, P. (2008). A review of estimation of distribution algorithms in bioinformatics. *BioData mining*, 1(1), 6.

Arroyo, J. E. C., & Armentano, V. A. (2005). Genetic local search for multi-objective flowshop scheduling problems. *European Journal of operational research*, 167(3), 717-738.

Arroyo, J. E. C., & de Souza Pereira, A. A. (2011). A GRASP heuristic for the multi-objective permutation flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 55(5), 741-753.

Baesler, F., Ceballos, L., y Ramírez, M. (2006). Programación multiobjetivo de máquinas moldreras a través de algoritmos meméticos. *Maderas. Ciencia y Tecnología*, 8(3), 183-192.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*. John Wiley & Sons .

Becerra Rodríguez, F., Cárdenas Aguirre, D. M., Castrillón Gómez, Ó. D., García, A., Giraldo García, J. A., Ibarra Mirón, S., ... & Zapata Gómez, A. (2008). Gestión de la producción: Una aproximación Conceptual.

Bellman, R., & Gross, O. (1954). Some combinatorial problems arising in the theory of multi-stage processes. *Journal of the Society for Industrial and Applied Mathematics*, 2(3), 175-183.

Bellman, R., Esogbue, A. O., & Nabeshima, I. (2014). *Mathematical Aspects of Scheduling and Applications: Modern Applied Mathematics and Computer Science* (Vol. 4). Elsevier.

Bertel, S., & Billaut, J. C. (2004). A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159(3), 651-662.

Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *The International Journal of Production Research*, 20(1), 27-45.

Bulfin, R. L., & M'Hallah, R. (2003). Minimizing the weighted number of tardy jobs on a two-machine flow shop. *Computers & Operations Research*, 30(12), 1887-1900.

Caraffa, V., Ianes, S., Bagchi, T. P., & Sriskandarajah, C. (2001). Minimizing makespan in a blocking flowshop using genetic algorithms. *International Journal of Production Economics*, 70(2), 101-115.

Carlier, J., & Rebaï, I. (1996). Two branch and bound algorithms for the permutation flow shop problem. *European Journal of Operational Research*, 90(2), 238-251.

Castrillón, O., Giraldo, J., & Sarache, W. (2010). Técnicas inteligentes y estocásticas en scheduling. Un enfoque en la producción y las operaciones. Unibiblios, Bogotá.

Chase, R. B. A., Jacobs, N., Chase, R., Aquilano, N., Jacobs, R., Heizer, J., ... & MILDRED, T. (2000). *Administración de producción y operaciones: manufactura y servicios*. McGraw-Hill,.

Chen, B., & Vestjens, A. P. (1997). Scheduling on identical machines: How good is LPT in an on-line setting?. *Operations Research Letters*, 21(4), 165-169.

Chen, J. C., Wu, C. C., Chen, C. W., & Chen, K. H. (2012). Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. *Expert Systems with Applications*, 39(11), 10016-10021.

Chou, F. D., & Lee, C. E. (1999). Two-machine flowshop scheduling with bicriteria problem. *Computers & industrial engineering*, 36(3), 549-564.

Coca Ortegón, G. A. *Diseño y aplicación de una metodología multiobjetivo, basada en los algoritmos genéticos "VEGA" y "MOGA", para mejorar el desempeño de las variables tiempo de procesamiento, porcentaje de producción defectuosa causada por la fatiga del operario y costo de mano de obra directa, en un sistema de fabricación tipo "Job Shop" de una compañía metalmecánica*(Doctoral dissertation, Universidad Nacional de Colombia-Sede Manizales).

Coca, G.A. Castrillón, O.D. y Ruiz, S. (2013). Metodología basada en los algoritmos VEGA y MOGA para solucionar un problema multiobjetivo en un sistema de producción Job Shop. *Revista EIA*, 2013 (19), 175 – 191.

Coello, C. A. C. (2003). Evolutionary multi-objective optimization: A critical review. In *Evolutionary optimization* (pp. 117-146). Springer US.

Coello, C. A. C., & Zacetenco, C. S. P. (2004). Introducción a la computación evolutiva (Notas de curso). *CINVESTAV-IPN, Departamento de Ingeniería Eléctrica, Sección de Computación. México, DF.*

Cuartas Torres, B. A. *Metodología para la optimización de múltiples objetivos basada en ag y uso de preferencias* (Doctoral dissertation, Universidad Nacional de Colombia).

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4), 577-601.

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.
- Domínguez Machuca, J. A. (1995). Dirección de Operaciones: Aspectos estratégicos en la producción y los servicios.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics*, *IEEE Transactions on*, 26(1), 29-41.
- Duan, W., Li, Z., Yang, Y., Liu, B., & Wang, K. (2017, June). EDA based probabilistic Memetic Algorithm for distributed blocking permutation flowshop scheduling with sequence dependent setup time. In *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 992-999). IEEE.
- Echegoyen, C., Santana, R., Mendiburu, A., & Lozano, J. A. (2015). Comprehensive characterization of the behaviors of estimation of distribution algorithms. *Theoretical Computer Science*, 598, 64-86.
- Echegoyen, C., Zhang, Q., Mendiburu, A., Santana, R., & Lozano, J. A. (2011, June). On the limits of effectiveness in estimation of distribution algorithms. In *Evolutionary Computation (CEC), 2011 IEEE Congress on* (pp. 1573-1580). IEEE.
- Feng, H., Lu, S., & Li, X. (2009, July). Genetic algorithm for hybrid flow-shop scheduling with parallel batch processors. In *Information Engineering, 2009. ICIE'09. WASE International Conference on* (Vol. 2, pp. 9-13). IEEE.
- Fink, A., & Voß, S. (2003). Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, 151(2), 400-414.
- Frutos, M., Olivera, A. C., & Tohmé, F. (2010). A memetic algorithm based on a NSGAI scheme for the flexible job-shop scheduling problem, *Ann. Oper. Res.*, vol. 181, no. 1, 745– 765.
- García, A. M. M., Miguel, A., Merelo Guervós, J. J., & Castillo Valdivieso, P. Á. (2009). Resolución del problema militar de búsqueda de camino óptimo multiobjetivo mediante el uso de algoritmos de optimización basados en colonias de hormigas. Editorial de la Universidad de Granada.
- Gaither, N., & Frazier, G. (2000). Administración de producción y operaciones.
- Gallego, R. A., Escobar, A., & Romero, R. (2006). Técnicas de optimización combinatorial. *Universidad Tecnológica de Pereira*, 19-77.
- Gómez, P. (2007, September). Un nuevo algoritmo genético basado en un sistema multiagente para la programación de la producción en un taller de flujo híbrido. In *XI Congreso de Ingeniería de Organización* (pp. 1675-1685).
- Grabowski, J., & Pempera, J. (2007). The permutation flow shop problem with blocking. A tabu search approach. *Omega*, 35(3), 302-311.
- Gupta, J. N., Hennig, K., & Werner, F. (2002). Local search heuristics for two-stage flow shop problems with secondary criterion. *Computers & Operations Research*, 29(2), 123-149.
- Gupta, D., & Singh, H. (2015). A String of Disjoint Job Blocks on Two Stage Open Shop Scheduling with Transportation Time. *Control Theory and Informatics*, 5(2), 42-47.

Han, Y., Gong, D., Li, J., & Zhang, Y. (2016). Solving the blocking flow shop scheduling problem with makespan using a modified fruit fly optimisation algorithm. *International Journal of Production Research*, 54(22), 6782-6797.

Hansen, P., Mladenovic, N., y Moreno, J.A. (2003). Búsqueda de entorno variable. *Revista Iberoamericana de Inteligencia Artificial*, 7(19), 77-92.

Hauschild, M. W., Pelikan, M., Sastry, K., & Goldberg, D. E. (2012). Using previous models to bias structural learning in the hierarchical BOA. *Evolutionary Computation*, 20(1), 135-160.

Heizer, J., y Render, B. (2004). Principios de Administración de Operaciones (5.ª Ed.). México D.F: Pearson Educación.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems*. The University of Michigan.

Hsu, H. M., Hsiung, Y., Chen, Y. Z., & Wu, M. C. (2009). A GA methodology for the scheduling of yarn-dyed textile production. *Expert Systems with Applications*, 36(10), 12095-12103.

Huang, X. W., Zhao, X. Y., & Ma, X. L. (2014). An improved genetic algorithm for job-shop scheduling problem with process sequence flexibility. *International Journal of Advanced Manufacturing Technology*, 28(5-6), pp 532-540.

Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3), 392-403.

Ishibuchi, H., Yoshida, T., & Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE transactions on evolutionary computation*, 7(2), 204-223.

Iyer, S. K., & Saxena, B. (2004). Improved genetic algorithm for the permutation flowshop scheduling problem. *Computers & Operations Research*, 31(4), 593-606.

Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European journal of operational research*, 137(1), 50-71

Jarboui, B., Eddaly, M., & Siarry, P. (2009). An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers & Operations Research*, 36(9), 2638-2646.

Jin, F., Gupta, J. N., Song, S., & Wu, C. (2010). Single machine scheduling with sequence-dependent family setups to minimize maximum lateness. *Journal of the Operational Research Society*, 61(7), 1181-1189.

Kammer, M., Van den Akker, M., & Hoogeveen, H. (2011). Identifying and exploiting commonalities for the job-shop scheduling problem. *Computers & Operations Research*, 38(11), 1556-1561.

Karimi, N., & Davoudpour, H. (2014). A high performing metaheuristic for multi-objective flowshop scheduling problem. *Computers & Operations Research*, 52, 149-156.

Kazanas, H., & Lerwick, L. P. (2003). Manufacturing Processes Technology. *Encyclopedia of Physical Science and Technology (Third Edition)*, 95–119.

- Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6), 975-986.
- Kononov, A. (2015). O (log m)-approximation for the routing open shop problem. *Rairo operations research*. Volumen: 49 Número: 2 383-391.
- Koren, Y., Wang, W., & Gu, X. (2017). Value creation through design for scalability of reconfigurable manufacturing systems. *International Journal of Production Research*, 55(5), 1227-1242.
- Koulamas, C., & Kyparisis, G. (2015). The three-machine proportionate open shop and mixed shop minimum makespan problems. *European journal of operational research*. Volumen: 243 Número: 1 70-74.
- Krajewski, L.; Ritzman, L. (2000) Administración de Operaciones. Estrategia y Análisis (5a ed). Pearson Education, México D.F.
- Larrañaga P., Exteberria R., Lozano J., Peña J., (2000). Optimization in continuous domains by learning and simulation of Gaussian networks. En Wu A. (Ed.), *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 201-204.
- Liaw, C. F. (2013). An improved branch-and-bound algorithm for the preemptive open shop total completion time scheduling problem. *Journal of Industrial and Production Engineering*, 30(5), 327-335.
- López, Juan Camilo. Metodología de programación de producción en un flow shop híbrido flexible con el uso de algoritmos genéticos para reducir el makespan. Aplicación en la industria textil. 2013. Tesis de maestría. Universidad Nacional de Colombia. Sede Manizales
- Loranca, M. B. B., Rosales, J. E. E., & Rodríguez, J. R. (2009). Evaluación de un algoritmo de recocido simulado con superficies de respuestas. *Revista de Matemática: Teoría y Aplicaciones*, 16(1), 159-177
- Lozano, J. A. (Ed.). (2006). *Towards a new evolutionary computation: advances on estimation of distribution algorithms* (Vol. 192). Springer Science & Business Media.
- Mahdavi, I., Mojarad, M. S., Javadi, B., & Tajdin, A. (2008, December). A genetic approach for solving a hybrid flow shop scheduling problem. In *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on* (pp. 1214-1218). IEEE.
- Martín Martín, Q., Santos Martín, M. T., & de Paz Santana, Y. R. (2003). Investigación Operativa. Problemas y ejercicios resueltos. . Investigación operativa: problemas y ejercicios resueltos, Pearson Prentice Hall, 407 pages. 2005.
- Medina, R., Pradenas, L., & Parada V. (2011). Un algoritmo genético para el problema de Job Shop Flexible. *Ingeniería-Revista chilena de ingeniería*, 19(1), 53-61.
- Merchan, A. F., & Maravelias, C. T. (2016). Preprocessing and tightening methods for time-indexed MIP chemical production scheduling models. *Computers & Chemical Engineering*, 84, 516-535.
- Minella, G., Ruiz, R., and Ciavotta, M. (2008). A review and evaluation of multi-objective algorithms for the owshop scheduling problem. *INFORMS Journal on Computing*, 20(3):451 471.

Montgomery, D. C. (2004). *Diseño y Análisis de Experimentos*. México: Limusa Wiley

Moraglio, A., Ten Eikelder, H., & Tadei, R. (2005). Genetic local search for job shop scheduling problem. Tech. Rpt. CSM-435, Dept. of Comput. Sci., Univ. of Essex.

Mühlenbein, H., & Paass, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel problem solving from nature—PPSN IV*, 178-187.

Neto, R. T., & Godinho Filho, M. (2013). Literature review regarding Ant Colony Optimization applied to scheduling problems: Guidelines for implementation and directions for future research. *Engineering Applications of Artificial Intelligence*, 26(1), 150-161.

Pan, J. C. H., Chen, J. S., & Chao, C. M. (2002). Minimizing tardiness in a two-machine flow-shop. *Computers & Operations Research*, 29(7), 869-885.

Pan, Q. K., & Ruiz, R. (2012). An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega*, 40(2), 166-180.

Pan, Q. K., & Wang, L. (2012). Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. *Omega*, 40(2), 218-229.

Panahi, H., & Tavakkoli-Moghaddam, R. (2011). Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization. *Expert Systems with Applications*, 38(3), 2817-2822.

Pérez, R., Jöns, S., & Hernández, A. (2015). Solución de un problema de secuenciamiento en configuración jobshop flexible utilizando un Algoritmo de Estimación de Distribuciones. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 12(1), 49-57..

Phanden, R. K., Jain, A., & Verma, R. (2012). A genetic algorithm-based approach for job shop scheduling. *Journal of Manufacturing Technology Management*, 23(7), 937-946.

Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. Springer (New York).

Pinedo, M. (2012). *Scheduling*. New York: Springer

Rahimi-Vahed, A. R., & Mirghorbani, S. M. (2007). A multi-objective particle swarm for a flow shop scheduling problem. *Journal of combinatorial optimization*, 13(1), 79-102.

Rabello, G.R., Moretti, A.C., y Salles, N. L.L. (2008). A grasp metaheuristic for the ordered cutting stock problem. *Revista Chilena de Ingeniería*, 16(3), 421-427.

Reeves, C. R., & Yamada, T. (1998). Genetic algorithms, path relinking, and the flowshop sequencing problem. *Evolutionary computation*, 6(1), 45-60.

Rego, M. F., Souza, M. J. F., & Arroyo, J. E. C. (2012, November). Multi-objective Algorithms for the Single Machine Scheduling Problem with Sequence-dependent Family Setups. In *Chilean Computer Science Society (SCCC), 2012 31st International Conference of the* (pp. 142-151). IEEE.

Riquelme, N., Von Lüken, C., & Baran, B. (2015, October). Performance metrics in multi-objective optimization. In *Computing Conference (CLEI), 2015 Latin American* (pp. 1-11). IEEE.

- Roshanaei V. (2012). Mathematical Modelling and Optimization of Flexible Job Shops Scheduling Problem. Electronic Theses and Dissertations. Paper 157,
- Ruiz, Santiago. Metodología multiobjetivo basada en un comportamiento evolutivo para programar sistemas de producción flexible job shop. Aplicaciones en la industria metalmecánica. 2015. Universidad Nacional de Colombia. Sede Manizales.
- Santana, R., Larrañaga, P., & Lozano, J. A. (2008). Protein folding in simplified models with estimation of distribution algorithms. *IEEE transactions on Evolutionary Computation*, 12(4), 418-438.
- Sarıçiçek, İ., & Çelik, C. (2011). Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness. *Applied Mathematical Modelling*, 35(8), 4117-4126.
- Scgroeder, R. G. (1988). *Administración de operaciones: Toma de decisiones en la función de operaciones*. McGraw-Hill,.
- Schrimpf, G., Schneider, J., Stamm, H., y Dueck G. (1999). Record Breaking Optimization Results Using the Ruin and Recreate Principle. *Journal of computational Physics*, 159, 139-171
- Shapiro, J. L. (2005). Drift and scaling in estimation of distribution algorithms. *Evolutionary computation*, 13(1), 99-123.
- Siarry, P., & Michalewicz, Z. (Eds.). (2007). *Advances in metaheuristics for hard optimization*. Springer Science & Business Media.
- Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2), 59-66.
- Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetics algorithms. *Evolutionary computation*, 2(3), 221-248.
- Toktaş, B., Azizoğlu, M., & Köksalan, S. K. (2004). Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan. *European Journal of Operational Research*, 157(2), 286-295.
- Toro, E., Garcés, A., & Ruiz, H. (2008). Solución al problema de empaquetamiento bidimensional usando un algoritmo híbrido constructivo de búsqueda en vecindad variable y recocido simulado. *Revista Facultad de Ingeniería Universidad de Antioquia*, 46, 119- 131.
- Toscano, G. (2001). Optimización multiobjetivo usando un micro algoritmo genético. Tesis de maestría, Facultad de Física e Inteligencia Artificial, Universidad Veracruzana, Universidad Veracruzana.
- Toumi, S., Jarboui, B., Eddaly, M., & Rebai, A. (2013, May). Branch and bound algorithm for solving blocking flowshop scheduling problem with total completion time. In *Control, Decision and Information Technologies (CoDIT), 2013 International Conference on* (pp. 746-749). IEEE.
- Trabelsi, W., Sauvey, C., & Sauer, N. (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. *Computers & Operations Research*, 39(11), 2520-2527.

Ulusoy, G. (2004). Multiprocessor task scheduling in multistage hybrid flow-shops: a genetic algorithm approach. *Journal of the Operational Research Society*, 55(5), 504-512.

Wang, J. B., Ng, C. D., Cheng, T. E., & Liu, L. L. (2006). Minimizing total completion time in a two-machine flow shop with deteriorating jobs. *Applied Mathematics and Computation*, 180(1), 185-193.

Wang, L., Zhang, L., & Zheng, D. Z. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 33(10), 2960-2971.

Yalaoui, N., Mahdi, H., Amodeo, L., & Yalaoui, F. (2011, March). A particle swarm optimization under fuzzy logic controller to solve a scheduling problem. In *Communications, Computing and Control Applications (CCCA), 2011 International Conference on* (pp. 1-6). IEEE.

Yenisey, M. M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, 45, 119-135.

Zhang, S. (2010, June). Large-scale flow shop scheduling based on genetic algorithm. In *Education Technology and Computer (ICETC), 2010 2nd International Conference on* (Vol. 1, pp. V1-308). IEEE.

Zhang, Q., Manier, H., & Manier, M. A. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. *Computers & Operations Research*, 39(7), 1713-1723.

Zhang, Y., Dai, G., Peng, L., & Wang, M. (2014, July). Hmoeda_ile: A hybrid multi-objective estimation of distribution algorithm combining locally linear embedding. In *Evolutionary Computation (CEC), 2014 IEEE Congress on* (pp. 707-714). IEEE.

Zhou, S., Li, X., Chen, H., & Guo, C. (2016). Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. *International Journal of Production Research*, 54(16), 4919-4937.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32-49.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4), 257-271

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm.