

Tipo de artículo: Artículo original  
Temática: Inteligencia artificial  
Recibido: 4/11/2013 | Aceptado: 17/12/2013 | Publicado: 21/01/2014

## **Influencia de los parámetros principales de un Algoritmo Genético para el Flow Shop Scheduling**

### *Behavior of the main parameters of the Genetic Algorithm for Flow Shop Scheduling Problems*

**Yunior César Fonseca Reyna<sup>1\*</sup>, Yailen Martínez Jiménez<sup>2</sup>, Ángel Enrique Figueredo León<sup>1</sup>, Luis Alberto Pernía Nieves<sup>1</sup>**

<sup>1</sup> Departamento de Informática. Universidad de Granma, km 18 ½, Carretera a Manzanillo, Bayamo, Granma, Cuba

<sup>2</sup> Departamento de Ciencia de la Computación. Universidad Central “Marta Abreu” de las Villas, Carretera a Camajuaní, km 5 ½, Santa Clara, Villa Clara, Cuba

\*Autor para correspondencia: [fonseca@udg.co.cu](mailto:fonseca@udg.co.cu)  
[yailenm@uclv.edu.cu](mailto:yailenm@uclv.edu.cu)

---

#### **Resumen**

Existen valores sugeridos para adaptar los parámetros básicos de un Algoritmo Genético, sin embargo, estos valores pueden no ser los óptimos para todo tipo de aplicaciones. En la siguiente investigación se presenta una metaheurística basada en un Algoritmo Genético para resolver problemas de scheduling de tipo Flow Shop con el objetivo de minimizar el tiempo de finalización de todos los trabajos, conocido en la literatura como makespan o  $C_{\max}$ . Este problema es típico de la optimización combinatoria y se presenta en talleres con tecnología de maquinado donde existen máquinas-herramientas convencionales y se fabrican diferentes tipos de piezas que tienen en común una misma ruta. Se implementa un conjunto de operadores de cruzamiento y de selección para el Algoritmo Genético propuesto, y una vez calibrados los factores principales del mismo, como son el tamaño de la población, número de

generaciones, factor de mutación y el factor de cruzamiento, se realiza un estudio estadístico para determinar de las combinaciones de estos parámetros, cuales tienen una mayor influencia. Por último, la combinación de parámetros de mejor desempeño se prueba con problemas de diferentes niveles de complejidad de la literatura especializada con el objetivo de obtener resultados satisfactorios en cuanto a la calidad de las soluciones.

**Palabras clave:** Algoritmos genéticos, flow shop, makespan, optimización, secuenciación.

### **Abstract**

*There are different suggested values to adapt the basic parameters of a Genetic Algorithm, however, these values may not be the optimal for all kinds of applications. The following research presents a metaheuristic based on a Genetic Algorithm to solve problems of type Flow Shop Scheduling with the objective of minimizing the completion time of all jobs, known in literature as makespan or  $C_{max}$ . This problem is typical of combinatorial optimization and can be found in manufacturing environment, where there are conventional machines-tools and different types of pieces which share the same route. A set of crossover and selection operators are implemented methods for the proposed Genetic Algorithm once its main factors are calibrated, the size of the population, number of generations, mutation and crossover factors, statistical study is performed in order to determine the combinations of these parameters that has a greater influence. Finally, the combination of parameters whit the best performance is tested with problems of different levels of complexity in order to obtain satisfactory results in terms of solutions quality.*

**Keywords:** Genetic algorithms, flow shop, makespan, optimization, scheduling.

---

## **Introducción**

El problema de secuenciación de tareas *Flow Shop Scheduling Problems* (FSSP, por sus siglas en inglés) es un problema clásico de la programación de trabajos. La solución del modelo matemático consiste en encontrar una secuencia de tareas que emplee un tiempo mínimo de procesamiento (Pinedo, 2008). Esta programación de operaciones es una prioridad dentro del plan de acción de cualquier tipo de empresa porque de ello depende, en buena parte, la rentabilidad de la misma. Dichos problemas deben ser resueltos en una amplia gama de aplicaciones tales como: programación de despacho de vuelos en los aeropuertos, programación de líneas de producción de una fábrica, programación de cirugías en un hospital, reparación de equipos o maquinarias en un taller (Márquez, 2012), construcción de piezas en procesos tecnológicos de maquinado, entre otros.

Este problema está incluido dentro de la gran variedad de problemas de planificación de recursos, el cual, como muchos otros en este campo, es de difícil solución y está clasificado técnicamente como de solución en un tiempo no polinomial (*NP-hard*) (Ancau, 2012; Čičková, *et al.*, 2010).

El desarrollo actual de las computadoras y la aparición de nuevas técnicas de simulación y optimización heurística que aprovechan plenamente las disponibilidades de cálculo intensivo que estas proporcionan, han abierto una nueva vía para abordar los problemas de secuenciación o problemas de *scheduling* como también se le conocen, suministrando un creciente arsenal de métodos y algoritmos cuyo uso se extiende paulatinamente sustituyendo a las antiguas reglas y algoritmos usados tradicionalmente.

En los últimos años se han propuesto un gran número de enfoques para modelar y solucionar el problema de la planificación de tareas con diferentes grados de éxito. Entre estos enfoques se pueden mencionar la *Programación Matemática* (Ramezani, *et al.*, 2010) y dentro de este ámbito la *Programación Lineal Entera Mixta* (Šeda, 2007). Otros métodos utilizados para resolver el problema de la programación de la producción han sido las técnicas de *Ramificación y Acotación* (Sierra, 2009) y las heurísticas basadas en *Cuello de Botellas* (Yamada, 2003).

Con el desarrollo de la Inteligencia Artificial (*IA*) han emergido otras metodologías como es el caso de las *Redes Neuronales* (Zhang, 1995) y últimamente los métodos de Búsqueda Locales y Metaheurísticas, entre los cuales se encuentra el *Recocido Simulado* (Álvarez, *et al.*, 2008); la *Búsqueda Tabú* (Jungwattanakit, *et al.*, 2009), la *Optimización basada en Nubes de Partículas (PSO)* (Quan-Ke, y otros, 2008), *Optimización basada en Colonias de Hormigas (ACO)* (Yecit, *et al.*, 2009) y los *Algoritmos Genéticos (AGs)* (Chaudhry, *et al.*, 2012), (Márquez, 2012), (Matía, 2010), por solo mencionar algunas, siendo esta última la metodología utilizada en el desarrollo de esta investigación.

Los AGs son particularmente aplicados en problemas complejos de optimización: problemas con diversos parámetros o características que precisan ser combinados en función de encontrar mejores soluciones; problemas con muchas restricciones o condiciones que no pueden ser representadas matemáticamente, y problemas con grandes espacios de búsqueda.

Aunque los AGs son métodos robustos para resolver un amplio rango de problemas, la solución de un problema determinado requiere un conjunto de parámetros específicos que garantice un desempeño satisfactorio. La optimización de estos parámetros es una tarea que consume gran cantidad de tiempo y que demanda a menudo la intervención subjetiva de un experto.

Existen valores sugeridos para adaptar estos parámetros, sin embargo, estos valores pueden no ser los óptimos para todo tipo de aplicaciones. También es común analizar cada parámetro por separado, sin tener en cuenta la influencia de cada uno sobre los demás.

El problema de encontrar los parámetros básicos de control óptimos en un AG (factor de reproducción, factor de mutación y el tamaño de la población) han sido estudiado ampliamente durante las últimas décadas (Palominos, 2010; Eiben, 2009; Goldberg, 1992), determinándose así desempeños aceptables en la optimización de funciones modificando los valores de estos por separados, así como todos a la vez. A pesar de estos análisis, de igual forma, existen otros parámetros en los AGs que influyen notablemente en los resultados de los problemas de optimización y que no se pueden dejar pasar por alto. Nos referimos al *operador de selección*, al *operador de cruzamiento* y al uso del *elitismo* empleados en los mismos.

En esta investigación se pretende realizar un estudio estadístico para determinar cuáles de estos operadores tienen una mayor influencia sobre el comportamiento y el desempeño de un AG en el problema de planificación en procesos tecnológicos de maquinado, teniendo en cuenta las interacciones que se presentan entre ellos. Por último, se aplicará el AG ya con los parámetros de mayor influencia a problemas clásicos de secuenciación para comparar los resultados con los propuestos por la literatura especializada.

## **Materiales y métodos**

Los sistemas productivos pueden ser clasificados en tres grandes categorías: producción artesanal o por unidad (producción discreta no-repetitiva), producción mecanizada o masiva (producción discreta repetitiva), y la producción de proceso continuo. El problema del *FSSP* está enmarcado en la producción mecanizada o masiva (Toro, y otros, 2006). Este consiste en programar de forma óptima un conjunto de  $N$  tareas que deben de ser procesadas en un conjunto de  $M$  máquinas, considerando que todas las tareas tienen la misma secuencia de producción. El objetivo es minimizar el tiempo total requerido para terminar todas las tareas (*makespan*).

### **Descripción del Problema de Secuenciación de Tareas o Flow Shop Scheduling**

El *FSSP* está sujeto a las siguientes restricciones:

- Solamente se cuenta con una máquina-herramienta de cada tipo (Ej.: un torno, un taladro, una fresadora, una rectificadora cilíndrica, etc.).

- Las restricciones tecnológicas están bien definidas y son previamente conocidas, además de que son inviolables.
- No está permitido que dos operaciones del mismo trabajo se procesen simultáneamente.
- Ningún trabajo puede ser procesado más de una vez en la misma máquina.
- Cada trabajo es procesado hasta concluirse, una vez que se inicia una operación esta se interrumpe solamente cuando se concluye.
- Ninguna máquina puede procesar más de un trabajo a la vez.
- Los tiempos de configuración y de cambio de máquina están incluidos en los tiempos totales de procesamiento.

Bajo estas condiciones, el *makespan* o  $C_{max}$  como también se le conoce, corresponde al tiempo del último trabajo en la última máquina (Toro, *et al.*, 2006b). En otras palabras, es el tiempo para completar todos los trabajos.

En la Tabla 1 se solicita el ordenamiento de 4 trabajos que deben ejecutarse en 3 máquinas cumpliendo con los tiempos de operación presentados.

Tabla 1. Tiempos de Procesamiento

$P_{ji}$	$i = 1$	2	3	4
$j = 1$	4	4	3	2
2	7	3	2	1
3	5	5	3	1

La Figura 1 muestra a través de un diagrama de Gantt la secuencia de trabajos  $X = [2, 1, 3, 4]$  donde  $C_{max} = 24$ .

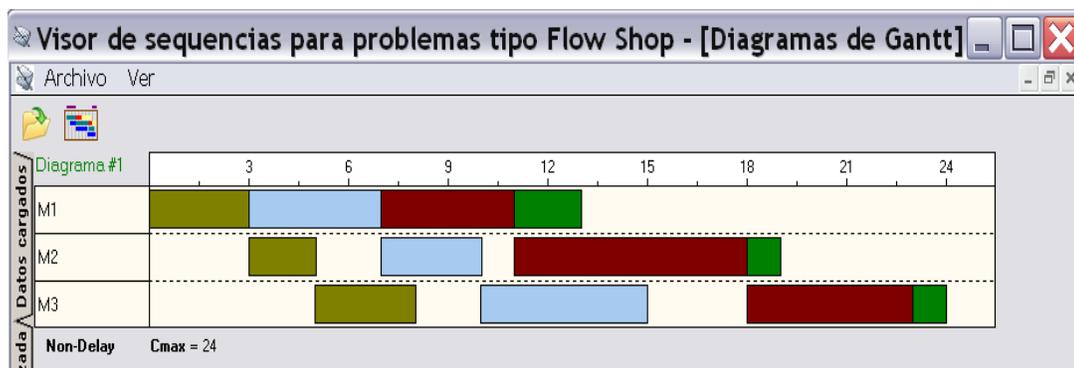


Figura 1. Visualización de la solución a través del Diagrama de Gantt.

## Algoritmo Genético para el Flow Shop Scheduling

Los Algoritmos Genéticos son técnicas de búsqueda estocásticas e iterativas basadas en los mecanismos de evolución natural que pueden usarse para resolver problemas de búsqueda y optimización. Estos usan una estrategia de selección y operadores para generar y explorar nuevos puntos en el espacio de soluciones que originen una nueva población, la cual estará constituida por un conjunto de individuos denominados cromosomas que están compuestos por elementos menores llamados genes, los cuales codifican las instrucciones, parámetros y demás elementos usados para construir una solución del problema a resolver. Los AG utilizan un conjunto de individuos - soluciones - sobre los cuales se aplican operadores y estrategias para encontrar soluciones mejoradas del problema. El conjunto de individuos que usa el AG se denomina población. Los AG no buscan una única solución, sino que están orientados a que, en promedio, los individuos de la población busquen una mejora en la función objetivo de las soluciones, de manera que en posteriores generaciones se enriquezcan (evolucionen), preservando sólo las mejores características de los individuos de generaciones anteriores (Matía, 2010; Sushil, *et al.*, 2009).

Como en todos los problemas que se enfrentan en el mundo real, para poder resolverlos se tiene que encontrar una forma de abstraerlos y poder representar sus posibles soluciones. Varias han sido las formas de codificar las soluciones para el problema del *FSSP* usando Algoritmos Genéticos, (Márquez, 2012; Yamada, 2003; Holsapple, 1993). Para el desarrollo de esta investigación se utilizó el método propuesto por Jacob Holsapple, el cual consiste en construir un vector de tamaño  $n$  (que corresponde al número de trabajos a ejecutar). Por lo tanto, la  $k$ -ésima posición del vector representa el trabajo que se realizará en el  $k$ -ésimo lugar. La población de alternativas de solución se conforma por un número determinado de cromosomas como el mostrado en la Figura 2. El cromosoma representa la secuencia natural en la que se programaran los trabajos requeridos. A continuación, se muestra un cromosoma conformado por una secuencia de números que representa los distintos trabajos (piezas), y donde coinciden genes y alelos.

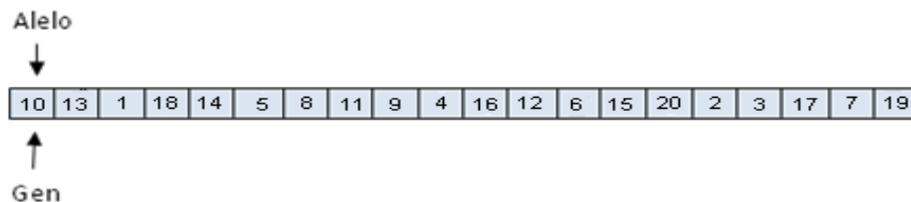


Figura 2. Representación de un cromosoma para un problema tipo *Flow Shop* de 20 trabajos.

## Resultados y discusión

Dado que los AGs son mecanismos de carácter estocástico y no exacto, su validez como método de búsqueda de soluciones debe ser realizada de forma experimental. En general se deben evaluar no solamente la eficiencia y eficacia como en cualquier otro método de búsqueda, sino también la estabilidad por tratarse de un método de naturaleza estocástica.

En el caso de los problemas de *scheduling* existen bancos de ejemplos de uso común entre los investigadores, lo cual facilita la comparación de distintos métodos de solución así como evaluar la influencia de cada uno de ellos variando internamente algún parámetro de control. En esta sección se realiza un estudio experimental apoyado en métodos estadísticos en instancias de problemas seleccionados de la literatura especializada permitiendo determinar la influencia de los métodos de selección y de los operadores de cruzamiento implementados para el problema de secuenciación en procesos tecnológicos de maquinado, así como el uso del elitismo en un AG, determinando cuál de las combinaciones presenta un mejor desempeño. Al final de la sección, la combinación seleccionada se le aplica a otras instancias del problema de secuenciación y se realiza una comparación entre los resultados arrojados por el algoritmo con los resultados propuestos por otros autores.

El algoritmo desarrollado tiene la posibilidad de usar seis operadores de selección: *Selección por Rueda de Ruleta*, *Selección por Torneo*, *Selección por Ranking*, *Selección Uniforme*, *Selección por Muestreo Determinístico* y *Selección por Muestreo Estocástico*; cuatro operadores de cruzamiento: *Un Punto de Cruce*, *Dos Puntos de Cruce*, *Tres Puntos de Cruce* y *el Partial Matched Crossover (PMX)*; y un solo operador de mutación: *Operador de Mutación de Orden*.

Inicialmente, para encontrar el tamaño de la población, el número de generaciones, el factor de cruzamiento y el factor de mutación óptimos para el algoritmo, se escogieron tres problemas estándar de diferentes tamaños y complejidad, específicamente: ta001, ta035, ta076 disponible en la Internet en el repositorio *OR-Library*<sup>1</sup> y se crearon parámetros iniciales para cada uno. Luego se fueron variando los valores de los mismos hasta lograr una calibración que presentara las mejores soluciones. Estos problemas fueron evaluados 10 veces cada uno con varias combinaciones de parámetros y se tomó la combinación más estable en cuanto a la calidad de los resultados. A

---

<sup>1</sup> OR-Library: <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>

continuación se muestran los parámetros iniciales que mejores resultados reportaron en las pruebas efectuadas a las instancias de problemas anteriormente mencionadas. Cabe mencionar que solo se variaron los parámetros anteriormente mencionados. El método de cruzamiento, selección, mutación y el uso del elitismo se mantuvieron inalterables.

Tabla 2. Parámetros iniciales del Algoritmo Genético

Parámetro	Valor
Número de Generaciones	5000
Cantidad de cromosomas	100
Factor de Cruzamiento	0.53
Factor de Mutación	0.021
Método de Selección	Selección por Torneo
Método de Cruzamiento	Un Punto de Cruce
Método de Mutación	Mutación por Valor
Tipo de Modelo	Modelo Estacionario

Con los factores de cruzamiento y mutación, así como el número de generaciones y el tamaño de la población previamente calibrados, se evaluaron las 48 combinaciones posibles de operador de mutación y operador de cruce haciendo uso indistintamente del elitismo para cada una de las instancias anteriormente mencionadas. Para cada combinación se realizaron 10 ejecuciones para un total general de 480 para cada instancia.

Para un análisis más detallado de estos resultados, tomamos de cada ejecución las 100 mejores secuencias, es decir, las secuencias de mejor makespan (en nuestro caso la población de la última generación) completando así 1000 observaciones para cada una de las posibles combinaciones, logrando al final 48000 observaciones por instancia. A partir de estas se realizó un análisis estadístico para evaluar el desempeño de cada una de las posibles combinaciones determinando así la de mejor desempeño para cada instancia.

### Metodología Estadística de Evaluación

El comportamiento no determinístico de los algoritmos sobre múltiples conjuntos de datos es una razón por la cual no existe un procedimiento establecido para poder compararlos. En la literatura especializada, existen numerosas técnicas de comparaciones múltiples, como por ejemplo, los procedimientos de Tukey, Scheffe, Dunnet, etc. (García, *et al.*, 2008), (Shesking, 2006). Sin embargo, la mayoría de ellos requieren un diseño ANOVA anterior a su

aplicación. La familia de procedimientos derivados del test de Bonferroni no requiere dicha condición, por lo que pueden ser aplicados en un conjunto de hipótesis cualquiera; en nuestro caso, el obtenido por el test de Friedman.

En esta investigación se aplican las pruebas no paramétricas debido a que los resultados obtenidos no cumplen las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas. Para demostrar lo planteado anteriormente se realizó la prueba de Kolmogorov Smirnov.

Nos interesa abordar las comparaciones múltiples, de tal forma que una combinación se pueda comparar con 2 o más combinaciones simultáneamente mientras que el nivel de confianza estadístico es previamente definido. Distintas técnicas estadísticas son utilizadas para tratar de determinar si las diferencias encontradas entre dos algoritmos son significativas. Se aplicará el test de Friedman para detectar diferencias entre las posibles combinaciones, si no se detectan diferencias, se puede concluir que las combinaciones involucradas obtienen resultados que no difieren significativamente unas de otras. En caso de que sea necesario establecer una comparación entre dos combinaciones, se realiza una prueba de Wilcoxon.

Una vez aplicado el Test de Friedman a los valores de makespan arrojados por cada una de las combinaciones, el mismo obtuvo valor 0.000 detectándose de esta forma diferencias significativas. La de menor valor de rangos fue la variante: *Dos puntos de cruce-Ranking-Con Elitismo* afirmándose que dicha combinación es la que arroja mejores resultados para el problema en cuestión.

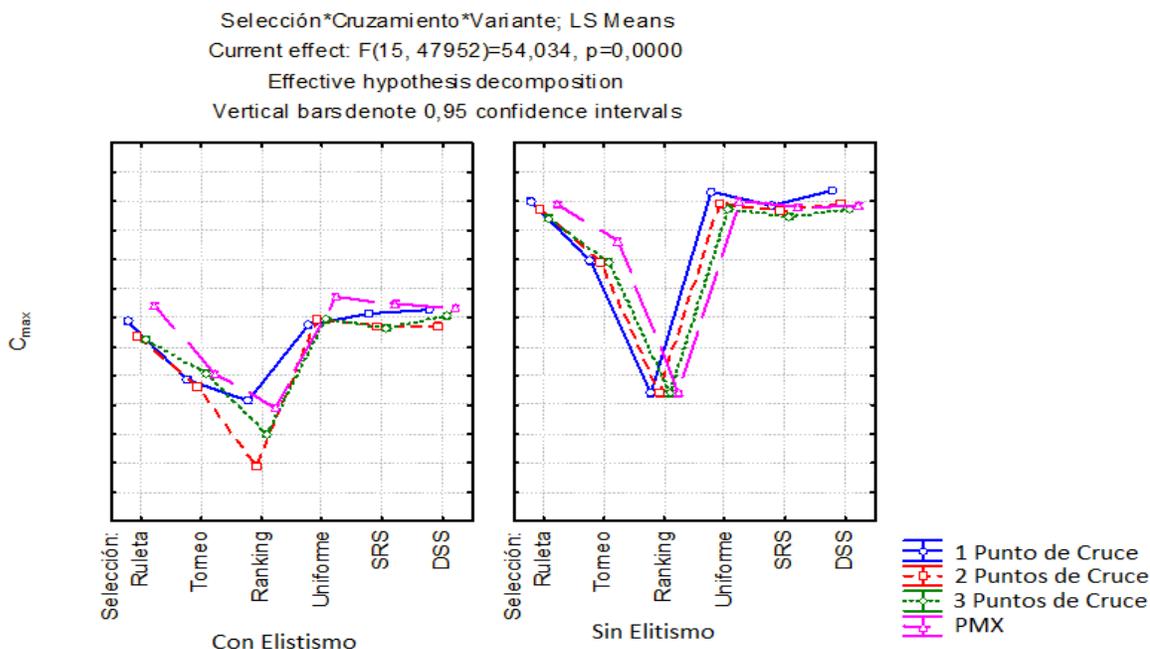


Figura 3. Influencia de los parámetros Selección, Cruzamiento y Elitismo en un AG para problemas de secuenciación.

### Pruebas con otras instancias

La tabla 3 muestra una serie de resultados obtenidos en un estudio experimental para el cual se escogió un conjunto de casos (32 instancias de problemas) propuestos por su autor Erick Taillard<sup>2</sup>, que sirven para comparar los resultados obtenidos con la solución ofrecida al problema en esta investigación. Particularmente, se seleccionaron aleatoriamente instancias conformadas por problemas de pequeño tamaño, mediano y grandes teniendo en cuenta las instancias utilizadas en el análisis estadístico.

Tabla 3. Resultados obtenidos por el AG para problemas de tipo Flow Shop.

	Instancias de problemas	Resultados de Taillard		C <sub>max</sub>		Instancias de problemas	Resultados de Taillard		C <sub>max</sub>
		Lower bound	Upper bound				Lower bound	Upper bound	
20x5	ta001	1232	1278	<b>1278</b>		ta011	1448	1582	<b>1575</b>
	ta002	1290	1359	<b>1329</b>		ta012	1479	1659	<b>1595</b>

<sup>2</sup> Eric Taillard: Profesor de la Universidad de Ciencias Aplicadas de Western Switzerland.

	ta005	1198	1236	<b>1235</b>	20x10	ta017	1388	1484	<b>1484</b>
	ta008	1170	1206	<b>1206</b>		ta020	1356	1591	<b>1586</b>
20x20	ta022	1711	2100	<b>2099</b>	50x5	ta035	2837	2863	<b>2863</b>
	ta024	1810	2223	<b>2223</b>		ta036	2793	2829	<b>2829</b>
	ta027	1875	2273	<b>2273</b>		ta037	2689	2725	<b>2722</b>
	ta029	1840	2237	<b>2237</b>		ta040	2776	2782	<b>2782</b>
50x20	ta052	3424	3715	<b>3715</b>	100x5	ta061	5437	5493	<b>5493</b>
	ta055	3313	3635	<b>3635</b>		ta064	4963	5014	<b>5023</b>
	ta057	3427	3716	<b>3716</b>		ta067	5198	5246	<b>5246</b>
	ta059	3457	3765	<b>3765</b>		ta070	5272	5328	<b>5334</b>
100x10	ta071	5759	5770	<b>5795</b>	200x20	ta102	10947	11420	<b>11480</b>
	ta075	5431	5468	<b>5524</b>		ta105	11132	11311	<b>11396</b>
	ta076	5246	5303	<b>5360</b>		ta106	11085	11282	<b>11364</b>
	ta078	5556	5623	<b>5688</b>		ta109	10965	11343	<b>11406</b>

## Conclusiones

La aplicación de metaheurísticas ha reportado buenos resultados en la solución de problemas de secuenciación de tareas. En particular, en este trabajo se seleccionó Algoritmos Genéticos para solucionar un problema de este tipo proponiéndose varios métodos de cruzamiento y de selección haciendo uso indistintamente del elitismo. Se logró la implementación de cada uno de estos métodos obteniéndose buenas soluciones con cada una de las posibles combinaciones. Las soluciones obtenidas por las diferentes variantes propuestas para el AG fueron comprobadas estadísticamente obteniéndose los resultados siguientes:

- Se adaptaron apropiadamente los valores de los principales parámetros del AG al FSSP determinando estadísticamente que la combinación de parámetros que mayor influye en los resultados del AG aplicados a este problema es: Ranking como método de selección, Dos Puntos de Cruce como método de cruzamiento, haciendo uso del Elitismo.
- La metodología propuesta fue evaluada con una parametrización única para una gran diversidad de instancias de problemas devolviendo resultados de alta calidad comparados con los propuestos por la literatura especializada. Este comportamiento muestra la robustez de la técnica AG en el problema de secuenciamiento de tareas.

- Este método se proyecta como una alternativa interesante en la solución de problemas de alta complejidad matemática, y con el cual se podrían resolver problemas de mayor dificultad que los estudiados en esta investigación.

## Referencias

- ÁLVAREZ, M., Toro, E. y GALLEGO, R. Simulated Annealing Heuristic For Flow Shop Scheduling Problems. *Scientia et Technica*, 2008, XIV (40):159-164.
- ANCAU, M. On Solving Flow Shop Scheduling Problems. *Proceedings of the Romanian Academy*, 2012, 13(1):71-79.
- ČIČKOVÁ, Z. y ŠTEVO, S. Flow Shop Scheduling using Differential Evolution. *Management Information Systems*, 2010, 5(2):008-013.
- CHAUDHRY, I. A. y Munem khan, A. Minimizing makespan for a no-wait flowshop using genetic algorithm. *Sadhana*, 2012, 36(6):695-707.
- EIBEN, Á. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* 2009, 3(2):124-141.
- GARCÍA, S. y HERRERA, F. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 2008, 9:2677-2694.
- GOLDBERG, D. E. e. a. Genetic algorithms, noise and the sizing of populations. *Complex Systems*, 1992, 6:333-362.
- HOLSAPPLE, J. A genetics-based hybrid scheduler for generating static schedules in flexible manufacturing contexts. *IEEE Transactions on Systems, Man and Cybernetics*, 1993, 23(4):953-972.
- JUNGWATTANAKIT, J., REODECHA, M., CHAOVALITWONGSE, P. y WERNER, F. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & Operations Research*, 2009, 36(2):358-378.
- MÁRQUEZ, J. E. Optimización de la programación (scheduling) en Talleres de Mecanizado. Tesis Doctoral. Universidad Politécnica de Madrid. Madrid, España, 2012.187.
- MATÍA, J. Optimización de la secuenciación de tareas en taller mediante algoritmos genéticos. Tesis de Maestría. 2010.134.

- PALOMINOS, P. B. Influencia del tipo de Cruza y Tamaño de Población en la Calidad de la Solución de un Algoritmo Genético: Aplicación al Problema de Asignación de Trabajos a Máquinas en Paralelo. Revista ICHIO, 2010, 1(1):59 – 78.
- PINEDO, M. Scheduling Theory, Algorithms, and Systems. New Jersey. Prentice Hall Inc., Englewood Cliffs, U.S.A. 2008.
- QUAN-Ke, P., M. FATIH, T. y YUN-CHIA, L. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Computers and Operations Research, 2008, 35 (9):2807-2839.
- RAMEZANIAN, R., ARYANEZHAD, M. B. y HEYDAR, M. A Mathematical Programming Model for Flow Shop Scheduling Problems for Considering Just in Time Production. International Journal of Industrial Engineering & Production Research, 2010, 21(2):97-104.
- ŠEDA, M. Mathematical Models of Flow Shop and Job Shop Scheduling Problems. World Academy of Science, Engineering and Technology, 2007, 1(31):122-127.
- SHESKING, D. Handbook of parametric and nonparametric statistical procedures. London. Chapman & Hall. 2006. 1704.
- SIERRA, M. Mejora de algoritmo de búsqueda heurística mediante poda por dominancia. Aplicación a problemas de scheduling. Tesis Doctoral. Universidad de Oviedo. Oviedo, 2009.9-46.
- SUSHIL, J. y ZHIJIE XU, L. Genetic Algorithms for Open Shop Scheduling and Re-Scheduling. Department of Computer Science, 2009: 4.
- TORO, M., RESTREPO, G. y GRANADA, M. Adaptación de la técnica de Particle Swarm al problema de secuenciación de tareas. Scientia et Technica UTP, 2006, XII (32):307-313.
- TORO, M., RESTREPO, G. Y. y GRANADA, E. M. Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal - Flow Shop. Scientia et Technica, 2006b, XII (30):285-290.
- YAMADA, T. Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems. Doctor en Informática. Kyoto University. Kyoto, Japan, 2003. 120.
- YECIT, O., SUESCÚN, B., BRITO, R. y MEJÍA, G. Minimización de la tardanza ponderada total en talleres de manufactura aplicando colonia de hormigas. Revista Científica y Técnica de la Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, 2009, 14(1):25-30.
- ZHANG, H. Applications of Neural Networks in Manufacturing: A state-of-the-art Survey. International Journal of Production Research, 1995, 33(3):705-728.