

# Solución del Problema de Flow Shop Flexible Aplicando el Algoritmo Genético de Chu- Beasley<sup>1</sup>

## Solving the Flexible Flow Shop Problem Using the Chu\_ Beasley's Genetic Algorithm

A. P. Jiménez, C. A. Muñoz, E. M. Toro

Recibido Marzo 4 de 2013 – Aceptado Mayo 30 de 2013

**Resumen** - El problema de secuenciación de tareas en sistemas de producción lineal Flow Shop ha sido un tema de gran importancia en la investigación de operaciones donde se busca establecer la programación óptima de trabajos en máquinas dentro de un proceso de producción en una industria en general. En este trabajo se presenta una metodología para dar solución al problema de Flow Shop Flexible teniendo en cuenta las particularidades del problema y aprovechando las características combinatoriales del mismo. El modelo matemático es resuelto usando Algoritmo Genético de Chu-Beasley. La metodología es validada mediante la evaluación de 9 casos de prueba de la literatura especializada de baja, media y alta complejidad matemática.

**Palabras clave** - Algoritmo Genético de Chu-Beasley, Flow Shop Flexible, makespan, metaheurísticas, optimización combinatorial.

**Abstract** - The tasks scheduling problem on linear production systems Flow Shop has been a topic of great importance in the operations research which seeks to establish optimal job scheduling in machines within a production process in an industry. A methodology for solving the Flexible Flow Shop problem is presented taking into account the features and the combinatorial characteristics of the problem. The mathematical model is solved through using Chu-Beasley's Genetic

Algorithm. The methodology is probed by evaluating nine tests cases of the specialized literature with low, medium and high mathematical complexity.

**Key Words** - Genetic Algorithm Chu-Beasley, Flexible Flow Shop, makespan, metaheuristics, combinatorial optimization.

### I. NOMENCLATURA

AGBC: Algoritmo Genético de Chu-Beasley  
 $C_{max}$ : Makespan

### II. INTRODUCCIÓN

El problema de Flow Shop Flexible al igual que muchos otros en el campo de secuenciación de tareas es de difícil solución. Esta característica hace que el problema sea permutacional, es decir, cada una de las combinatorias que pueden formarse deben contener las  $n$  tareas a procesar y una permutación se diferencia de otra únicamente en el orden de colocación de los elementos, obteniéndose de esta manera que el número de permutaciones posibles es  $n!$ . En la literatura especializada los problemas combinatoriales están catalogados como  $NP$ - difícil, dado que no se cuenta con algoritmos de orden polinomial a partir de los cuales se pueda resolver el problema de forma óptima y en el menor tiempo posible. Se dice que un problema es  $NP$ -difícil cuando se demuestra que cualquier algoritmo de solución tiene un tiempo de ejecución que aumenta, en el peor de los casos, exponencialmente con el tamaño del problema [1].

<sup>1</sup> Producto derivado del proyecto de investigación "Solución del problema de Flow shop flexible aplicando el algoritmo genético de Chu-Beasley", realizado como prerrequisito de grado de pregrado y Desarrollado dentro del grupo de investigación GAOPE (Grupo en aplicaciones en optimización y procesos estocásticos), vinculado a la Facultad de Ingeniería Industrial de la Universidad Tecnológica de Pereira.

A. Jiménez, trabaja en el Grupo Enel Colombia. (correo e.: angelajimenez@ingenieros.com).

C. Muñoz, trabaja en XM, Medellín, Colombia. (correo e.: adrianmunoz@ingenieros.com)

E. Toro, trabaja en la Universidad Tecnológica de Pereira. Facultad de Ingeniería Industrial. (correo e.: elianam@utp.edu.co).

El problema del Flow Shop puede ser estudiado considerando múltiples características y de acuerdo a éstas el problema se clasificará como Flow Shop Flexible, Flow Shop Híbrido, Flow Shop Permutacional, entre otros. El objeto de estudio de esta investigación es el problema de la programación del Flow Shop Flexible ó problema de secuenciación de tareas en líneas de flujo con máquinas en paralelo.

Desde que el problema fue abordado por primera vez, a mediados de los años 50, múltiples investigadores han intentado desarrollar modelos y estrategias para darle solución. Tanto las estrategias como los modelos han evolucionado a través del tiempo, convirtiéndose en la actualidad en un reto para las técnicas computacionales. En la Tabla 1 se muestra el recuento de los estudios realizados desde que Johnson lo planteó por primera vez en 1953 hasta el año 2009.

En este trabajo se plantea una metodología para solucionar el problema de Flow Shop Flexible a través de la aplicación de una técnica metaheurística de optimización con la que se pretende establecer la secuencia de tareas y a la vez optimizar una medida de efectividad que para este caso es minimizar el tiempo total requerido para la terminación de todas las tareas (makespan ó  $C_{max}$ ). El problema fue resuelto empleando el Algoritmo Genético de Chu-Beasley AGBC. Para validar la metodología planteada fueron estudiados 9 casos de prueba de la literatura especializada.

En la Tabla I se muestra el recuento de los estudios realizados desde que Johnson lo planteó por primera vez en 1953 hasta el año 2009.

TABLA I  
Estudios realizados

Año	Autor	Problema	Metodología
[2] 1953	S. M. Johnson	Programación óptima de la producción con dos y tres máquinas y tiempos de ajuste	Heurística
[3] 1995	Colin R. Reeves	Flow-Shop Permutacional	Algoritmo Genético Clásico
[4] 1997	Chandrasekharan Rajendran, Hans Ziegler	Flow-Shop	Heurística
[5] 1998	Christos Koulamas	Flow-Shop	Heurística de Programación de Flow-Shop
[6] 1998	M.C. Portmann, A. Vignier, D. Dardilhac	Flow-Shop Híbrido	Algoritmo Branch and Bound
[7] 1998	Eugeniusz Nowickiy, Czeslaw Smutnicki	Flow-Shop con Máquinas Paralelas	Algoritmo Búsqueda Tabú
[8] 2000	T. Sawik	Flow-Shop Flexible con capacidad finita	Programación Entera Mixta
[9] 2001	Meral Azizoglu, Ergin Cakmak, Suna Kondakci	Flow-Shop Flexible	Algoritmo Branch and Bound
[10] 2001	JiyinLiu, Colin R Reeves	Flow-Shop Permutacional	Heurística Constructiva
[11] 2002	Vincent T'kindt, Nicolas Monmarché, Fabrice Tervinet	Flow-Shop	Colonia de Hormigas
[12] 2002	Edward F. Stafford Jr., Fan T. Tseng	Flow-Shop regular, Flow-Shop sin espera (NIQ), Flow-Shop con tiempos de preparación (SDST), Flow-Shop SDST/NIQ.	Modelos de Programación Lineal Entera Mixta (MILP)
[13] 2003	Xi Sun, Kazuko Morizawa, Hiroyuki Nagasawa	Flow-Shop Montaje-Tipo	Heurísticas de Gran Alcance
[14] 2003	Andreas Fink	Permutación de trabajos procesados secuencialmente en una serie de máquinas	Metaheurísticas
[15] 2004	Bagas Wardono, Yahya Fathi	Flow Shop Flexible con capacidad de buffers limitada entre las etapas	Algoritmo Búsqueda Tabú
[16] 2004	S. Bertel, J.-C. Billaut	Flow-Shop Multiprocesador de tres etapas	Programación Lineal Entera, un Algoritmo Goloso y un Algoritmo Genético
[17] 2004	Chandrasekharan Rajendran, Hans Ziegler	Flow-Shop de Permutación	Algoritmo Colonia de Hormigas
[18] 2004	Mary E. Kurz, Ronald G. Askin	Flow Shop Flexible	Programación Entera
[19] 2004	Can Akkan, SelcukKarabatli	Flow-Shop de n trabajos en dos	Algoritmo Branch and Bound
[20] 2005	Rubén Ruiz, Concepción Maroto	Flow-Shop Permutacional	Evaluación comparativa de 18 heurísticas y 7 Algoritmo de Recocido Simulado
[21] 2005	T.K. Varadharajan, Chandrasekharan Rajendran	Flow-Shop	Algoritmo de Recocido Simulado
[22] 2005	José Elias, Claudio Arroyo, Vinicius Amaral Amentano	Flow-Shop con múltiples criterios de desempeño	Algoritmo Genético
[23] 2005	Rubén Ruiz, Concepción Maroto, Javier Alcaraz	Flow-Shop con secuencia dependiente de los	Algoritmos Genéticos Avanzados
[24] 2006	Rubén Ruiz, Concepción Maroto	Flow-Shop Híbrido	Algoritmo Genético
[25] 2007	M. Fatih Tasgetiren, Yun-Chia Liang, Mehmet Sevkli	Flow-Shop Permutacional	Algoritmo de Optimización Cúmulo de Partículas
[26] 2009	M. Zandieh, E. Mozaffari, M. Chahami	Líneas de Flujo Flexibles Híbridas	Algoritmo Genético

### III. DESCRIPCIÓN DEL PROBLEMA

El problema del Flow Shop Permutacional o problema de secuenciación de tareas de producción lineal consiste en programar de forma óptima un conjunto de  $N$  tareas en  $M$  máquinas, considerando que todas las tareas tienen el mismo orden de procesamiento, los tiempos de ejecución varían según la etapa que sea visitada, no se consideran tiempos de ajuste de las máquinas entre un trabajo y otro, y los trabajos no pueden ser interrumpidos.

De acuerdo al ambiente de las máquinas se puede dar que el problema tenga una sola máquina, máquinas en paralelo, talleres de producción continua, producción intermitente y talleres abiertos [27].

Una de las generalizaciones del problema de Flow Shop es el Flow Shop Flexible (FFS Flexible Flow Shop). Este problema consiste en la programación de  $N$  tareas en  $E$  etapas en serie, donde cada etapa está compuesta por  $M$  máquinas idénticas en paralelo. El número de etapas y el número de máquinas por etapa varían según el tamaño de la empresa y el tamaño del centro de producción respectivamente. La secuencia de los trabajos es la misma para todo el proceso, iniciando en la etapa 1 y finalizando en la etapa  $E$ .

### IV. FORMULACIÓN DEL PROBLEMA

De manera general, la formulación del problema de Flow Shop Flexible se puede plantear de la siguiente manera: el objetivo es la minimización del tiempo total requerido para terminar todas las tareas o makespan ( $C_{max}$ ), de lo cual resulta la secuencia óptima de procesamiento de todas las tareas.

$$f.o. \min C_{max}$$

De acuerdo con las características del problema del Flow Shop Flexible se deben tener en cuenta las siguientes consideraciones, las cuales corresponden a las restricciones del modelo.

§ Las máquinas en cada etapa son idénticas. Cada etapa  $E$  está compuesta por un conjunto de  $ME = \{1, \dots, m_e\}$  máquinas.

§ El conjunto de  $N$  tareas  $N = \{1, \dots, n\}$  tiene que ser procesado en cada etapa  $E$ .  $E = \{1, \dots, e\}$  en sólo una máquina. Una máquina puede procesar sólo un trabajo a la vez.

§ El procesamiento de un trabajo en una máquina no puede ser interrumpido. La secuencia de los trabajos debe ser la misma para todo el proceso. En una misma etapa se pueden procesar varias tareas simultáneamente, pero siempre teniendo en cuenta la secuencia.

### V. METODOLOGÍA DE SOLUCIÓN

#### A. Codificación

La codificación es un factor clave en la solución de cualquier problema la adecuada selección de la codificación puede eliminar muchas de las restricciones planteadas en el modelo matemático y puede hacerse un mejor control sobre la factibilidad, en este caso se hace para evitar la repetición de tareas como se muestra en la Fig. 1.



Fig. 1. Esquema Codificación

#### B. Cálculo de la función objetivo

Como se ha definido anteriormente, existen múltiples formas de medir el desempeño o la efectividad del Flow Shop y del Flow Shop Flexible particularmente. Una de las métricas más utilizadas en la literatura especializada ha sido el Makespan ( $C_{max}$ ) el cual calcula el tiempo total requerido para terminar todas las tareas. Los insumos para el cálculo del makespan o datos de entrada son la matriz de tiempos de procesamiento de cada tarea en cada etapa, la cantidad de máquinas por cada etapa y la secuencia propuesta del orden de ejecución de las tareas para todo el proceso.

El primer paso del proceso del cálculo del makespan consiste en hacer un reordenamiento de la matriz de tiempos de procesamiento de acuerdo con el orden dado por la secuencia propuesta; posteriormente inicializar la matriz de tiempo de terminación la cual tendrá tamaño de número de máquinas por número de tareas [ $M \times T$ ]. Una vez teniendo ordenada la matriz de tiempos y creada la matriz de tiempos de terminación se procede a hacer el cálculo de los tiempos de terminación de cada tarea en cada máquina de cada etapa. Este cálculo inicia con la asignación de cada tarea en cada máquina de la primera etapa y calcular el tiempo que tarda cada tarea para pasar la siguiente etapa. Este proceso se hace iterativamente hasta completar todas las etapas y se debe garantizar siempre que el orden de las tareas sea respetado y que la máquina que vaya a ser asignada ya se encuentre desocupada, por lo cual es necesario verificar si en cada etapa quedan tareas pendientes por ejecutar.

El valor del  $C_{max}$  corresponderá al tiempo de terminación de la última tarea dada por la secuencia en la última máquina de la última etapa, este valor también corresponde al valor mayor de la matriz de tiempos de terminación de las tareas. Para explicar e ilustrar el cálculo del makespan a continuación se plantea un problema de Flow-Shop Flexible, el cual tiene como objetivo la minimización del Makespan. Se asumen 4 tareas que tienen que ser procesadas en 3

etapas. Cada etapa con un número de máquinas y tiempos de procesamiento diferentes, los cuales se presentan mediante la figura 2 y la tabla II.



Fig. 2. Número de máquinas por etapa.

TABLA II  
Tiempos de procesamiento de las tareas en cada una de las etapas

Etapa \ Tarea	1	2	3
1	10	8	5
2	5	10	10
3	5	5	8
4	8	6	8

Asumiendo una secuencia  $S = 3 \ 2 \ 4 \ 1$ . En el tiempo inicial  $t=0$  todas las máquinas están disponibles, por tanto se pueden asignar los trabajos necesarios para operar las máquinas disponibles considerando la secuencia. La etapa 1 cuenta con tres máquinas:  $m_{11}$ ,  $m_{12}$ ,  $m_{13}$  a las cuales se le asignarán las tareas de acuerdo a la secuencia, Ver Fig. 3.

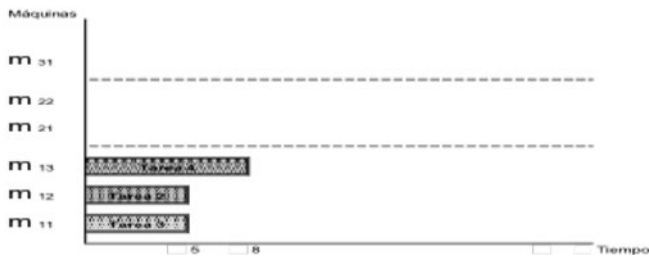


Fig. 3. Fase 1 Cálculo Makespan.

Las tareas 3 y 4 que se encuentran en las máquinas  $m_{11}$  y  $m_{12}$  de la etapa 1 terminan su procesamiento en  $t=5$ , e inmediatamente pasan a la etapa 2, las máquinas se encuentran disponibles, quedando desocupada la máquina  $m_{11}$  en la que se asigna la tarea 1, y la máquina  $m_{12}$ , como se ilustra en la Fig. 4.



FIG. 4. FASE 2 CÁLCULO MAKESPAN

La tarea 3 termina su procesamiento en la máquina  $m_{21}$  de la etapa 2 en  $t=10$  y pasa a la máquina  $m_{31}$  de la etapa 3 y como la máquina  $m_{21}$  queda disponible, la tarea 4 pasa a la etapa 2, como se muestra en la Fig. 5. El proceso sigue su curso como se muestra sucesivamente a través de las figuras 6,7 y 8. Donde, el valor del makespan que se obtiene es de  $C_{max} = 41$

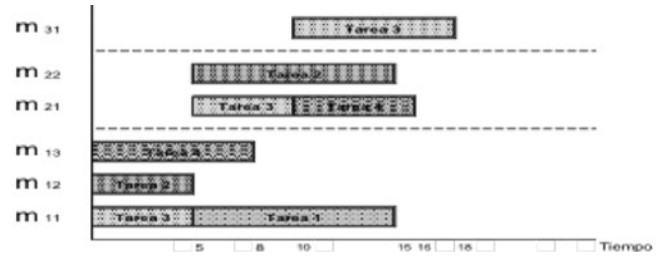


Fig. 5. Fase 3 Cálculo Makespan.

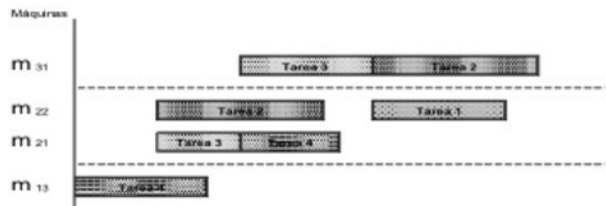


Fig. 6. Fase 4 Cálculo Makespan.

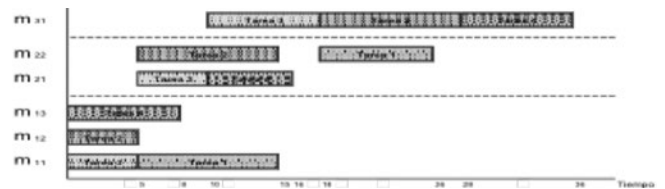


Fig. 7. Fase 5 Cálculo Makespan.

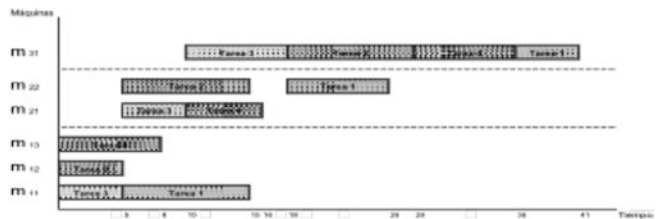


Fig. 8. Fase 6 Cálculo Makespan.

### C. Algoritmo implementado

La estrategia del AGCB para dar solución al problema de Flow Shop Flexible se basa en proponer alternativas de solución y a través del proceso evolutivo ir encontrando soluciones de mejor calidad hasta encontrar la que minimice el makespan. La característica fundamental del AGCB consiste en mantener durante todo el proceso la diversidad de los individuos que conforman la población. En cada

generación solo un individuo es reemplazado, siempre y cuando se cumplan con las condiciones de optimalidad y/o factibilidad establecidas.

Para explicar el algoritmo de la Heurística NEH se plantea un ejemplo cuya construcción se basa en los datos de la Tabla III.

Tabla III  
Tiempos de procesamiento de las tareas

Tarea Máquina	1	2	3
1	1	2	6
2	8	4	2
3	4	5	8

Se calculan los tiempos de procesamiento de las tareas en todas las etapas para saber cuáles toman mayor tiempo: S(1)=13, S(2)=11, S(3)=16.

Luego se ordenan dichas tareas de forma descendiente de tal manera que la tarea con mayor tiempo de procesamiento quede en primer lugar y la de menor tiempo en último lugar: S(3)=16, S(1)=13, S(2)=11

A partir de este vector secuencia de tareas ordenado se toman la primera y segunda tarea, se construyen las secuencias posibles y se evalúa una a una para obtener la de menor  $C_{max}$ .

Secuencia: 1 3 1

Se evalúa la secuencia 1 3 en donde se obtiene un  $C_{max} = 21$  y en la secuencia 3 1, el  $C_{max} = 20$ . La secuencia que continua es la de menor  $C_{max}$ , tal como se muestra en la tabla IV.

Tabla IV  
Evaluación  $C_{max}$  secuencias parciales

Tarea Máquina	1	3
1	0/1	1/7
2	1/9	9/11
3	9/13	13/21

Tarea Máquina	3	1
1	0/6	6/7
2	6/8	8/16
3	8/16	16/20

Habiendo determinado la mejor secuencia parcial se toma la siguiente tarea de la secuencia ordenada y se construyen todas las secuencias posibles como se observa en la Fig. 9.

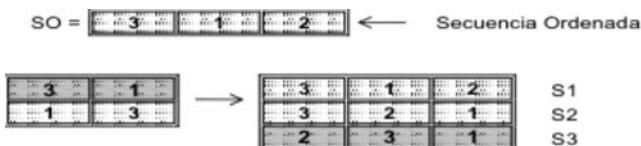


Fig. 9. Secuencias parciales

Calculando el  $C_{max}$  para las secuencias se obtiene:  $C_{max}$  S1= 25;  $C_{max}$  S2= 25;  $C_{max}$  S3= 23

De acuerdo al cálculo anterior, la secuencia seleccionada es la S3. El proceso evolutivo del AGCB se desarrolla de la forma clásica a través de los procesos de selección, recombinación, mutación y una fase de aspiración.

Para el proceso de recombinación se utiliza la recombinación denominado Recombinación Parcialmente Compatible (PMX Partially Matched Crossover). Luego se aplica un criterio de aspiración que consiste en generar un número aleatorio entre 0 y 1 y se compara con una tasa de mutación previamente establecida en los parámetros ajustables del algoritmo. Si el número aleatorio es menor que la tasa de mutación entonces se ejecuta el proceso de mutación el cual se realiza de la forma clásica a través de una metodología de trueque o swap en la cual se seleccionan dos puntos aleatorios (genes) y se cambia uno por el otro, de lo contrario el descendiente pasa sin modificaciones. Luego se mide la diversidad entre un individuo y otro siempre debe ser mayor a dos posiciones.

Si se cumple el criterio de diversidad se chequean básicamente dos criterios: el primero consiste en que si el nuevo individuo tiene una mejor función objetivo que el individuo de peor calidad, éste reemplaza directamente al individuo de peor calidad de la población, sino se compara un valor aleatorio con la tasa de aceptación y si éste es menor se le da la oportunidad de pasar a la siguiente generación, en caso contrario no se le permite entrar en la población y entonces se dice que se pierde un ciclo generacional. La descripción de todo el proceso del AGCB se muestra en la Fig. 10.

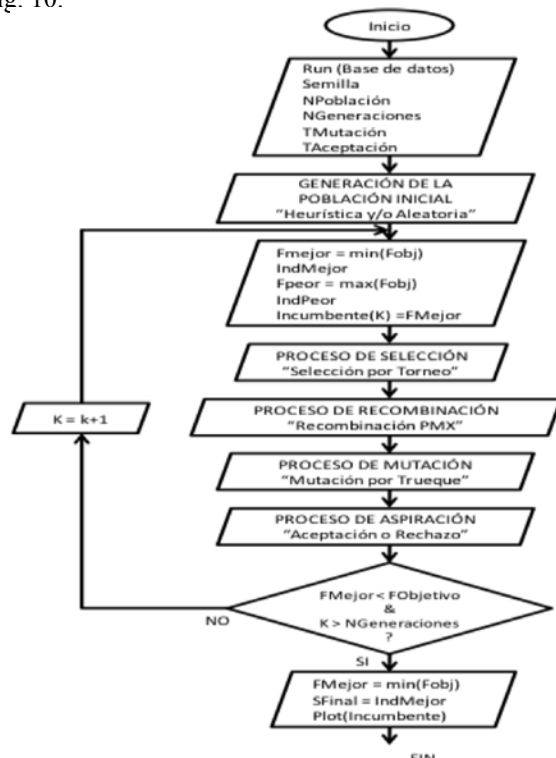


Fig. 10. Diagrama de Flujo AGCB.

## VI. ANÁLISIS DE RESULTADOS

Con el fin de validar la metodología propuesta e implementada es necesario evaluar casos de prueba de la literatura especializada a partir de los cuales se pueda medir tanto la eficiencia computacional como la precisión del algoritmo para obtener los resultados.

Dicha validación se hará enfrentando el algoritmo a problemas de baja, media y alta complejidad refiriéndose esto al tamaño de los casos de prueba.

## A. Casos de Prueba

Los casos de prueba fueron tomados de la página de la Koç University [28] de Estambul, Turquía.

Como se muestra en la Tabla V. los casos de prueba fueron seleccionados de tal forma que se tuviera una gama de problemas con baja, mediana y alta complejidad matemática, donde se tienen problemas con 2, 20 y 100 tarea y con 2, 5 y 8 etapas, a partir de los cuales se pudiera medir la eficiencia computacional del algoritmo implementado.

## B. Resultados obtenidos

La metodología se validó inicialmente con una población generada totalmente aleatoria y posteriormente se realizó la validación inicializando la población utilizando la heurística NEH y completándola de manera aleatoria. En cada uno de los casos se varió la semilla y según el tamaño del problema se ajustó el tamaño de la población. En las tablas VI y VII se muestran los resultados.

De los resultados encontrados se observa que se encuentra la misma función objetivo o  $C_{max}$  en cada uno de los casos de prueba evaluados. Al comparar los resultados obtenidos en el tiempo promedio de ejecución se puede notar que al inicializar la población con la Heurística NEH se obtienen mejores resultados en los tiempos de ejecución, ya que la Heurística genera una secuencia de buena calidad y muy cercana al óptimo del problema, por tanto el tiempo computacional disminuye.

Tabla V  
Casos de prueba

Caso	Tareas	Etapas	Máquinas por Etapa
P5S2T05	5	2	5 1
P20S2T09	20	2	1 5
PH1S2T04	100	2	1 2
P5S5T06	5	5	5 4 3 2 3
P20S5T06	20	5	2 1 5 4 3
PH1S5T05	100	5	4 2 1 5 4
P5S8T02	5	8	3 4 1 2 3 5 4 5
P20S8T09	20	8	1 5 4 2 5 1 5 3
PH1S8T03	100	8	2 1 2 3 2 2 5 4

Tabla VI  
Resultados con población inicial totalmente aleatoria

Caso	Parámetros			Fo Encontrada	T Promedio [s]
	P ini	TM	TA		
P5S2T05	10	0.35	0.10	267	0,30264
P20S2T09	10	0.35	0.10	1288	0,50232
PH1S2T04	20	0.35	0.10	4911	220,53237
P5S5T06	10	0.35	0.10	304	0,33384
P20S5T06	10	0.35	0.10	1170	1,18404
PH1S5T05	20	0.35	0.10	4985	455,24055
P5S8T02	10	0.35	0.10	445	0,30732
P20S8T09	10	0.35	0.10	1493	0,66144
PH1S8T03	20	0.35	0.10	5449	155,28027

Tabla VII  
Resultados con población generada usando neh

Caso	Parámetros			Fo Encontrada	T Promedio [s]
	P ini	TM	TA		
P5S2T05	10	0.35	0.10	267	0,27144
P20S2T09	10	0.35	0.10	1288	0,28704
PH1S2T04	20	0.35	0.10	4911	3,30096
P5S5T06	10	0.35	0.10	304	0,31512
P20S5T06	10	0.35	0.10	1170	0,32916
PH1S5T05	20	0.35	0.10	4985	7,67521
P5S8T02	10	0.35	0.10	445	0,31201
P20S8T09	10	0.35	0.10	1493	0,40248
PH1S8T03	20	0.35	0.10	5449	12,08386

## VII. CONCLUSIONES

Fue resuelto el problema del Flow Shop Flexible teniendo en cuenta como medida de efectividad la optimización del tiempo total requerido para terminar todas las tareas (Makespan  $C_{max}$ ), a través de la implementación de una metodología basada en la técnica Metaheurística AGCB.

Utilizar la heurística NEH en los casos de alta complejidad presenta grandes ventajas, ya que reduce el costo computacional del AGCB al generar una secuencia muy cercana a la solución. Por otro lado, para los casos de baja o mediana complejidad no se justifica la evaluación exhaustiva de la heurística.

Emplear una codificación entera para dar solución al problema del Flow Shop Flexible resulta una alternativa bastante eficiente ya que basados en las experiencias expuestas en la literatura especializada, dadas las características del problema y después de haber analizado las ventajas y desventajas que se presentarían al emplear una u otra codificación (binaria, entera), se encontró que facilita la implementación de los operadores genéticos y además permite un control eficaz de la factibilidad.

El ajuste de los parámetros del Algoritmo Genético de Chu-Beasley no tuvo mayor incidencia en la solución del problema del Flow Shop Flexible, pese a que se evaluaron casos de prueba de baja, media y alta complejidad matemática, ya que los resultados encontrados se obtuvieron con un costo computacional razonable y llegando siempre a la misma función objetivo.

Como trabajo futuro puede plantearse utilizar otra técnica Metaheurística de optimización con el fin de validar y comparar la eficiencia computacional de distintas técnicas

## REFERENCIAS

- [1] R. Gallego, A. Escobar y E. Toro, “*Técnicas Metaheurísticas de Optimización*”, Texto Universitario Universidad Tecnológica de Pereira, Segunda Edición, 2008.
- [2] S. M. Johnson, “*Optimal Two and Three Stage Production Schedules with Setup Times Included*”, The Rand Corporation, pp 402 (1 a t10), 1953.
- [3] C. Reeves, “*A Genetic Algorithm for Flow-Shop Sequencing*”, Computers Ops. Vol. 22, N° 1, pp 5-13, 1995
- [4] C. Rajendran, H Ziegler, “*An Efficient Heuristic for Scheduling in a Flow-Shop to Minimize Total Weighted Flow time of Jobs*”, European Journal of Operational Research 103, pp 129-138, 1997.
- [5] C. Koulamas, “*A New Constructive Heuristic for The Flow-Shop Scheduling Problem*”, European Journal of Operational Research 105, pp 66-71, 1998.
- [6] M.C. Portmann, A. Vignier, D. Dardilhac, D. Dezalay, “*Branch and Bound Crossed With GA to Solve Hybrid Flow-Shops*”, European Journal of Operational Research 107, pp389-400, 1998.
- [7] E. Nowicki, C. Smutnicki, “*The Flow-Shop with Parallel Machines: A Tabu Search Approach*”. European Journal of Operational Research 106, pp226-253, 1998.
- [8] T. Sawik, “*Mixed Integer Programming for Scheduling Flexible Flow Lines with Limited Intermediate Buffers*”, Mathematical and Computer Modelling 31, pp39-52, 2000.
- [9] M. Azizoglu, E. Cakmak, S. Kondakci, “*A Flexible Flow-Shop Problem with Total Flow Time Minimization*”, European Journal of Operational Research 132, pp 528-538, 2001.
- [10] J. Liu, C. Reeves, “*Constructive and Composite Heuristic Solutions to the P//ECi Scheduling Problem*”, European Journal of Operational Research 132, pp 439-452, 2001.
- [11] V. T'kindt, N. Monmarché, F. Tercinet, D. Laugt, “*An Ant Colony Optimization Algorithm to Solve a 2-Machine Bicriteria Flow-Shop Scheduling Problem*”, European Journal of Operational Research 142, pp250-257, 2002.
- [12] E. Stafford., F Tseng, “*Two Models for a Family of Flow-Shop Sequencing Problems*”, European Journal of Operational Research 142, pp 282-293, 2002.
- [13] X. Sun, K. Morizawa, H. Nagasawa, “*Powerful Heuristics to Minimize Makespan in Fixed, 3-Machine, Assembly-type Flow-Shop Scheduling*”, European Journal of Operational Research 146, pp498-516, 2003.
- [14] A. Fink, “*Solving the Continuous Flow-Shop Scheduling Problem by Metaheuristics*”, European Journal of Operational Research 151, pp 400-414, 2003.
- [15] B. Wardono, Y. Fathi, “*A Tabu Search Algorithm for the Multi-Stage Parallel Machine Problem with Limited Buffer Capacities*”, European Journal of Operational Research 155, pp 380-401, 2004.
- [16] S. Bertel, J.-C. Billaut, “*A Genetic Algorithm for an Industrial Multiprocessor Flow Shop Scheduling Problem with Recirculation*”, European Journal of Operational Research 159, pp 651-662, 2004.
- [17] C. Rajendran, H. Ziegler, “*Ant-Colony Algorithms for Permutation Flow-Shop Scheduling to Minimize Makespan/Total Flowtime of Jobs*”, European Journal of Operational Research 155, pp 426-438, 2004.
- [18] M Kurz, R. Askin, “*Scheduling Flexible Flow Lines with Sequence-Dependent Setup Times*”, European Journal of Operational Research 159, pp 66-82, 2004.
- [19] C.Akkan, S. Karabati, “*The Two-Machine Flow-Shop Total Completion Time Problem: Improved Lower Bounds and a Branch-And-Bound Algorithm*”, European Journal of Operational Research 159, pp420-429, 2004.
- [20] R. Ruizy C. Maroto, “*A Comprehensive Review and Evaluation of Permutation Flow-ShopHeuristics*”, European Journal of Operational Research 165, pp479-494, 2005.
- [21] T.K. Varadharajan, C. Rajendran, “*A Multi-Objective Simulated-Annealing Algorithm for Scheduling in Flow-Shops to Minimize the Makespan and Total Flowtime of Jobs*”, European Journal of Operational Research 167, pp 772-795, 2005.
- [22] J. Arroyo, V. Amaral, “*Genetic Local Search for Multi-Objective Flow-Shop Scheduling Problems*”, European Journal of Operational Research 167, pp717-738, 2005.
- [23] R. Ruiz, C. Maroto, J.Alcaraz, “*Solving the Flow-Shop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics*”, European Journal of Operational Research 165, pp34-54, 2005.
- [24] R. Ruiz y C.Maroto, “*A Genetic Algorithm for Hybrid Flow-Shops with Sequence Dependent Setup Times and Machine Eligibility*”. European Journal of Operational Research 169, pp781-800, 2006.
- [25] M. Tasgetiren, Y. Liang, M.Sevkli, G. Gencyilmaz, “*A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flow-Shop Sequencing Problem*”, European Journal of Operational Research 177, pp 1930-1947, 2007.
- [26] M. Zandieh, E. Mozaffari, M. Gholami, “*A Robust Genetic Algorithm for Scheduling Realistic Hybrid Flexible Flow Line Problems*”, J IntellManuf, pp 731-743, 2009.
- [27] D. Sipper, y B. Robert, “*Planeación y Control de la Producción*”, México, McGraw Hill. 1998.
- [28] <http://portal.ku.edu.tr/~coguz/research.htm> Koç University

**Ángela Patricia Jiménez Morales.** Ingeniera Industrial graduada de la Universidad Tecnológica de Pereira, Risaralda, Colombia 2012. Actualmente Profesional de Recursos Humanos de Global Service Function para Colombia y Perú en el Grupo Enel, Bogotá, Colombia.

**Cesar Adrián Muñoz Buitrago.** Ingeniero Electricista graduado de la Universidad Tecnológica de Pereira, Risaralda, Colombia 2009. Es estudiante de la Maestría en Ingeniería Eléctrica de la Universidad Tecnológica de Pereira. Actualmente Analista Coordinación Operación en XM, Medellín, Colombia.

**Eliana Mirledy Toro Ocampo.** Ingeniera Industrial graduada de la Universidad Tecnológica de Pereira, 1994. Magister en Ingeniería Eléctrica en el área de Optimización de la Universidad Tecnológica de Pereira, Risaralda, Colombia, 2006. Magister en Investigación de Operaciones y Estadística de la Universidad Tecnológica de Pereira, Risaralda, Colombia, 2008. Actualmente docente de la Facultad de Ingeniería Industrial de la Universidad Tecnológica de Pereira, Risaralda, Colombia.