Revista de Matemática: Teoría y Aplicaciones 2015 **22**(1) : 113–134
cimpa – ucr issn: 1409-2433 (Print), 2215-3373 (Online)

METODOLOGÍA SELECTIVA DE DINÁMICA POBLACIONAL PARA OPTIMIZAR UN AMBIENTE MULTIOBJETIVO DE PRODUCCIÓN JOB SHOP

SELECTIVE METHODOLOGY OF POPULATION DYNAMICS FOR OPTIMIZING A MULTIOBJECTIVE ENVIRONMENT OF JOB SHOP PRODUCTION

Santiago Ruiz* Omar Castrillón† William Sarache‡

Received: 29/Nov/2013; Revised: 18/Aug/2014; Accepted: 25/Aug/2014

---

*Universidad Nacional de Colombia—Sede Manizales—Facultad de Ingeniería y Arquitectura—Departamento de Ingeniería Industrial—GTA en Innovación y Desarrollo Tecnológico, Campus la Nubia—Manizales—Código Postal 170001, Colombia. E-mail: sruizhe@unal.edu.co

†Misma dirección que/*same address as* S. Ruiz. E-mail: odcastrillong@unal.edu.co

‡Misma dirección que/*same address as* S. Ruiz. E-mail: wasarachec@unal.edu.co

Resumen

El presente artículo desarrolla una metodología basada en genética poblacional que permite mejorar el desempeño de dos o más variables en un sistema de producción *job shop*. La metodología aplica un algoritmo genético con características especiales en la selección de individuos que pasan de generación en generación. Los resultados permitieron demostrar mejores desempeños de la metodología propuesta en las variables *makespan*, tiempo muerto y costo de energía al ser comparada con el método FIFO. Al comparar la metodología con el método NSGA II no se obtuvieron diferencias en las variables *makespan* y tiempo muerto; sin embargo, se obtuvo un mejor desempeño en el costo de la energía y, principalmente, mayor eficiencia en relación al número de iteraciones realizadas para obtener el *makespan* óptimo.

Palabras clave: algoritmo genético; job shop; multiobjetivo; subpoblaciones; recursos energéticos; makespan; dinámica de poblaciones.

Abstract

This paper develops a methodology based on population genetics to improve the performance of two or more variables in job shop production systems. The methodology applies a genetic algorithm with special features in the individual selection when they pass from generation to generation. In comparison with the FIFO method, the proposed methodology showed better results in the variables makespan, idle time and energy cost. When compared with NSGA II, the methodology did not showed relevant differences in makespan and idle time; however better performance was obtained in energy cost and, especially, in the number of required iterations to get the optimal makespan.

Keywords: genetic algorithm; job; multiobjective; subpopulations; energy resources; makespan; population dynamics.

Mathematics Subject Classification: 91B72, 62J99, 62J05.

1 Introducción

La configuración en *job shop* pertenece al grupo de sistemas de producción que responden a una estrategia de flujo flexible. En este sistema, las máquinas se organizan por función; es decir, agrupando aquellas del mismo tipo con el fin de maximizar su utilización. De esta manera, el problema de la programación de pedidos consiste en definir la secuencia

de fabricación más favorable para un grupo de productos con flujos de fabricación variados y tamaños de lote distintos [5, 20]. La característica especial en el problema de programación de la producción, está en la secuencia de fabricación distinta que sigue cada pedido que entra al sistema, tal y como se ilustra en la Figura 1. Esta ilustra la ejecución de 4 pedidos P en 3 máquinas M. El problema principal es distribuir, en una franja de tiempo, el desarrollo de los pedidos de tal manera que una máquina no esté ocupada ejecutando más de un pedido a la vez y todos ellos se procesen en el menor tiempo posible.

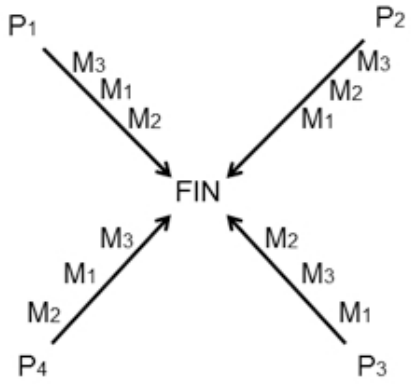

Figura 1: Grafo del *job shop*$_4 \times 3$.

Hasta aquí, encontrar el menor tiempo posible para el conjunto de pedidos (*makespan*) es el objetivo que mayormente se persigue en la programación de un sistema *job shop*; sin embargo, cuando se exige agregar uno o más objetivos, es necesaria una solución multiobjetivo. Para Frutos y Tohmé [14], entre los métodos multiobjetivo más recurrentes en la literatura, basados en algoritmos genéticos aplicados a problemas de programación de la producción en sistemas *job shop*, están el NSGA II (*Non-dominated Sorting Genetic Alghorithm II*), SPEA II (*Strength Pareto Evolutionary Algorithm II*) y sus antecesores NSGA y SPEA. Para dichos autores, el NSGA II es el que ha reportado mayor eficiencia. Dicho método parte de una población inicial y selecciona los individuos ubicados en una frontera de Pareto o cerca de ella, donde cada uno de estos busca satisfacer el conjunto de objetivos planteados de forma simultánea. Estos individuos se mezclan en un proceso iterativo para establecer fronteras de Pareto más eficientes.

La metodología multiobjetivo planteada en este artículo, al igual que los métodos anteriores, se soporta en un algoritmo genético [3, 16, 17, 25]. No obstante, su contribución metodológica fundamental se basa en la incorporación del concepto de dinámica poblacional a partir de la cual la selección se concentra en subgrupos de mejores individuos para satisfacer cada objetivo de forma individual, en subgrupos que satisfagan simultáneamente dos o más objetivos y, adicionalmente, en subgrupos que aleatoriamente ingresan a la población como si fueran inmigrantes. La función multiobjetivo de la metodología entrega soluciones factibles, ya que computa individuos que arrojan un tiempo de proceso posterior al tiempo de inicio, **los demás individuos, no factibles, son descartados**. Es importante resaltar que los objetivos son independientes entre sí, es decir, no son contradictorios, pues un objetivo no está en función de otro.

La dinámica poblacional ha sido estudiada y aplicada en otras áreas del conocimiento tales como el desempeño de poblaciones evolutivas [12], la biología [24] y la genética [2, 23], entre otras.

Según el concepto de dinámica poblacional, el espacio donde se desarrolla una población presenta particularidades que promueven la existencia de individuos con características similares. Si existen varios espacios, con diferentes características, las poblaciones que habitan en ellos son heterogéneas al compararse entre sí. El manejo y conservación de estos espacios y las mezclas que se puedan dar entre individuos son las que permiten obtener mejores desempeños en la población [21], dado que aumenta la diversidad genética [1].

Mediante la diversidad genética se establecen las bases para la caracterización de genotipos diferentes y se promueve el mejoramiento de la selección de combinaciones mejores y favorables a los cruzamientos [9]. El problema de la diversidad en las poblaciones ha sido estudiado desde varias perspectivas; por ejemplo, el estudio de Toro [28] abordó el papel de las razas y los fenómenos históricos como medio para generar gran diversidad de tipos sociales, carácteres y costumbres. Por su parte, Hernández y Esparza [18] en un estudio de los apellidos en diferentes poblaciones, abordaron el problema de la falta de diversidad como causa de la redundancia en una población.

La redundancia, a su vez, genera insuficiente divergencia genética para la generación de filiales, lo cual disminuye la posibilidad de encontrar genotipos con combinaciones genéticas prósperas dada su poca variación [10]. A dicho concepto le subyace un principio básico que establece que las

poblaciones divergentes y heterogéneas tienden a generar mejores alternativas y resultados. Este principio, al ser incorporado en la metodología propuesta, se establece como una contribución en la solución del problema de programación de la producción en sistemas *job shop* (*Job Shop Scheduling Problem*, JSSP).

Para su presentación, el contenido del artículo se ha organizado de la siguiente manera: en la primera sección se expone la metodología propuesta, con su estructura, variables y el proceso de iteración con el algoritmo de selección. En la segunda sección, se aplicó la metodología en un caso de estudio para analizar y comparar los resultados con los métodos FIFO y NSGA II. Se demostró que la metodología propuesta supera al método FIFO en las variables *makespan* (22.52%), tiempo muerto (41.92%) y costo de energía (2.29%). Aunque no se observaron resultados superiores al ser comparada con el método NSGA II en las variables *makespan* y tiempo muerto, sí se obtuvo alguna mejora en el costo de energía y, ante todo, una mayor eficiencia en relación al número de iteraciones necesarias para llegar al *makespan* óptimo. Finalmente, en la última sección se expone un cuerpo de conclusiones y se sugieren algunas líneas generales de trabajo para abordar nuevos casos de estudio.

2 Metodología

La metodología propuesta se compone de 7 pasos, tal y como se muestra en la Figura 2. Esta aporta una solución para dos o más objetivos: en este caso se tratan dos objetivos: 1) reducir el tiempo de procesamiento y 2) reducir el costo de los recursos usados. En la medida en que se agregan más objetivos, estos se orientan a reducir costos de otros recursos diferentes a los anteriores. Una explicación de cada uno de los pasos propuestos se expone a continuación:

Paso1. Plantemiento del problema JSSP

El sistema de producción *job shop* se establece para un número n de pedidos P o solicitudes del cliente y un número m de máquinas M o puestos de trabajo. Con base en los aportes de Cheng y Smith [6] y Castrillón et al. [4] para la programación de sistemas de producción *job shop*, se determinaron las siguientes matrices:

a) Tiempo de proceso

Para ello se determinan dos matrices: la primera matriz (Tabla 1) presenta en cada columna la secuencia de operaciones ($Po(i,j)$, $Po(l,j)$, ..., $Po(k,j)$) en la cual se ejecuta el pedido j ($j = 1, 2, \ldots, n$) en las diferentes máquinas, donde $i_j, l_j, k_j \in \{1, 2, \ldots, n\}$ y $i_j \neq l_j \neq k_j$. Las columnas de los pedidos difieren algunas de otras por la característica propia del sistema *job shop*; es decir, no todas las secuencias son iguales. El conjunto de elementos $Po(i,j), Po(l,j), \ldots, Po(k,j)$ para el pedido j tiene un orden que lo imponen las características del trabajo a realizar.

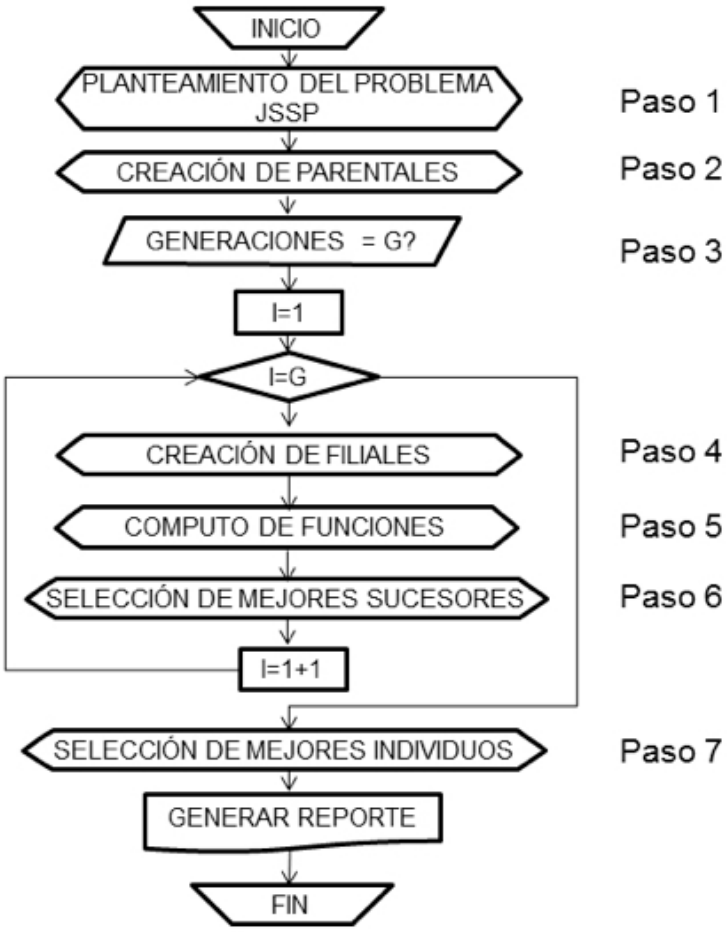

Figura 2: Flujograma de la metodología propuesta.

Tabla 1: Secuencia de pedidos en máquinas.

Pedidos	$\mathbf{P_1}$	$\mathbf{P_2}$	...	$\mathbf{P_n}$
Secuencia de máquinas	$Po(i,1)$	$Po(i,2)$	...	$Po(i,n)$
	$Po(l,1)$	$Po(l,2)$	...	$Po(l,n)$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$Po(k,1)$	$Po(k,2)$	...	$Po(k,n)$

La segunda matriz (Tabla 2), presenta en cada columna $(Tp(1,j),\ldots,Tp(w,j))$, el tiempo que tarda el pedido j en cada máquina M_w, $(w = 1,2,3,\ldots,m)$.

Tabla 2: Tiempos de proceso.

Máquinas	Pedidos			
	$\mathbf{P_1}$	$\mathbf{P_2}$	...	$\mathbf{P_n}$
M_1	$Tp(1,1)$	$Tp(1,2)$	...	$Tp(1,n)$
M_2	$Tp(2,1)$	$Tp(2,2)$	...	$Tp(2,n)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
M_w	$Tp(w,1)$	$Tp(w,2)$	...	$Tp(w,n)$

Aplicando las matrices de las Tablas 1 y 2 en un sistema *Job Shop* real, se obtiene información simplificada para programar las máquinas y los recursos con el fin de lograr el primer objetivo de la programación de un sistema *job shop*: procesar los pedidos en el menor tiempo posible (*makespan*) [13]. La ecuación 1 expresa una función *fitness* (aptitud), la cual permite encontrar la secuencia de fabricación que logra los tiempos más bajos de fabricación. La sumatoria de T_{p_i} representa el tiempo utilizado en el total de las máquinas (w) y la sumatoria T_{o_i} representa el tiempo total de ocio en el total de las máquinas. La función se aplica a un individuo el cual será explicado en el paso 2 "Creación de parentales",

el individuo es el cromosoma del algoritmo genético que programa la producción. Finalmente, la ecuación del *makespan* arroja el tiempo en que termina de trabajar la máquina que completa todos los pedidos:

$$Fitness_{Makespan} = F\left(\frac{\sum_{i=1}^{w} Tp_i + \sum_{i=1}^{w} To_i}{w}\right). \tag{1}$$

b) Minimizar el costo del recurso consumido

El segundo objetivo consiste en minimizar el costo de un recurso consumido o la probabilidad de generar pérdida, como minimizar las averías, accidentes o productos no conformes. Para ello se determinan dos matrices: la primera presenta la cantidad de recurso requerido R por cada pedido en cada máquina, según se muestra en la Tabla 3. En ella, cada columna $(Rp(1,j),\ldots,Rp(m,j))$ define el recurso R requerido por el pedido j en cada máquina M.

Tabla 3: Consumo de recurso R.

Máquinas	Pedidos			
	$\mathbf{P_1}$	$\mathbf{P_2}$	...	$\mathbf{P_n}$
M_1	$Rp(1,1)$	$Rp(1,2)$	...	$Rp(1,n)$
M_2	$Rp(2,1)$	$Rp(2,2)$	...	$Rp(2,n)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
M_m	$Rp(m,1)$	$Rp(m,2)$	...	$Rp(m,n)$

La segunda, contiene el costo del recurso en un tiempo determinado según se expone en la Tabla 4; en ella, la columna "tiempo" representa el número de periodos en una jornada laboral, donde C representa el tiempo máximo de la jornada laboral. La columna "Costo de unidad de recurso" tiene el costo $Cr_{(i)}$ por cada unidad de recurso para cada periodo $i(i = 1, 2, \ldots, c)$.

La ecuación (2) expresa una función *Costo total*, la cual permite encontrar el costo mínimo de los pedidos asignados en las máquinas; la variable i representa el momento en el tiempo en que se determina el costo; la variable j es la máquina (w); la variable $A = (0, 1)$ representa la actividad de la máquina, 0 si está en tiempo de ocio y 1 si está asignada a un pedido; la variable Rp es la cantidad de recurso del pedido consumido en la

Tabla 4: Costo de recurso por unidad por tiempo.

Tiempo	Costo de unidad de recurso
1	$Cr_{(1)}$
2	$Cr_{(2)}$
$\vdots$	$\vdots$
C	$Cr_{(c)}$

máquina; la variable Cr es el costo del recurso en el tiempo (i); el tiempo (i) se determina con la ecuación (3); la variable C es tiempo máximo de una jornada laboral:

$$Costo\ total = \sum_{i=1}^{makespan} \sum_{j=1}^{w} A \times Cr \times Rp \tag{2}$$

$$Tiempo(i) = i - \left(Entero\left(\frac{i-1}{C}\right) \times C \right). \tag{3}$$

Un ejemplo de la aplicación de la ecuación (3) es el siguiente. Para $i = 17$ y $C = 8$ se tiene: el tiempo máximo de la jornada es 8 y el tiempo (i) sería 1.

Paso 2: Creación de parentales

Cada individuo requiere una codificación para ser computado en el algoritmo genético que programa la producción. La codificación del cromosoma que se plantea en esta metodología se puede apreciar en la primera fila de la Tabla 5, la cual representa una solución sencilla a la programación en un sistema *job shop* de n pedidos y m máquinas ($\text{JSSP}_{n\times m}$); la segunda fila representa el pedido asignado y la tercera representa la posición de la máquina asignada. En esta última fila se hace referencia a la nomenclatura de la Tabla 1.

Si se tiene un $\text{JSSP}_{3\times 4}$, (3 pedidos y 4 máquinas), el cromosoma debe tener 12 genes. Cada gen, en su orden de izquierda a derecha, representa la asignación de un pedido en una máquina de acuerdo con los tiempos prestablecidos. Esta etapa finaliza creando una población inicial de N individuos; cada uno representa un reordenamiento aleatorio de la fila "Cromosoma" de la Tabla 5, la cual es el vector x representado en la ecuación (1). Al conjunto de N individuos se les llama "Parentales".

Tabla 5: Codificación del cromosoma para $\text{JSSP}_{n\times m}$.

Cromosoma	1	2	...	...	...	...	...	...	...	$n \times m$
Asignación de pedido	1_1	1_2	...	1_M	2_1	2_2	...	2_M	...	n_M
Asignación de la posición de la máquina	$Po(1,1)$	$Po(2,1)$	...	$Po(m,1)$	$Po(1,2)$	$Po(2,2)$	...	$Po(m,2)$	...	$Po(m,n)$

Paso 3: Generaciones

Los individuos mejoran su desempeño en la medida en que se adaptan al medio al pasar de población en población (*Fitness*). Entre más generaciones se establezcan, existe una mayor probabilidad de obtener individuos superiores. Dicha condición se da siempre y cuando en la población se conserve una característica de heterogeneidad, es decir que no se genere redundancia. En este paso se ingresa el número G de generaciones, que finalmente determinará la iteración de los pasos siguientes. El criterio para determinar el número G se establece de acuerdo con el tiempo de procesamiento y el tiempo en que se logre una solución significativa para quien aplique la metodología.

Paso 4: Creación de filiales

La población de N individuos (N es un número par) se convierte en $2N$ al agregar individuos "filiales" que resultan de cruzar los "parentales" y de aplicar una mutación previamente establecida. El proceso de cruzamiento se basa en los aportes de Díaz et ál. [8]. Este proceso se hace mediante una explotación de recombinaciones (*Partially Matched Crossover*, PMX), el cual consiste en tres etapas:

1. Elegir aleatoriamente dos puntos de cruce en dos cromosomas parentales.

2. El segmento comprendido entre los puntos de cruce en un parental se escoge para crear una sección parcial de un hijo.

3. Los espacios vacíos del hijo obtenido se llenan de izquierda a derecha con los genes del otro parental. Los genes del otro parental se van eligiendo de izquierda a derecha, siempre y cuando no se repitan

en los genes del nuevo hijo. Por ejemplo, si se tienen dos parentales así: $(3, 6, 5, 2, 4, 1)$ y $(1, 2, 3, 4, 5, 6)$, se escoge dos puntos de cruce, aleatoriamente se toman las posiciones del 3 y 4, el primer hijo será $(, , 5, 2, ,)$ y el segundo $(, , 3, 4, ,)$. Con el fin de completar las posiciones vacías del primer hijo, del segundo parental se escogen los genes faltantes, finalmente el primer hijo queda $(1, 3, 5, 2, 4, 6)$ (Figura 3).

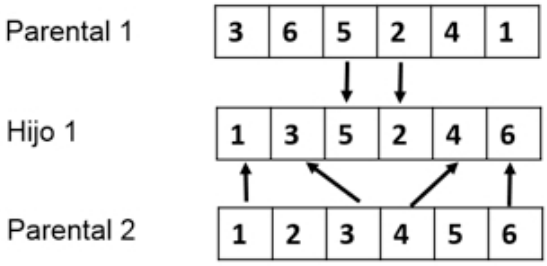

Figura 3: Creación de un hijo (filial).

El segundo hijo queda así: $(6, 5, 3, 4, 2, 1)$.

Paso 5: Cómputo de funciones

Los objetivos previamente definidos son el *makespan* (Ec. 1) y el costo de energía (Ec. 2). Por cada cromosoma se hace un cómputo de funciones, el cual consiste en tres etapas: en la primera se asignan las operaciones de los pedidos en un diagrama de Gantt, haciendo uso de las Tablas 1 y 2 como se aprecia en la Figura 4; en el eje vertical se exponen las máquinas y en el eje horizontal el tiempo. La segunda etapa establece el *makespan*, que es el tiempo en el cual termina de trabajar la última máquina para completar el total de los pedidos. Finalmente en la tercera etapa, haciendo uso de las Tablas 3 y 4 y el diagrama de Gantt creado en la primera etapa, se establece el costo total de la energía utilizada.

Paso 6: Selección de mejores sucesores

Para todos los individuos de la población $2N$, se obtiene un resultado en cada una de las funciones objetivo (*makespan* y costo de energía). Con el fin de pasar de generación en generación, es necesario utilizar un artificio de los AGs, el cual es representado en la selección de los mejores individuos, para finalmente obtener una población N que pasa a la siguiente iteración.

Como el principio hipotético establecido anteriormente determina que las poblaciones divergentes y heterogéneas, tienden a generar mejores alternativas y resultados, este se aplica al seleccionar N individuos de la población *2N*. Esta selección se hace con base en varios grupos llamados subpoblaciones (SPs). Una forma de lograr la heterogeneidad en el caso de estudio es obteniendo subpoblaciones así: una en la cual los individuos se desempeñen bien en el primer objetivo, otra en el segundo objetivo, otra en los dos objetivos, y una última con individuos aleatorios, esta última representa individuos que inmigran. En general si la metodología se aplica en un número B de objetivos $(O_1, O_2, \ldots, O_B)$, el número total de subpoblaciones (SPs) es $B + 2$, donde la SP_1 *se desempeña bien en el primer objetivo*, SP_2 *se desempeña bien en el segundo objetivo, así hasta el* SP_B, SP_{B+1} contiene individuos que se desempeñan bien en todos los objetivos y SP_{B+2} contiene individuos que se desempeñan de cualquier forma, simulando una inmigración. El número de individuos por cada SP_i se establece previamente por parte de quien aplica la metodología para asegurar la heterogeneidad, este criterio arbitrario se elige en la medida en que se obtienen mejores resultados. Una forma de representar la estructura anterior se puede visualizar en la Tabla 6.

Tabla 6: Selección de mejores sucesores.

	Mejores sucesores					Total
Individuos	$I_1 I_2 \ldots I_{SP1}$	$I_1 I_2 \ldots I_{SP2}$	$\ldots$	$I_1 I_2 \ldots I_{SP(B+1)}$	$I_1 I_2 \ldots I_{SP(B+2)}$	N
Subpoblación	Sp_1	Sp_2	$\ldots$	Sp_{B+1}	Sp_{B+2}	$B+2$
Buenos en el objetivo	O_1	O_2	$\ldots$	$O_1, O_2, \ldots, O_B$	Indiferentes (Aleatorios)	

Una aproximación a este tipo de estructura, se desarrolla en los aportes de Herrera et ál. [26], el cual aborda el problema de programación de la producción en un sistema *job shop*, seleccionando un método similar al de subpoblaciones, pero sin incluir la subpoblación de aleatorios.

Paso 7: Selección de mejores individuos

Al llegar a la generación G la población tiene N individuos que finalmente se convierten en los mejores individuos seleccionados; es decir los de mejor *fitness*. Esta población tendrá individuos y cromosomas, que servirán para programar la producción del sistema *job shop*, algunos con buen desempeño en un objetivo y otros con buen desempeño en todos los objetivos que se pretenden conseguir. Con todos los individuos de la población final

se presenta un reporte que servirá para que la organización, la empresa o la industria, que lo aplique, tome decisiones pertinentes de acuerdo a sus necesidades o prioridades.

3 Experimentación

Se aplicaron los siete pasos de la metodología: en el primer paso se ingresaron las restricciones que se exponen más adelante en el caso de estudio. El paso dos genera aleatoriamente 10 cromosomas (N individuos) que se convierten en la primera población de parentales. En el tercer paso, (con base en la experimentación) se eligieron 1000 iteraciones. En el cuarto paso mediante el cruce de cromosomas parentales se generan 10 filiales (hijos). En el quinto paso, al tener una población de 20 individuos (cromosomas), por cada individuo se calcula el diagrama de Gantt y el resultado en las dos funciones objetivo (*makespan* y costo de energía). En el sexto paso, se seleccionan los mejores 10 individuos mediante subpoblaciones; estos pasarán a la siguiente generación como los parentales. Los pasos 4, 5 y 6 se repiten hasta completar las 1000 iteraciones. Finalmente en el paso 7, se exponen los 10 mejores individuos que resultaron de las 1000 iteraciones con los resultados que se presentan más abajo.

3.1 Caso de Estudio

El modelo planteado para la aplicación de la metodología se basó en 6 máquinas y 6 pedidos. Para este caso, la metodología se aplicó con base en dos objetivos: *makespan* y costo de recurso energético consumido. La secuencia y los tiempos se extraen del modelo FT06 de Muth y Thompson [11] (Ver Tablas 7 y 8).

Tabla 7: Secuencias para el caso $\text{JSSP}_{6\times6}$.

Pedidos	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{P_3}$	$\mathbf{P_4}$	$\mathbf{P_5}$	$\mathbf{P_6}$
Secuencia de centros de trabajo	3	2	3	2	3	2
	1	3	4	1	2	4
	2	5	6	3	5	6
	4	6	1	4	6	1
	6	1	2	5	1	5
	5	4	5	6	4	3

Tabla 8: Tiempos de los pedidos en los centros de trabajo (Ms) para el caso JSSP $_{6\times6}$.

Máquinas	Trabajos					
	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{P_3}$	$\mathbf{P_4}$	$\mathbf{P_5}$	$\mathbf{P_6}$
M_1	3	10	9	5	3	10
M_2	6	8	1	5	3	3
M_3	1	5	5	5	9	1
M_4	7	4	4	3	1	3
M_5	6	10	7	8	5	4
M_6	3	10	8	9	4	9

En la Tabla 9 se presenta el consumo del recurso energético para cada pedido en cada máquina.

Tabla 9: Consumo del recurso energético para JSSP$_{6\times6}$.

Máquinas	Pedidos					
	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{P_3}$	$\mathbf{P_4}$	$\mathbf{P_5}$	$\mathbf{P_6}$
M_1	5.1	19.0	7.5	2.3	1.6	25.7
M_2	9.0	26.6	24.8	15.9	6.2	1.2
M_3	24.4	9.1	3.6	9.7	21.3	1.0
M_4	16.2	21.1	15.2	9.2	26.2	18.8
M_5	18.8	27.6	9.7	23.1	7.6	21.1
M_6	5.6	8.5	11.5	10.5	9.7	24.8

En la Tabla 10 se presenta el costo por unidad de recurso energético para cada hora del día, en una jornada de 9 horas.

El cromosoma con el cual se codifica la solución del JSSP$_{6\times6}$ de 36 genes (véase primera fila de la Tabla 9), resulta de multiplicar los centros de trabajo por los pedidos. El sistema *Job Shop* que se está resolviendo, tiene una cantidad de combinaciones de aproximadamente 3.71993×10^{41}.

3.2 Resultados

Para efectos de valoración de la aplicación de la metodología, primero se procedió a realizar una primera programación usando la secuencia tradicional FIFO, con los resultados que se exponen en la Figura 4. En esta se aplica la primera fila de la Tabla 9.

Tabla 10: Costo de unidad del recurso energético por hora del día.

Hora del día	Costo de unidad de recurso
10-11	0.5785714
11-12	0.5785714
12-13	0.3119027
13-14	0.3119027
14-15	0.3119027
15-16	0.3119027
16-17	0.3119027
17-18	0.5785714
18-19	0.5785714

Seguidamente y con el fin de comparar un algoritmo convencional con la metodología propuesta, se aplicó el algoritmo NSGA II. Utilizando el software MATLAB, para 1.000 iteraciones, 10 individuos parentales y una mutación del 30%, se obtienen los datos de la Tabla 11. La función 1 es el resultado del *makespan* y la función 2 es el costo total de energía consumida.

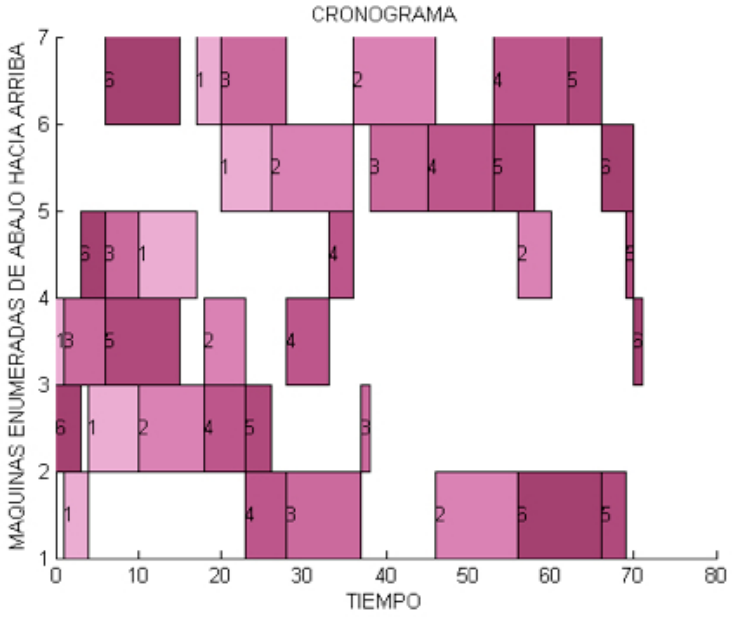

Figura 4: Secuencia tradicional FIFO. *Makespan* = 71; tiempo muerto = 229.

Tabla 11: Soluciones con NSGA II

Individuo	Función 1	Función 2
1	55	1273,6
2	**55**	**1273.0**
3	56	1249.8
4	57	1244.3
5	57	1244.3
6	59	1231.8
7	59	1231.8
8	61	1228.6
9	62	1213.2
10	62	**1213.2**

Bajo los mismos parámetros anteriores, utilizando la metodología propuesta de subpoblaciones y dividiendo el número de individuos parentales (1 para el primer objetivo, 4 para el segundo objetivo, 4 para los dos objetivos y 1 indiferente), se obtienen los resultados de la Tabla 12.

Al comparar los resultados de la metodología SPs con la NSGA-II se determina que el mínimo en la primera función es igual (55) para ambas. En la segunda función la metodología propuesta da menor que la NSGA-II.

Tabla 12: Soluciones obtenidas con la metodología propuesta.

Individuo	Función 1	Función 2
1	55	1284.0
2	70	**1207.7**
3	70	1207.7
4	70	1207.7
5	70	1207.7
6	**55**	**1271.1**
7	55	1271.1
8	5	1271.1
9	55	1271.1
10	55	**1284.0**

En la Figura 5 se presenta el diagrama de Gantt con la secuencia del individuo 6.

El promedio de los resultados en ambas funciones para ambas metodologías muestra un promedio menor en la NSGA-II, esto no es significativo,

lo relevante son los mínimos encontrados, el tiempo en las dos metodologías es 55, el costo de la energía en SPs es 1207.7, mientras que en NSGA-II es 1213.2. Por otro lado, el rango en los resultados de SPs es más amplio que en NSGA-II, lo que presenta un mayor espacio de búsqueda en el SPs comparándolo con la otra metodología. Entre mayor sea el espacio de búsqueda aumenta la diversidad y por ende aumenta la exploración del espacio de soluciones [22].

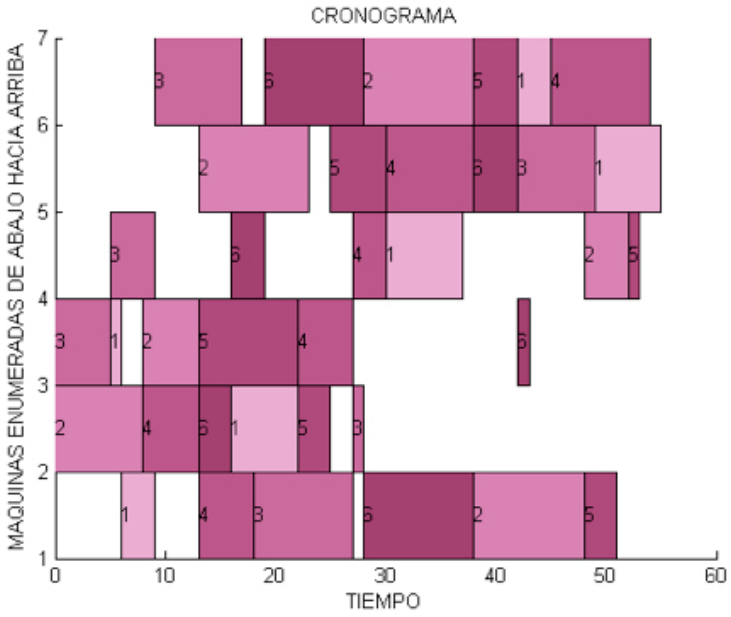

Figura 5: Solución con la metodología propuesta. *Makespan* = 55; tiempo muerto = 133.

4 Discusión

Comparación entre la metodología SP y la técnica FIFO

Se aprecia una mejora del 22.53% respecto a la variable tiempo total de proceso. Este porcentaje es mayor del 22% proyectado inicialmente. Igualmente en los tiempos de subutilización de los centros de trabajo, se evidencia en la técnica tradicional 229 unidades de tiempo muerto, mientras con la metodología propuesta, trabajando subpoblaciones se obtienen 133 unidades de tiempo muerto, lo cual representa una mejora del 41.92%. En

cuanto al costo de energía se obtiene 1236.0 en FIFO, mientras que en SP se tiene 1207.7, esto representa una mejora del 2.29%.

Comparación de resultados entre la metodología SP y la aplicación del NSGA II

Se puede determinar que las soluciones fueron similares, por lo menos en el tiempo de proceso del modelo FT06, es decir el *makespan*. En ambos métodos se obtienen 55 unidades de tiempo, el tiempo muerto en las máquinas es igual, esto es la solución óptima encontrada por la comunidad científica en el ambiente mono-objetivo. Frente al costo del recurso energético difiere un poco. Mientras que el NSGA II, con el tiempo de 55, obtuvo un costo mínimo de 1273.0 la metodología de propuesta generó un costo de 1271.1; es decir, hubo una mejora del 0.1493%.

Como una de las metas propuestas por la empresa que requiere solucionar este tipo de JSSP, de acuerdo a su prioridad, puede ser obtener una solución en un solo objetivo, NSGA II entrega un individuo con un costo de recurso energético de 1213.2, mientras que la metodología propuesta proporcionó un individuo con un costo de 1207.7 con una mejora del 0.4533%. Los datos entre las dos aplicaciones son similares; sin embargo se muestra una tendencia a la mejora en la metodología.

Comparación de eficiencia entre la metodología SP y la aplicación del NSGA II

Frente al número de iteraciones que se necesitan para llegar al óptimo global de 55 unidades de tiempo, luego de 60 simulaciones para cada una, se obtiene una media aritmética asi: 60 iteraciones para la metodología propuesta y 211 para la NSGA II. Aparentemente la metodología es más eficiente que la NSGA II, sin embargo es necesario hacer pruebas estadísticas para demostrarlo. Ingresando los datos en el software estadístico (SPSS) y pasando las pruebas de supuestos estadísticos, ya que los datos obtenidos no presuponen una distribución normal o alguna otra distribución teórica, se aplicó la prueba no paramétrica de Mann-Whitney (ver Tabla 13).

H_0: $m_1 = m_2$ (no hay diferencias entre medias). H_1: $m_1 \neq m_2$ (diferencias estadísticamente significativas). Se rechaza H_0 y se acepta H_1. Hay diferencias altamente significativas.

Tabla 13: Prueba de Mann-Whitney.

Estadísticos de contraste	Iteracíon
U de Mann-Whitney	1215.500
W de Wilcoxon	3045.500
Z	-3.069
Sig. asintót. (bilateral)	0.002

a. Variable de agrupación: Modelo

5 Conclusiones

La metodología SP con selección de subpoblaciones que se propone, constituye un aporte que permite mejorar las soluciones de un problema JSSP. Se demuestran mejoras considerables al compararla con la técnica tradicional FIFO, tanto en tiempos de procesos como en tiempos de subutilización (tiempos muertos).

Igualmente al comparar con la aplicación NSGA II, la metodología SP tiene resultados mejores en el cumplimiento del objetivo relacionado con el costo del recurso. En cuanto al recurso computacional la metodologia SP requiere de menos iteraciones para llegar al óptimo que las requeridas por el NSGA II.

Queda un camino por recorrer haciendo experimentación con casos más complejos, en los cuales se aumente el número de centros de trabajo, número de pedidos, más objetivos, iniciar pedidos en diferentes tiempos, tomar restricciones de tiempos muertos en las máquinas o puestos de trabajo, repetir operaciones para un pedido y/o tomar pedidos con diferente número de operaciones. También en un futuro se puede hacer experimentación con problemas o casos de producción donde las características del sistema evidencien contextos de *open shop* y *flow shop*.

La metodología de subpoblaciones entra en las técnicas eficientes de búsqueda local.

Referencias

[1] Aramendiz Tatis, H.; Sudré, C.; Espitia, M.; Cardona, C.; Correa, E. (2009) "Caracterización morfoagronómica de la berenjena (Solanum melongena L.)", *Revista U.D.C.A. Actualidad & Divulgación Científica* **12**(2): 101–111.

[2] Barbadilla, A. (2012) "Genética de Poblaciones". Universidad Autónoma de Barcelona. En: `http://biologia.uab.es/divulgacio/genpob.html#factores`, consultado: 27/10/2012.

[3] Carvajal Pérez, R.N. (2011) "Un Algoritmo genético especializado en planeamiento de redes de distribución. Parte I: Fundamentos técnicos del algoritmo", *Ingeniería Energética* **XXXII**(1): 72–76.

[4] Castrillón, O.D.; Giraldo, J.A.; Sarache, W.A. (2009) "Solución de un problema job shop con un agente inteligente", *Ingeniería y Ciencia* **5**(10): 75–92.

[5] Chase, R.B.; Aquilano, N.J.; Jacobs, F.R. (2006) *Operations Management for Competitive Advantage (11th ed.)*. Mc Graw-Hill/Irwin, New York.

[6] Cheng, C C.; Smith, S.F. (1997) "Applying constraint satisfaction techniques to job shop scheduling", *Annals of Operations Research* **70**(0): 327–357.

[7] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002) "A fast and elitist multiobjective genetic algorithm", *NSGA-II, IEEE. Transactions on evolutionary computation* **6**(2): 182–197.

[8] Díaz, A.; Glover, F.; Ghaziri, H.; Gonzalez J.; Laguna, M.; Moscato, P. (1996) *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo, Madrid.

[9] Falconer, D.S. (1989) *Introduction to Quantitative Genetics (3rd ed.)*. Lobman, New York.

[10] Fehr, W.R.; Fehr, E.L.; Jessen, J. (1987) *Principles of Cultivar Development: Theory and Technique*. Macmillan, New York.

[11] Fisher, H.; Thompson, G.L. (1963) "Probabilistic learning combinations of local job-shop scheduling rules", in: Muth J.F.; Thompson G.L. (Eds), *Industrial Scheduling*, Prentice Hall, Englewood Cliffs, New Jersey: 225–251.

[12] Font Fernández, J.M. (2012) *Sistema Evolutivo Bioinspirado en el Comportamiento Bacteriano*. Tesis de doctorado, Universidad Politécnica de Madrid, Madrid.

[13] Friese, R.; Brinks, T.; Oliver, C.; Siegel, H.J.; Maciejewski, A.A. (2012) "Analyzing the trade-offs between minimizing makespan and minimizing energy consumption in a heterogeneous resource allocation problem", *Think Mind – INFOCOMP The Second International Conference on Advanced Communications and Computation*: 81–89.

[14] Frutos Alazard, M.; Tohmé Hauptmann, R. (2012) "Técnicas evolutivas en problemas multi-objetivos en el proceso de planificación de la producción", *Ingeniería Industrial* **XXXIII**(1): 50–59.

[15] Ginnobili S. (2010) "La teoría de la selección natural darwiniana y la genética de poblaciones", *Theoria* **67**: 37–58.

[16] Goldberg, D.; Sastry, K. (2008) *Genetic Algorithms: the Design of Innovation*. Springer, Berlin.

[17] Guerra, J.M.; Pereira, J.; Vilà, M. (2012) "A hybrid genetic algorithm for the assembly line balancing problem with fixed number of workstations", in: *6^{th} International Conference on Industrial Engineering and Industrial Management / XVI Congreso de Ingeniería de Organización*. Vigo, España: 895–902.

[18] Hernández M.; Esparza, M. (2011) "Los datos del padrón en el estudio biodemográfico de la población del delta del Ebro a finales del siglo XIX", *Revista de la Sociedad Española de Antropología* **32**: 20–35.

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

[20] Krajewski, L.; Ritzman, L. (2000) *Administración de Operaciones. Estrategia y Análisis (5^a ed)*. Pearson Education, México D.F.

[21] Martella, M.B.; Trumper, E.V.; Bellis, L.M.; Renison, D.; Giordano, P.F.; Bazzano, G.; Gleiser R.M. (2012) "Manual de ecología dinámica espacial en el manejo de las poblaciones", *Reduca (Biología). Serie Ecología* **5**(1): 116–136.

[22] Martínez Pérez, C.A.; Puris Cáceres, A.Y.; Bello Pérez, R. (2012). "Mecanismos de limpieza del espacio de soluciones para fomentar la diversidad de la meta heurística PSO", *Universidad Central "Marta Abreu". Facultad de Matemáticas, Física y Computación. Centro de Estudios de Informática. Departamento de Ciencia de l Computación*: 2–23.

[23] Melián Batista, B.; Moreno Pérez, J.A.; Moreno Vega, J.M. (2009) "algoritmos genéticos. Una visión práctica", *Números, Revista Didáctica de las Matemáticas* **71**: 29–47.

[24] Niño, L.; Prieto, L.; Santiago, V.; Acevedo, E. (2009) "Fluctuación poblacional de la mosca minadora (*Liriomyza huidobrensis Blanchard*) en cultivos de papa de Pueblo Llano, Estado Mérida, Venezuela", *Entomotrópica* **24**(2): 65–70.

[25] Pérez de la Cruz, C.; Ramírez Rodríguez, J. (2011) "Un algoritmo genético para un problema de horarios con restricciones especiales", *Revista de Matemática: Teoría y Aplicaciones* **18**(2): 215–229.

[26] Ruiz Herrera, S.; Castrillón, O.D.; Sarache, W.A. (2012) "Metodología multiobjetivo para programar la producción en un ambiente job shop: open shop", in: *Décima Primera Conferencia Iberoamericana en "Sistemas, Cibernética e Informática: CISCI"*, Orlando, Florida.

[27] Ruiz, S.; Castrillón, O.D.; Sarache, W.A. (2012) "Una metodología multiobjetivo para optimizar un ambiente job shop", *Revista Información Tecnológica* **23**(1): 35–46.

[28] Toro Murillo, A.M. (2010) "Florencio Conde y los valores de la nación de acuerdo con la ideología liberal de José María Samper", *Estudios de Literatura Colombiana* (27).