



Ingeniería. Investigación y Tecnología

ISSN: 1405-7743

iit.revista@gmail.com

Universidad Nacional Autónoma de México
México

Salazar-Hornig, E.J.; Rojas-Oyarzún, R.S.
Configuración multi-objetivo de sistemas de producción utilizando estrategias evolutivas
Ingeniería. Investigación y Tecnología, vol. XI, núm. 4, diciembre, 2010, pp. 423-430
Universidad Nacional Autónoma de México
Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=40419188006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Configuración multi-objetivo de sistemas de producción utilizando estrategias evolutivas

Multi-Objective Configuration of Manufacturing Systems Using Evolutionary Strategies

Salazar-Hornig E.J.

Departamento de Ingeniería Industrial

Universidad de Concepción, Chile

E-mail: esalazar@udec.cl

Rojas-Oyarzún R.S.

Departamento de Ingeniería Industrial

Universidad de Concepción, Chile

E-mail: rociroja@udec.cl

(Recibido: abril de 2008; aceptado: mayo de 2010)

Resumen

Se presenta un método multi-objetivo para determinar la configuración de un sistema de producción tipo jobshop flexible dinámico, de múltiples centros de trabajo con multicapacidad, utilizando la estrategia evolutiva conocida como $(\mu+)$ -ES. La población de soluciones que evoluciona de generación en generación, es evaluada mediante simulación. Se determina el número adecuado de máquinas por centro de trabajo, balanceando la utilización de los recursos y el tiempo de flujo de los trabajos. Si bien, el método se aplica a un sistema prototipo, es generalizable a cualquier problema de configuración y tipo de recursos (discreto o continuo).

Descriptores: configuración de sistemas, optimización multiobjetivo, algoritmos evolutivos, estrategias evolutivas, simulación.

Abstract

A multi-objective optimization method using the evolutionary strategy known as $(\mu+)$ -ES to determine the configuration of a dynamic flexible jobshop manufacturing system with multiple multi-capacity workcenters is presented. The population of solutions, which evolve from generation to generation is evaluated by simulation. The number of identical machines at each workstation is determined, balancing the resource utilization and jobs flowtime. The method is applied to a prototype system, but it can be generalized to any problem of system configuration and resource type (discrete or continuous).

Keywords: System configuration, multiobjective optimization, evolutionary algorithms, evolutionary strategies, simulation.

Introducción

Configurar un sistema productivo implica determinar el nivel de recursos (máquinas, operarios, almacenes, etc.) necesarios para alcanzar una operación eficiente, dada la necesidad de controlar los costos de producción, manteniendo un nivel razonable de recursos (capacidad de producción) y de satisfacción al cliente (cumplimiento de la demanda).

El problema tiene muchas variantes; sin embargo, el objetivo principal se centra en la definición del nivel de recursos que hagan más eficiente la operación de un sistema productivo (Paris *et al.*, 2001; Pierreval *et al.*, 2003). Pierreval y Tautou (1997) combinan simulación y algoritmos evolutivos para determinar la capacidad de un silo de almacenamiento y elegir entre dos métodos de manufactura. Almutawa *et al.* (2005) presentan un enfoque de optimización basada en simulación para optimizar el número de máquinas de diferente capacidad de procesamiento en un proceso de producción por lotes. Feyzioglu *et al.* (2005) presentan un método multi-objetivo para minimizar el nivel de recursos en un sistema de producción, satisfaciendo restricciones operativas. Defersha y Chen (2006) presentan una heurística basada en algoritmos genéticos para formar familias de partes y células de máquinas en el diseño de un sistema celular.

Configuración de un sistema de producción

En este trabajo se desarrolla un método para resolver un problema de configuración de un sistema productivo tipo jobshop flexible dinámico de M centros de trabajo (CT) de multicapacidad. En cada CT_k ($k = 1, \dots, M$) existen m_k máquinas idénticas. El sistema procesa N tipos diferentes de órdenes de producción (OP), que arriban en forma dinámica a través de un proceso aleatorio. Una OP de tipo i ($i = 1, \dots, N$) se compone de n_i operaciones a procesarse en una secuencia predeterminada (el tipo de OP determina la ruta de proceso). Los tiempos de proceso son dependientes de cada operación y se modelan en forma aleatoria.

El desempeño del sistema es evaluado en función de dos criterios, el tiempo de flujo y la utilización de los recursos, empleando la regla de despacho FIFO para secuenciar las OP ; sin embargo, otras reglas de despacho pueden utilizarse dependiendo de los objetivos del problema. Haupt (1989) describe y clasifica reglas de despacho clásicas, Dominic *et al.* (2004) y Holthaus (1997) presentan conceptos recientes de reglas de despacho.

Para el primer objetivo, que se relaciona con la capacidad de respuesta de la empresa para cumplir con la demanda de sus productos, se considera la fracción de proceso media (f), definida como el promedio de la razón entre el tiempo de proceso total y el tiempo de flujo de cada trabajo. Para el segundo objetivo, que se relaciona con un adecuado uso de la inversión, se define la utilización media del sistema (U) representada por el promedio de utilización de los centros de trabajo.

La medida de desempeño fracción de proceso media (f) posee una relación inversa con la utilización media del sistema (U). Mientras mayor es el número de máquinas en el sistema, menor es la utilización media del sistema, aumentando la fracción media de proceso, dado que al existir más recursos disminuyen las esperas producidas por la escasez de estos. Por lo tanto, el objetivo es determinar el número adecuado de máquinas por centro de trabajo que permita alcanzar un equilibrio entre ambas medidas. Este es un problema de optimización multiobjetivo, puesto que presenta dos funciones objetivos a maximizar, f y U :

Maximizar (f, U)

s.a.

$\min_k \quad m_k \quad \max_k \quad \text{para } k=1, \dots, M$

donde (f, U) es el vector objetivo a maximizar, f y U representan las funciones objetivos individuales:

$$f = \frac{\sum_{i=1}^n \frac{p_i}{F_i}}{n} \quad U = \frac{\sum_{k=1}^M U_k}{M}$$

\min_k y \max_k es el número mínimo y máximo de máquinas en el centro de trabajo k ; p_i y F_i el tiempo de proceso y de flujo para el i -ésimo trabajo procesado, y U_k la utilización media del centro de trabajo k . La fracción p_i/F_i es la fracción de proceso para el trabajo i , es decir, la proporción del tiempo que representa su proceso respecto del tiempo total de permanencia en el sistema.

Optimización multiobjetivo

Según Coello (1999), la optimización multiobjetivo se define como la optimización de un vector de funciones (objetivo) dependientes de un vector de variables de decisión en una región factible, las que representan matemáticamente diferentes objetivos individuales, generalmente en conflicto. En este sentido, el término "optimizar" significa encontrar soluciones aceptables a todos

los objetivos, y en general, no existe una única solución que sea mejor con respecto a todos los objetivos.

En un problema de optimización multiobjetivo típico, existe un conjunto de soluciones que son superiores en algún objetivo, pero inferiores a otras soluciones en otros objetivos. Estas soluciones son conocidas como *Soluciones Pareto Óptimas*. Una solución factible A para un problema multiobjetivo es *Pareto Óptima* (solución eficiente o no dominada), si ninguna otra solución factible es tan buena como A con respecto a cada objetivo, y estrictamente mejor con respecto a por lo menos un objetivo. El conjunto de soluciones eficientes es denominada *Frontera de Pareto*.

Algoritmos evolutivos en optimización multi-objetivo

Bajo el término *algoritmos evolutivos* se entiende a todos los algoritmos basados en procedimientos de búsqueda inspirados en mecanismos de la evolución natural, como por ejemplo, *algoritmos genéticos* y *estrategias evolutivas*, utilizando un vocabulario de genética natural. Asocia el concepto de individuo a solución factible del problema, y el de población a un conjunto de soluciones factibles.

Los individuos están formados por unidades (genes o caracteres) ordenadas en sucesión lineal, siendo evaluados a través de su aptitud (fitness).

Se identifican dos generaciones de algoritmos evolutivos multiobjetivo (Coello, 2000), la primera clasifica los métodos en Pareto y no Pareto, la segunda generación introduce los conceptos de elitismo y de población secundaria.

Entre los algoritmos evolutivos no Pareto de primera generación se puede mencionar a *VEGA* (vector evaluated genetic algorithm) de Schaffer (1985); y otros métodos como *Agregación de Funciones*, *Programación por Metas* y el *Enfoque Min – Max* (Coello, 1999).

Dentro de los algoritmos evolutivos Pareto de primera generación se puede mencionar *MOGA* (*multi-objective genetic algorithm*) de Fonseca y Fleming (1993), *NSGA* (*Nondominated Sorting Genetic Algorithm*) de Srinivas y Deb (1993) y *NPGA* (*Niched Pareto Genetic Algorithm*) de Horn et al. (1994).

Entre los algoritmos evolutivos de segunda generación se pueden mencionar *SPEA* (*Streng Pareto Evolutionary Algorithm*) de Zitzler y Thiele (1999), *PAES* (*Pareto Archived Evolutionary Strategy*) de Knowles y Corne (1999), *PESA* (*Pareto Envelope based Selection Algorithm*) de Corne et al. (2000), *SPEA II* de Zitzler et al. (2001), *PESA II* de Corne et al. (2001), *MOMGA* (*Multi-Objective with*

Messy Genetic Algorithm) de Van Veldhuizen y Lamont (2000) y *MOMGA II* de Zydallis et al. (2001).

Algoritmo para la configuración de un sistema de producción

En este trabajo se utiliza el enfoque de optimización del vector objetivo, específicamente el método Min-Max. Este método utiliza la distancia entre una solución factible (eficiente) y una solución ideal (vector de variables de decisión correspondiente a los óptimos individuales). Este método busca encontrar un punto en la Frontera de Pareto donde la desviación máxima al punto ideal sea mínima.

Estrategia evolutiva

Las estrategias evolutivas son algoritmos evolutivos basados en procesos aleatorios de selección y de variación que fueron diseñados inicialmente para resolver problemas de optimización de parámetros. El proceso de selección favorece con mayor probabilidad a aquellos individuos con mayor fitness (padres) para ser sometidos a los operadores genéticos de variación: mutación y/o recombinación (Michalewicz, 1999; Back et al., 1997).

El proceso de recombinación permite una mezcla de información de los padres para traspasarla a los descendientes (nuevas soluciones). La mutación modifica ciertas características de los individuos y previene la pérdida completa de ciertas características en la población. Back et al. (1997) describen este proceso evolutivo, que se inicia con una población inicial de μ individuos, la que evoluciona de generación en generación por medio de operadores de variación, cuya finalidad es preservar la diversidad genética en la población. El proceso evolutivo se repite hasta cumplir algún criterio de finalización (generalmente hasta alcanzar un número de generaciones).

En este trabajo se utiliza la estrategia evolutiva $(+)-ES$, donde individuos producen descendientes que son evaluados determinando su fitness. La nueva población de $(+)$ individuos es reducida por un proceso de selección a $-$ individuos.

La representación genética de un individuo es un vector de M genes, (m_1, m_2, \dots, m_M) , donde el valor m_k del gen k ($k = 1, \dots, M$) representa el número de máquinas en el CT_k , limitado en el intervalo $[min_k, max_k]$.

La población inicial de soluciones se determina en forma aleatoria, según Pierrehal y Tautou (1997) y Michalewicz (1999). La solución ideal corresponde al caso cuando la fracción media de proceso es 1, los trabajos no

presentan tiempo de espera en su proceso y la utilización media es 1, lo que significa que las máquinas están siendo utilizadas el 100% del tiempo. Así, la función de evaluación para el individuo k de la población se define como la distancia euclidiana entre (f_k, U_k) y la solución ideal $(1,1)$:

$$d_k = \sqrt{(1 - f_k)^2 + (1 - U_k)^2} \quad k = 1, \dots, N_p$$

donde N_p corresponde al tamaño de la población.

Una vez obtenida la población de descendientes, el proceso de selección elige los individuos de mejor fitness, forma referida frecuentemente como *Ranking y Selección* (Spear *et al.*, 1993).

La mutación es un operador genético que modifica aleatoriamente uno o más genes de un individuo. Su propósito es prevenir que la población converja tempranamente a un óptimo local y se permita explorar nuevas vecindades de soluciones.

El operador de recombinación combina dos cromosomas padres para formar uno o dos descendientes (hijos) por intercambio de genes de los padres. Este operador permite realizar una búsqueda en profundidad, explotando las buenas características de la población actual.

En este trabajo se consideró un operador de *mutación en vecindad* basado en la idea original de las estrategias evolutivas en que todos los genes varían en una vecindad de su valor actual (Michalewicz, 1999), y una adaptación discreta de la *recombinación intermedia* de Mühlenbein *et al.* (1993).

Mutación en vecindad: el valor de cada gen del individuo mutado se obtiene de una vecindad del valor actual; si $m_k = \min_k$ (\max_k) entonces m_k se incrementa (decrementa) en 1. Si $\min_k < m_k < \max_k$ se determina en forma aleatoria si m_k se incrementa o decrementa en 1.

Recombinación intermedia: los valores de los genes de los descendientes se generan como una ponderación aleatoria de los valores de los genes de los padres; se calcula $v_k = a_k \cdot m_{1k} + (1 - a_k) \cdot m_{2k}$, con m_{ik} el valor del gen k del padre i ($i=1,2$), $a_k \sim U[-d, 1+d]$ (en este trabajo se utilizó $d=0.25$, por lo que $a_k \sim U[-0.25, 1.25]$). El valor v_k se redondea al entero más cercano; si $\min_k \leq v_k \leq \max_k$ entonces $m_k = v_k$, en otro caso, si $v_k < \min_k$ ($v_k > \max_k$) entonces $m_k = \min_k$ ($m_k = \max_k$).

Por ejemplo, en un sistema de 3 CT, con límites inferior y superior de máquinas por CT indicado en la tabla 1, el cromosoma (5, 6, 9) muta al cromosoma (4, 7, 10), si en forma aleatoria se determina que el primer gen se decrementa en 1 y el tercer gen se incrementa en 1 (el

gen 2 se incrementa en 1 dado que su valor actual era igual al mínimo).

Tabla 1. Ejemplo de sistema de 3 CT

CT	1	2	3
\min_k	3	6	5
\max_k	8	15	12

Como ilustración de recombinación, considere que para los padres (5, 10, 9) y (6, 8, 11) se genera en forma aleatoria $(a_1, a_2, a_3) = (0.8, -0.1, 1.2)$, obteniéndose $(v_1, v_2, v_3) = (5.2, 7.8, 8.6)$ (5, 8, 9). Dado que los valores redondeados se encuentran en el intervalo permitido, el descendiente es (5, 8, 9).

Aplicación al sistema prototipo

El sistema prototipo utilizado (Salazar y Larrazábal, 2006) consiste en diez centros de trabajo (CT) de multi-capacidad con máquinas idénticas. Se procesan 5 tipos de trabajos, que arriban a través de un proceso aleatorio con tiempos entre arribos distribuido exponencial con media 0.1 horas. En la tabla 2 se resumen las características por tipo de trabajo. Ruta corresponde a la secuencia de CT que procesan las n_i operaciones del trabajo tipo i . El tiempo de proceso de las operaciones se considera aleatorio distribuido triangular simétrico $p_{ij} T(a_{ij}, b_{ij}, c_{ij})$, donde p_{ij} representa el tiempo de proceso de la j -ésima operación del i -ésimo tipo de trabajo (tabla A1 en Anexo). Se utiliza la regla de despacho FIFO para la programación de los trabajos.

Tabla 2. Número mínimo y máximo de máquinas

Tipo	Probabilidad	n_i	Ruta
1	0.20	4	3-1-2-5
2	0.15	5	4-8-6-1-2
3	0.15	8	5-3-6-4-2-8-1-7
4	0.40	10	9-8-1-6-5-7-4-2-3-10
5	0.10	7	7-2-1-9-10-6-5

El rango para el número de máquinas en cada CT se obtuvo utilizando el factor de utilización $= ta/(s \cdot tp)$, siendo t_a y t_p la tasa media de arribo y de proceso, respectivamente, y s corresponde al número de máquinas en el CT (Law y Kelton, 2000). Para fijar los límites de

máquinas, se define una utilización de referencia u_r , obteniendo el valor de s que garantiza la utilización de referencia. Para el límite inferior se impone la condición de estado estacionario ($\rho < 1$), y para el límite superior se define $\rho = 0.3$. La tabla 3 especifica el rango para el número de máquinas por CT.

Tabla 3. Número mínimo y máximo de máquinas

CT	1	2	3	4	5	6	7	8	9	10
\min_k	7	7	4	5	5	6	4	4	3	2
\max_k	26	25	25	18	19	23	16	16	11	9

Para evaluar cada solución (configuración), se simuló el sistema prototipo durante 200 horas (40 horas de warm-up y 160 horas para la recolección de información). El procesamiento se realizó mediante el software SPS_Optimizer/Configuration¹.

Se utilizó $\mu = 30$ y $\sigma = 60$ para el tamaño de la población de padres y de descendientes, respectivamente.

Para el criterio de término se consideran 250 generaciones.

Para evaluar el desempeño del método se consideraron dos factores: *mejores soluciones* (evolución de la mejor solución a través de las generaciones y *calidad media de la población* (fitness promedio de la población en cada generación). Para obtener una evaluación más representativa se realizaron cinco réplicas.

La figura 1 muestra la evolución de la mejor solución en cada una de las 5 réplicas; la mejor solución obtuvo un fitness igual a 0.21218.

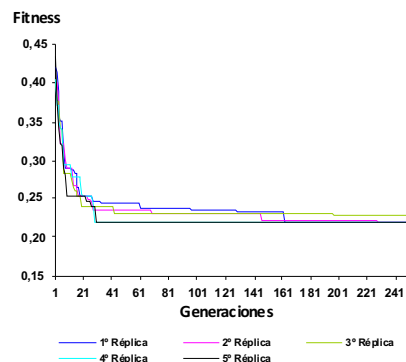


Figura 1: Evolución de mejores soluciones

La figura 2 muestra gráficamente la evolución de la calidad de la población, la que mejora a medida que aumenta el número de generaciones. La calidad de la población se obtiene promediando el fitness promedio de las 5 réplicas en cada generación.

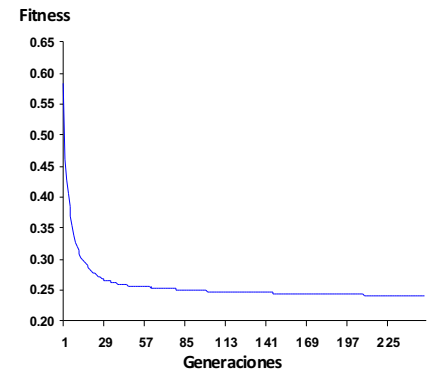


Figura 2: Calidad de la población

Se obtuvieron las cincuenta mejores soluciones generadas en el proceso evolutivo de cada una de las 5 réplicas, disponiendo de un conjunto de 250 buenas soluciones del problema. Se agrupan aquellas soluciones idénticas (igual número de máquinas en los CT), promediando sus utilidades medias por CT. De esta forma, es posible ajustar el fitness y se eligen las cincuenta mejores soluciones, para luego definir la Frontera de Pareto (figura 3 y tabla A2 en anexo).

Las soluciones que forman la Frontera de Pareto tienen la característica de que por lo menos en uno de los objetivos son mejores que las otras soluciones. Para acotar el número de soluciones se decide utilizar como criterio de selección que la fracción de proceso del sistema y la utilización media de las máquinas sean de al menos 80%.

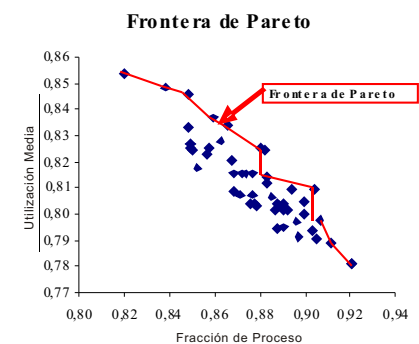


Figura 3. Frontera de Pareto

¹ Desarrollado por Profesor Eduardo Salazar H. (esalazar@udec.cl)

De acuerdo a este criterio, son nueve las soluciones que lo satisfacen. Cada una de estas soluciones presenta su propia configuración de máquinas en cada centro de trabajo.

Para determinar una única solución al problema, se presentan tres enfoques de análisis de las soluciones de la Frontera de Pareto: del *mejor fitness*, del *tomador de decisión* y de *análisis detallado*:

Mejor Fitness: se considera que la mejor solución es la que posee el mejor fitness, es decir, la solución que esté más cercana a la solución ideal. La mejor solución, con fitness 0.21218, presenta fracción de proceso promedio 88,16% y utilización media 82,39%. La configuración del sistema resulta: 9, 9, 5, 7, 8, 8, 5, 6, 4 y 3 máquinas, respectivamente.

Tomador de decisión: el tomador de decisión es quien establece el balance de los objetivos. Por ejemplo, si la utilización media del sistema es el objetivo crítico, se considera la solución con mayor utilización media: 85,37%, con fitness 0.23150 y fracción de proceso promedio 82,06%. La configuración del sistema resulta: 9, 8, 5, 6, 6, 8, 5, 5, 4 y 4 máquinas, respectivamente.

Análisis detallado: las soluciones de la Frontera de Pareto se analizan en forma detallada mediante estudios de simulación de más réplicas obteniendo una estimación más precisa de la fracción de proceso promedio y utilización media del sistema.

Conclusiones

Se han identificado aplicaciones de los enfoques evolutivos en el área de diseño de sistemas de manufactura, siendo posible incorporar múltiples objetivos y restricciones comúnmente encontradas en la práctica.

En este trabajo se ha utilizado la *estrategia evolutiva* conocida como (+) – ES para configurar un sistema productivo prototipo tipo jobshop flexible dinámico. Si bien las Estrategias Evolutivas han sido aplicadas principalmente a problemas continuos de optimización de parámetros, en este caso, esta estrategia se aplicó a un problema discreto, determinando el número de máquinas en cada centro de trabajo.

Este problema considera dos objetivos, maximizar la utilización media de las máquinas, relacionado con restricciones de recursos y maximizar la fracción de proceso de las órdenes de producción, relacionado con la capacidad del sistema para responder a la demanda, considerando el enfoque Min–Max para la optimización multiobjetivo.

Las estrategias evolutivas requieren definir y calibrar parámetros, antes de resolver cualquier problema, en particular, el problema de configuración analizado en este trabajo.

De acuerdo a esta investigación se observó que los parámetros que se utilicen inciden en la calidad media de la población como en la obtención de la mejor solución.

Anexo

Tabla A1. Parámetros distribución triangular tiempos de proceso

Tipo Orden	Parámetro	$p_{ij} \sim T(a_{ij}, b_{ij}, c_{ij})$
1	a	0.500, 0.600, 0.850, 0.500
	b	0.750, 0.850, 1.025, 0.650
	c	1.000, 1.100, 1.200, 0.800
2	a	0.800, 0.800, 0.750, 0.700, 0.600
	b	1.000, 1.050, 0.875, 0.800, 0.800
	c	1.200, 1.300, 1.000, 0.900, 1.000
3	a	0.400, 0.250, 0.700, 0.300, 0.350, 0.400, 0.500, 0.450
	b	0.800, 0.575, 0.850, 0.800, 0.725, 0.625, 0.85, 0.725
	c	1.200, 0.900, 1.000, 1.300, 1.100, 0.850, 1.200, 1.000
4	a	0.600, 0.400, 0.650, 0.800, 0.500, 0.650, 0.600, 0.550, 0.400, 0.450
	b	0.650, 0.500, 0.725, 0.850, 0.625, 0.725, 0.650, 0.650, 0.450, 0.525
	c	0.700, 0.600, 0.800, 0.900, 0.750, 0.800, 0.700, 0.750, 0.500, 0.600
5	a	0.300, 0.250, 0.300, 0.350, 0.400, 0.500, 0.200
	b	0.600, 0.525, 0.650, 0.625, 0.550, 0.750, 0.525
	c	0.900, 0.800, 1.000, 0.900, 0.700, 1.000, 0.850

Tabla A2: Frontera de Pareto

Fitness	f	U
0,21218	0,88164	0,82390
0,21221	0,88018	0,82485
0,21309	0,90389	0,80982
0,21365	0,86598	0,83362
0,21583	0,85941	0,83625
0,21643	0,84845	0,84549
0,21969	0,88287	0,81413
0,22198	0,83802	0,84822
0,22286	0,90663	0,79764
0,22919	0,91134	0,78866
0,23150	0,82056	0,85374
0,23299	0,92050	0,78099

Finalmente, se concluye que utilizando estrategias evolutivas se pueden resolver problemas discretos, como el de configuración de sistemas productivos, obteniéndose buenos resultados en tiempos computacionales aceptables (para el problema analizado en este trabajo, el tiempo CPU requerido fue del orden de 10 s en un computador centrino de 1,5 GHz).

Referencias

- Almutawa S, Savsar M., Al-Rashdan K. Optimum Machine Selection in Multistage Manufacturing Systems. *International Journal of Production Research*, 43(6):1109-1126, 2005.
- Back T, Hammel U., Schwefel H.P. Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*, 1(1):3-17, 1997.
- Coello C. An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. En: Proceedings of the 1999 Congress on Evolutionary Computation (1999, Washington, DC, United State). 1999, pp. 1-13.
- Coello C. An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys*, 32(2):109-143, 2000.
- Corne D., Jerram N., Knowles J., Oates M. PESA-II: Region Based Selection in Evolutionary Multiobjective Optimization. En: Proceedings of the 2001 Genetic and Evolutionary Computation Conference 2001, San Francisco, California, USA). 2001, pp. 283-290.
- Corne D, Knowles J., Oates M. The Pareto Envelope Based Selection Algorithm for Multiobjective Optimization. En: Proceedings of the 2000 International Conference on Parallel Problem Solving from Nature (6th, 2000, Berlin, Germany). 2000, pp. 839-848.
- Defersha F.M., Chen M. Machine Cell Formation Using a Mathematical Model and a Genetic-Algorithm-Based Heuristic. *International Journal of Production Research*, 44(12): 2421-2444, 2006.
- Dominic P.D.D, Kaliyamoorthy M., Saravana-Kumar. Efficient Dispatching Rules for Dynamic Job Shop Scheduling. *The International Journal of Advanced Manufacturing Technology*, 24:70-75, 2004.
- Feyzioglu O, Pierreval H., Deflandre D. A Simulation-Based Optimization Approach to Size Manufacturing Systems. *International Journal of production Research*, 43(2):247-266, 2005.
- Fonseca C.M., Fleming P.J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. En: Proceedings of the 1993 International Conference on Genetic Algorithms (5th, 1993, California, United State). 1993, pp. 416-423.
- Haupt R.A. Survey of Priority Rule-Based Scheduling. *OR Spectrum*, 11(1):3-16, 1989.
- Holthaus O. Design of Efficient Job Shop Scheduling Rules. *Computers and Industrial Engineering*, 33(1): 249-252, 1997.
- Horn J., Nafpliotins N., Golberg D. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. En: Proceedings of the First IEEE Conference on Evolutionary Computation (1994, Orlando, FL, United State). 1994, pp. 82-87.
- Knowles J., Corne D. The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Pareto. En: Proc. of the 1999 Congress on Evolutionary Computation (1999, Washington, DC, United State). 1999, pp. 1-8.
- Law A., Kelton W. *Simulation Modeling and Analysis*. 3rd. United States, Ed. McGraw Hill. 2000. 760 p.
- Michalewicz Z. *Genetic Algorithms+Data Structures=Evolution Programs*. 2nd. US, Ed. Springer. 1999. 387 p.

- Mühlenbein H., Schlierkamp-Voosen D. Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization. *Evolutionary Computation*, 1(1): 25-49, 1993.
- Paris J.L., Tautou L., Pierreval H. Dealing with Design Options in the Optimization of Manufacturing Systems: an Evolutionary Approach. *International Journal of Production Research*, 39(6):1081-1094, 2001.
- Pierreval H., Caux C., Paris J.P., Viguier F. Evolutionary Approach to the Design and Organization of Manufacturing System. *Computers & Industrial Engineering*, 44(3):339-364, 2003.
- Pierreval H., Tautou L. Using Evolutionary Algorithms and Simulation for the Optimization of Manufacturing Systems. *IIE Transactions*, 29(3):181-189, 1997.
- Salazar E. y Larrazábal M.C. Configuración de sistemas de producción utilizando algoritmos genéticos. Actas (CD) XII CLAIO-Congreso Latino Iberoamericano de Investigación Operativa. Montevideo, Uruguay, 27. 11-30. 11. 2006.
- Schaffer J.D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. En: Proceedings of the First International Conference on Genetic Algorithms (1985, NJ, United State). 1985, pp. 93-100.
- Spear W., De Jong K., Back T., Fogel D.B., Garis H. An Overview of Evolutionary Computation. En: Proceedings of the European Conference on Machine Learning (1993, London, UK). 1993, pp. 442-459.
- Srinivas N., Deb K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *The Journal Evolutionary Computation*, 2(3):221-248, 1993.
- Van-Veldhuizen D., Lamont G. Multiobjective Optimization with Messy Genetic Algorithms. En: Proceedings of the 2000 ACM Symposium on Applied computing (2000, Como, Italia). 2000, pp. 470-476.
- Zydallis J.B., Van Veldhuizen D.A., Lamont G.B. A Statical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II. En: Proceedings of the 2001 International Conference on Evolutionary Multi Criterion Optimization (2001, London, UK). 2001, pp. 226-240.
- Zitzler E., Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions On Evolutionary Computation*, 3(4):257-271, 1999.
- Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich Switzerland. 2001, pp. 21.

Semblanza de los autores

Eduardo Javier Salazar-Hornig. Es ingeniero matemático por la Universidad de Concepción (1984), obtuvo el grado de magíster en investigación de operaciones en la RWTH University of Aachen, Alemania en 1992. Su línea de investigación incluye gestión de operaciones, sistemas de producción, planificación y programación de producción y simulación. Es profesor de tiempo completo en el Departamento de Ingeniería Industrial y Programa de Magíster en Ingeniería Industrial de la Universidad de Concepción.

Rocío Soledad Rojas-Oyarzún. Es ingeniero civil industrial por la Universidad de Concepción (2006) y candidata a magíster del Programa de Magíster en Ingeniería Industrial de la misma universidad. Sus áreas de interés se concentran en la gestión de operaciones y evaluación de proyectos. Actualmente se desempeña como ingeniero asistente de proyectos de mejoramiento de procesos en la empresa Servicios Financieros Progreso en Santiago de Chile.