

# INTRODUCCIÓN DE ELEMENTOS DE MEMORIA EN EL MÉTODO SIMULATED ANNEALING PARA RESOLVER PROBLEMAS DE PROGRAMACIÓN MULTIOBJETIVO DE MÁQUINAS PARALELAS

## INTRODUCTION OF MEMORY ELEMENTS IN SIMULATED ANNEALING METHOD TO SOLVE MULTIOBJECTIVE PARALLEL MACHINE SCHEDULING PROBLEMS

Felipe Baesler<sup>1</sup> Reinaldo Moraga<sup>2</sup> Oscar Cornejo<sup>3</sup>

Recibido 28 de junio de 2005, aceptado 29 de abril de 2008  
Received: June 28, 2005 Accepted: April 29, 2008

### RESUMEN

El presente artículo introduce una variante de la metaheurística *Simulated Annealing*, para la resolución de problemas de optimización multiobjetivo. Este enfoque se denomina *MultiObjective Simulated Annealing with Random Trajectory Search*, MOSARTS. Esta técnica agrega al algoritmo *Simulated Annealing* elementos de memoria de corto y largo plazo para realizar una búsqueda que permita balancear el esfuerzo entre todos los objetivos involucrados en el problema. Los resultados obtenidos se compararon con otras tres metodologías en un problema real de programación de máquinas paralelas, compuesto por 24 trabajos y 2 máquinas idénticas. Este problema corresponde a un caso de estudio real de la industria regional del aserrío. En los experimentos realizados, MOSARTS se comportó de mejor manera que el resto de la herramientas de comparación, encontrando mejores soluciones en términos de dominancia y dispersión.

Palabras clave: Simulated Annealing, multiobjetivo, programación de la producción.

### ABSTRACT

*This paper introduces a variant of the metaheuristic Simulated Annealing, oriented to solve multiobjective optimization problems. This technique is called MultiObjective Simulated Annealing with Random Trajectory Search (MOSARTS). This technique incorporates short and long term memory concepts to Simulated Annealing in order to balance the search effort among all the objectives involved in the problem. The algorithm was tested against three different techniques on a real life parallel machine scheduling problem, composed of 24 jobs and two identical machines. This problem represents a real life case study of the local sawmill industry. The results showed that MOSARTS behaved much better than the other methods utilized, because found better solutions in terms of dominance and frontier dispersion.*

Keywords: Simulated Annealing, multiobjective, scheduling.

### INTRODUCCIÓN

Los problemas del ámbito aplicado por lo general son multiobjetivos por naturaleza. Rara vez una decisión es tomada en base a un solo criterio. Por esta razón, resulta de gran interés el desarrollo de herramientas que reflejen las necesidades industriales y que puedan entregar soluciones en un tiempo razonable. No existen algoritmos exactos que puedan resolver problemas reales con más de un objetivo en un tiempo razonable. Es por esta razón que las metaheurísticas surgen como una alternativa práctica para resolver este tipo de problemas. Simulated Annealing (SA) corresponde a una metaheurística que ha sido probada profundamente

en problemas de un solo objetivo con resultados que avalan su buen desempeño en términos de eficiencia y eficacia. En el ámbito multiobjetivo no existen muchos desarrollos asociados a esta herramienta, la mayoría de los desarrollos corresponden a algoritmos genéticos (AG) multiobjetivo que presentan la ventaja de un esquema poblacional lo cual facilita el modelamiento en términos multiobjetivo, sin embargo, debido a este mismo aspecto presentan una convergencia lenta. Es por este motivo que surge como oportunidad investigar en desarrollar un algoritmo multiobjetivo basado en Simulated Annealing que sea capaz de resolver problemas de tamaño industrial en un tiempo y calidad aceptable y con mayor velocidad de convergencia que AG.

<sup>1</sup> Departamento de Ingeniería Industrial. Universidad del Bío-Bío. Avenida Collao 1202. Concepción, Chile. E-mail: fbaesler@ubiobio.cl

<sup>2</sup> Department of Industrial and System Engineering. University of Northern Illinois. DeKalb, USA. E-mail: moraga@ceet.niu.edu

<sup>3</sup> Facultad de Ingeniería. Universidad Católica de la Santísima Concepción. Concepción, Chile. E-mail: ocornejo@ucsc.cl

Los problemas de programación de la producción corresponden a problemas combinatorios de alta complejidad considerados en su mayoría intratables a través de algoritmos exactos. Además, estos problemas son de alta aplicabilidad en el ámbito industrial, por lo cual resultan de gran interés tanto desde una perspectiva investigativa como de transferencia tecnológica. Es por esta razón que este trabajo muestra una nueva metaheurística basada en Simulated Annealing que incorpora elementos de memoria de largo y corto plazo dentro de su proceso de búsqueda. Esta hibridación permite incorporar a Simulated Annealing la capacidad de búsqueda en un contexto multiobjetivo.

## MARCO TEÓRICO

La optimización multiobjetivo puede ser definida como un problema de optimización que presenta dos o más funciones objetivo. El inconveniente principal que presenta este tipo de problemas, en relación a un modelo de objetivo único, radica en la subjetividad de la solución encontrada. Un problema multiobjetivo no tiene una solución óptima única, más bien, genera un set de soluciones que no pueden ser consideradas diferentes entre sí. De esta manera el conjunto de soluciones óptimas se denomina Frontera de Pareto. Esta frontera de soluciones contiene todos los puntos que no son superados en todos los objetivos por otra solución. Este concepto lleva el nombre de dominancia, por esta razón la frontera de Pareto consiste solo de soluciones no dominadas. Una solución domina a otra si y solo si es al menos tan buena como las otras soluciones en todos sus objetivos y es la mejor de todas en al menos un objetivo.

Para un vector de funciones objetivo (minimización)

$$\vec{f}(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x}))$$

Un candidato  $\vec{x}_1$  domina a  $\vec{x}_2$  ( $\vec{x}_1 \leq \vec{x}_2$ ). Si:

$$f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \quad \forall i \in \{1, \dots, k\}$$

y

$$\exists i \in \{1, \dots, k\} : f_i(\vec{x}_1) < f_i(\vec{x}_2)$$

La decisión final con respecto a cual solución seleccionar de esta frontera de puntos depende de la perspectiva de cada tomador de decisiones (TD). Es decir, depende del nivel de compromiso que un tomador de decisiones otorga a cada objetivo dentro de su particular análisis. Bajo esta perspectiva las técnicas para resolver problemas

multiobjetivo se dividen en tres grandes grupos, dependiendo de cómo se incorpora el punto de vista del TD en el problema. Estos grupos de técnicas se denominan como:

- Técnicas de articulación previa de preferencias del TD
- Técnicas búsqueda interactivas
- Técnicas de articulación posterior de preferencias del TD

En el primer grupo el TD entrega sus preferencias en relación a la importancia relativa de un objetivo con respecto al resto. Esta información es incorporada al modelo con el fin de encontrar la solución que satisface los requerimientos de ese TD en particular. Las técnicas interactivas requieren que el TD se involucre en el proceso de búsqueda en forma constante. Finalmente, las técnicas de articulación posterior de preferencias buscan generar la Frontera de Pareto en su totalidad con el fin que cada TD decida, de esta gama de soluciones, cuál punto satisface sus requerimientos. Esta forma de ver el problema multiobjetivo es considerado un enfoque moderno para enfrentar este tipo de problemas. La mayoría de las metodologías que se han desarrollado en los últimos años se enmarcan en esta categoría. La resolución de los problemas multiobjetivo bajo el enfoque de articulación posterior de preferencias del TD requiere un gran esfuerzo computacional cuando se enfrenta con las llamadas técnicas clásicas, como son el método de la restricción y el de las ponderaciones [1]. El uso de estas estrategias requiere la resolución de un problema diferente para la obtención de cada punto de la frontera de Pareto. Muchas veces el tiempo computacional ya es prohibitivo para encontrar a optimalidad un solo punto. Por esta razón el uso de estas técnicas clásicas no es viable en problemas de tamaño considerable [2]. Como alternativa aparecen diferentes metaheurísticas que han sido probadas con éxito en problemas de un objetivo, las cuales han sido adaptadas para la búsqueda de soluciones en un contexto multiobjetivo [3]. Sin duda las técnicas más utilizadas están ligadas al concepto de algoritmos evolutivos donde se destacan los algoritmos genéticos. Los primeros intentos en utilizar algoritmos genéticos en optimización multiobjetivo bajo el enfoque de generación de la frontera de Pareto son atribuidos a [4], quien presenta el algoritmo denominado VEGA (Vector Evaluated Genetic Algorithm). Este enfoque es bastante simple y hace uso de la condición poblacional que presentan los algoritmos genéticos. De esta manera el autor subdivide los individuos de cada generación en subpoblaciones asociadas a cada objetivo considerado en el estudio. Para cada subpoblación se realiza una selección en forma independiente. Posteriormente los cromosomas

seleccionados en cada grupo se mezclan en un grupo común con el fin de aplicar cruzamiento y así diversificar el proceso de búsqueda. VEGA representa el precursor de los algoritmos desarrollados posteriormente, sin embargo presenta significativas falencias, principalmente en la convergencia de las soluciones a zonas concentradas en los extremos de la frontera de Pareto. Con el fin de solucionar este inconveniente surgen diferentes propuestas que nacen a partir del trabajo realizado por [5], el cual incorpora el concepto de dominancia dentro del proceso de selección de individuos en el AG. El autor propone separar la población en grupos denominados ranking. Cada grupo contendrá soluciones no dominadas, por lo cual entre los miembros de un mismo ranking no debe existir diferencia en su función de fitness. En forma adicional el autor introduce el concepto de formación de nichos y la utilización de una función de *fitness compartido* con lo cual se combatiría el fenómeno de formación de grupos de soluciones concentrados en sectores de la frontera de Pareto (nichos). Estos elementos permiten diversificar la generación de soluciones sobre una mayor superficie de la frontera. Trabajos posteriores de diferentes autores, en general, han mantenido ambos conceptos, búsqueda no dominada a través del uso del concepto de ranking y la utilización de fitness compartido. Estos trabajos presentan mejoras en diversos sentidos, sin embargo la esencia de estos conceptos es mantenida. Algunas de estas técnicas de AG multiobjetivo son, Multi-Objective Genetic Algorithm (MOGA) [6], Nondominated Sorting Genetic Algorithm (NSGA), [7], Strength Pareto Evolutionary Algorithm (SPEA), [8], Multi-Objective Messy Genetic Algorithm (MOMGA), [9]. La mayoría de las propuestas posteriores corresponden a modificaciones de algunos de los algoritmos presentados, como son NSGA II, [10], MOMGA II, [11], SPEA II, [12].

Esta diversidad de alternativas de estrategias evolutivas, para enfrentar el problema de optimización multiobjetivo, ha generado algunos estudios en los cuales se compara el desempeño de algunas técnicas en la resolución de problemas de prueba. Uno de los estudios más importantes es presentado por [13]. Los autores comparan seis técnicas de optimización, cinco de las cuales corresponden a algoritmos genéticos presentados en este resumen, y una de ellas a un enfoque totalmente aleatorio. Estas seis estrategias son aplicadas para resolver seis problemas de prueba existentes en la literatura. Cada problema fue resuelto cinco veces por cada una de las estrategias de resolución y comparados los resultados en base a diferentes medidas de efectividad y eficiencia. Las conclusiones posicionan en primer lugar a SPEA, [8], y en último lugar a la estrategia totalmente aleatoria. Otros estudios similares son presentados por [14, 15 y 10]

Otra metaheurística que ha sido adaptada para la búsqueda en un contexto multiobjetivo corresponde Simulated Annealing (SA). A pesar que SA ha sido utilizada intensamente en diferentes problemas que presentan objetivo único, la utilización de SA en problemas con más de un objetivo está limitada a un puñado de casos. Los primeros intentos de utilizar esta técnica en problemas multiobjetivo pueden ser atribuidos a [16]. El enfoque presentado en este trabajo apunta a aceptar soluciones inferiores (bajo el prisma multiobjetivo una solución inferior corresponde a una solución dominada), con una probabilidad basada en la norma de Tchebycheff:

$$P(Y, X, T) = \text{Min} \left\{ 1, e^{-\max_j \{ \lambda_j (f_j(X) - f_j(Y)) / T \}} \right\}$$

lo cual representa la probabilidad de aceptar una solución  $Y$  dada una solución  $X$  a una temperatura  $T$ . El parámetro  $\lambda_j$  representa el peso de la función objetivo  $j$  con respecto a la importancia total. En otras palabras, el objetivo que presenta la mayor diferencia, el peor, es elegido para ser evaluado. Si es aceptado, el algoritmo avanza a la nueva solución dominada  $Y$ . Un enfoque similar es presentado por [17] y [18] a través de la heurística denominada Multi-Objective Simulated Annealing (MOSA). Este enfoque utiliza una suma ponderada de las diferencias de todos los objetivos involucrados en el análisis. Este valor es utilizado para calcular la probabilidad de aceptación. La ecuación siguiente muestra este enfoque.

$$P(Y, X, T) = \text{Min} \left\{ 1, e^{-\sum_{j=1}^k \{ \lambda_j (f_j(X) - f_j(Y)) / T \}} \right\}$$

La mayoría de los enfoques de SA multiobjetivo se diferencian en la manera de definir la probabilidad de aceptar un movimiento hacia una solución dominada. En [19] se realiza un estudio donde se compara Simulated Annealing multiobjetivo con algoritmos genéticos. Lo más importante de este estudio es la utilización de seis tipos de funciones de aceptación diferentes para la implementación de SA. Los resultados obtenidos al someter los algoritmos a diferentes pruebas demostraron que existen diferencias significativas entre las distintas funciones de aceptación. Algunos enfoques donde se proponen funciones de aceptación basadas en agregación de objetivos pueden ser encontrados en [20-21].

Una propuesta algo diferente es presentada bajo el nombre de Pareto Simulated Annealing (PSA) por [22]. A pesar de utilizar una función de aceptación basada en una suma

ponderada similar a MOSA, los autores introducen un concepto en el cual los pesos asociados a cada objetivo cambian dinámicamente en función a la calidad de la solución encontrada, es decir, el peso asociado a un objetivo que empeora en relación a la solución existente, es multiplicado por un factor mayor que 1, por lo cual su valor aumenta. De igual forma los objetivos que mejoraron en relación a la solución existente se dividen por este factor con lo cual se logra una disminución en su peso relativo. Este enfoque permite diversificar la búsqueda y alejarse lo más posible de la solución actual. PSA utiliza el concepto de población, similar a los algoritmos genéticos, de esta manera la búsqueda se realiza en paralelo. Un estudio comparativo de diferentes enfoques de SA multiobjetivo es presentado en [23]. También es posible encontrar artículos que muestran casos prácticos, especialmente problemas combinatorios, donde se ha utilizado SA con un enfoque multiobjetivo. En [24] se propone una variante de SA para resolver problemas de reconocimiento de imágenes. En [25] se utiliza una propuesta de SA multiobjetivo para definir los tiempos óptimos de exposición en exámenes radiológicos. En [26] se utiliza PSA con objetivos difusos y aplica esta técnica a un problema de programación de proyectos en el área de la agricultura. Otras aplicaciones se pueden encontrar en [27] recarga de agua presurizada en reactores y [28] diseño de estructuras para bicicletas.

## METODOLOGÍA

Los intentos por incorporar en SA la capacidad de búsqueda en más de un objetivo se concentran en proponer diferentes alternativas de agregación de las variaciones de cada objetivo dentro de la función de energía que define la probabilidad de aceptar una solución inferior. En esta línea se destacan los algoritmos propuestos por [16, 17 y 22]. Este proyecto propone eliminar este paradigma y explorar en un punto de vista alternativo que se orienta a seleccionar, en cada iteración, un solo objetivo el cual será el encargado de guiar la búsqueda durante esa iteración del algoritmo. De esta manera, la función de aceptación tradicional de SA puede ser utilizada sobre el objetivo seleccionado, y eliminar así la necesidad de modificar la estructura natural de SA. Este enfoque concentraría la investigación en incorporar elementos inteligentes, memorias de corto y largo plazo, que permitirán seleccionar la dirección en la cual el algoritmo deberá concentrarse en la iteración en cuestión. Con esta filosofía se utilizaría SA en su forma tradicional, un solo objetivo, con la salvedad que se incorporarán cambios de la dirección durante el proceso de búsqueda. La probabilidad de aceptación de una solución inferior está definida por la variación de la función objetivo con respecto a la solución anterior y por

la temperatura actual del sistema. Usando esta misma analogía se puede proponer que cada objetivo presente una temperatura independiente, la cual, en parte, sería responsable de la selección de un objetivo referente para esa iteración en particular. De esta manera se propone incorporar el concepto de aleación de metales donde cada metal tiene su propia temperatura de enfriamiento, la cual debe decrecer en forma controlada en relación al resto de los metales que forman parte de la aleación con el fin de alcanzar el equilibrio térmico del sistema.

### Memoria de largo plazo

La utilización de memoria de largo plazo permite controlar la evolución histórica de cada objetivo en forma individual. En otras palabras, consiste en un registro el cual contabiliza el porcentaje de mejoramiento que ha experimentado cada objetivo durante el proceso de búsqueda.

### Memoria de corto plazo

La memoria de corto plazo permite contabilizar el porcentaje de mejoramiento que ha experimentado un objetivo en particular en la última iteración del algoritmo.

En cada etapa de la búsqueda, tras la evaluación de todos los objetivos de la solución actual con respecto a la solución anterior, se pueden enfrentar tres escenarios: a) todos los objetivos mejoran (solución actual domina a la anterior); b) todos empeoran (solución actual es dominada por la anterior); o c) algunos mejoran y otros empeoran (solución actual y anterior son no dominadas, es decir, ninguna domina a la otra). En el primer caso, no hay mayor problema en decidir continuar la búsqueda en esa dirección, ya que la solución actual es mejor que la anterior, pues la domina. En caso que todos los objetivos empeoren, la solución actual (dominada) debe ser evaluada por algún tipo de función de aceptación. En el tercer caso (soluciones no dominadas) se debe tomar una decisión. Elegir uno de los objetivos como referente, y efectuar el movimiento acorde al desempeño de ese objetivo particular u objetivo referente (OR) en la etapa de la búsqueda en que se encuentre. La solución actual se debe aceptar si ese objetivo mejoró, o evaluar una función de aceptación particular para ese objetivo, en caso que haya empeorado. En este caso, un solo objetivo, el objetivo referente, guía la búsqueda de todo el problema, basado en una probabilidad de selección dada por una función denominada función de selección (FS).

La selección del objetivo referente en cada etapa está dada, como se mencionó anteriormente, por una función de selección, la cual no emplea tan sólo el criterio obvio de



selección aleatoria, sino que incorpora también la mejora o detrimento de cada objetivo, es decir, el desempeño, tanto en la última etapa de la evaluación (desempeño inmediato o memoria de corto plazo), como en todo el trayecto recorrido (desempeño histórico o memoria de largo plazo). Estos dos criterios se pueden aunar en una sola función de probabilidad, o función de selección, de acuerdo a ponderadores arbitrarios.

La nueva metaheurística basada en Simulated Annealing tiene por objetivo guiar la búsqueda hacia una solución que vaya en la dirección de satisfacer todos los objetivos en forma simultánea, o bien, construir una frontera eficiente de Pareto lo más dispersa posible.

La nueva metaheurística se basa en la toma de decisión acerca de cuál de los objetivos múltiples, en cada evaluación de las funciones objetivo a medida que avanza la búsqueda, es la que servirá de elemento de decisión para el siguiente paso. Esto se hace, como se mencionó anteriormente, mediante una función de selección de objetivo como objetivo referente. En el caso que la solución actual resulte no dominada (algunos objetivos empeoran y otros mejoran), primero se deben seleccionar al objetivo referente para que sirva de elemento de decisión. Esto se hace mediante la función de selección. Si el objetivo referente seleccionado es tal que la evaluación de su función objetivo resultó en una mejora, se debe continuar la exploración en esa dirección. Por otro lado, si el objetivo seleccionado es tal que la evaluación de su función objetivo resultó en una desmejora, se debe consultar la función de aceptación de Simulated Annealing, pero esta vez considerando la curva de enfriamiento particular para ese objetivo (CEP).

De este modo, la función de selección  $FS$  se puede expresar en general como:

$$FS = w_1 * F_1 + w_2 * F_2$$

$$F_1 = T_i / T_{i\max}$$

Donde:

$F_1$ : Función de probabilidad de selección de objetivo según desempeño histórico (memoria de largo plazo).

El concepto tras la construcción de la función  $F_1$  se basa en dar una mayor probabilidad de selección a aquellos objetivos que, en lo que va corrido de la búsqueda, no hayan tenido un buen desempeño, o bien, no hayan mejorado lo suficiente. Un parámetro que da cuenta de cuánto ha

avanzado un objetivo en particular es precisamente, su temperatura de enfriamiento particular ( $T_i$ ). Aquellos objetivos con una temperatura  $T_i$  más alta no se han “enfriado” lo suficiente, o bien, no han avanzado o sido explorados en muchas oportunidades, por lo que, para no “dejarlos atrás”, la función  $F_1$  les asigna una mayor probabilidad de selección como objetivo referente (OR), pues la razón  $T_i/T_{i\max}$  es directamente proporcional a la temperatura de enfriamiento particular  $T_i$ .

Por otro lado, para la función  $F_2$  se tiene:

$$F_2 = 1 - \left( \Delta C_{\%i} / \sum \Delta C_{\%i} \right)$$

donde:

$F_2$ : Función de probabilidad de selección de objetivo según desempeño reciente (memoria de corto plazo).

$\Delta C_{\%i}$ : Variación porcentual del objetivo  $i$ .

El concepto tras la construcción de la función  $F_2$  se basa en dar una mayor probabilidad de selección a aquellos objetivos que, en la última evaluación, hayan tenido un buen desempeño. Un parámetro que da cuenta de cuánto ha cambiado un objetivo en particular es la variación porcentual del objetivo ( $\Delta C_{\%i}$ ). Aquellos objetivos con una mayor variación porcentual  $\Delta C_{\%i}$  negativa (minimización) han mejorado más en la última evaluación, por lo cual la función  $F_2$  les asigna una mayor probabilidad de selección como objetivo referente (OR), pues la razón  $\Delta C_{\%i} / \sum \Delta C_{\%i}$  es directamente proporcional a la variación porcentual  $\Delta C_{\%i}$ .

Así, para cada objetivo  $i$ , su función de selección resulta:

$$FS_i = w_{1i} * F_1 + w_{2i} * F_2$$

Luego, para seleccionar el objetivo referente se debe construir una ruleta, la cual estaría compuesta de la probabilidad de selección ( $PS_i$ ) para cada objetivo en forma relativa a través de la siguiente transformación.

$$PS_i = FS_i / \sum FS_i$$

Para la ecuación anterior se cumple que:

$$\sum PS_i = 1$$

$w_{1i}$  y  $w_{2i}$  representan la importancia entre memoria de largo y corto plazo, respectivamente.

El algoritmo se muestra en la siguiente secuencia de pasos:

- 1) Definir solución inicial  $X_0$ .
- 2) Definir vecino de  $X_0$  como  $X_1$ .
- 3) Evaluar objetivos  $F_i(X_0)$  y  $F_i(X_1)$
- 4) Si  $X_1$  domina a  $X_0$ , hacer  $X_0 = X_1$  y volver a 2).
- 5) Si  $X_1$  es dominada por  $X_0$ , evaluar función de selección FS. Obtener objetivo referente OR. Ir a 7).
- 6) Si  $X_1$  no es dominada por  $X_0$ , evaluar función de selección FS. Obtener objetivo referente OR. Ir a 7).
- 7) Si OR mejora, hacer  $X_0 = X_1$ , actualizar temperatura particular  $T_i$  volver a 2).
- 8) Si OR empeora, obtener temperatura  $T_i$  de curva de enfriamiento particular  $TEP_i$ .

- 9) Evaluar función de aceptación particular FAP  $exp((\Delta C/T_i)$  para objetivo referente OR.
- 10) Si FAP rechaza la solución, volver a 2).
- 11) Si FAP es aceptada, hacer  $X_0 = X_1$ , actualizar temperatura particular  $T_i$  y volver a 2).

### CASO DE ESTUDIO

Con el fin de evaluar el comportamiento del algoritmo planteado se realizó una prueba con el problema de programación de la producción de máquinas paralelas idénticas. Este problema puede ser caracterizado como  $P_2$   $r_j, d_j, S_{ij}, C_{max}, \sum T_j$  y tiene la siguiente estructura matemática presentada en [29]. Ver figura 1

$$\begin{aligned} \text{Minimizar} \quad & C_{\max} \quad , \quad \sum_{j=1}^n T_j & (1) \\ \text{sujeto a} & \\ \sum_{\substack{i=0 \\ i \neq j}}^n \sum_{k=1}^m x_{i,j,k} = 1 & \quad \forall j = 1, \dots, n & (2) \\ \sum_{\substack{i=0 \\ i \neq h}}^n x_{i,h,k} - \sum_{\substack{j=0 \\ j \neq h}}^n x_{h,j,k} = 0 & \quad \forall h = 1, \dots, n, \forall k = 1, \dots, m & (3) \\ C_j \geq C_i + \sum_{k=1}^m x_{i,j,k} (S_{i,j,k} + p_{j,k}) + M \left( \sum_{k=1}^m x_{i,j,k} - 1 \right) & \quad \forall i = 0, \dots, n, \forall j = 1, \dots, n, & (4) \\ \sum_{j=0}^n x_{0,j,k} = 1 & \quad \forall k = 1, \dots, m & (5) \\ x_{i,j,k} \in \{0,1\} & \quad \forall i = 0, \dots, n, \forall j = 0, \dots, n, \forall k = 1, \dots, m & (6) \\ C_0 = 0 & & (7) \\ C_j \geq 0 & \quad \forall j = 1, \dots, n & (8) \end{aligned}$$

Figura 1. Estructura del problema.

Donde,

- $C_j$ : Tiempo de completación del trabajo  $j$ .
- $p_{j,k}$ : Tiempo de proc. del trabajo  $j$  en máquina  $k$ .
- $S_{i,j,k}$ : Tiempo de setup dependiente al procesar trabajo  $j$  después del trabajo  $i$  en la máquina  $k$ .
- $S_{0,j,k}$ : Tiempo de setup para procesar el trabajo  $j$  primero en la máquina  $k$ .
- $x_{i,j,k}$ : 1 si el trabajo  $j$  es procesado directamente después de  $i$  en la máquina  $k$  y 0 en otro caso.
- $x_{0,j,k}$ : 1 si el trabajo  $j$  es el primer trabajo a ser procesado en la máquina  $k$  y 0 en otro caso.
- $x_{j,0,k}$ : 1 si el trabajo  $j$  es el último en ser procesado en la máquina  $k$  y 0 en otro caso.
- $M$ : Un número positivo muy grande.

Los objetivos (1) son minimizar el tiempo de completación máximo y la tardanza total. Las restricciones (2) aseguran que cada trabajo es programado solo una vez y procesado por solo una máquina. Las restricciones (3) aseguran que cada trabajo no sea ni precedido o seguido por más de un trabajo. Las restricciones (4) son utilizadas para calcular el tiempo de completación y para asegurar que ningún trabajo sea precedido o seguido por el mismo trabajo. Las restricciones (5) aseguran que solo un trabajo puede ser programado primero en cada máquina. Las restricciones (6) especifican que la variable de decisión  $x$  es binaria. Las restricciones (7) estipulan que el tiempo de completación del trabajo ficticio 0 es cero y las restricciones (8) aseguran que los tiempos de completación son no negativos.

Los objetivos considerados para este estudio son la minimización del tiempo máximo de completación ( $C_{max}$ ) y la minimización de la tardanza total ( $T$ ). Como problema de prueba se utilizó un caso real correspondiente a la industria fabricación de molduras, específicamente en la etapa de moldurado que consta de dos máquinas en paralelo. El tamaño del lote a producir es variable, sin embargo para este estudio se utilizó un problema de 24 trabajos con tiempo de preparación dependiente del trabajo predecesor en la máquina. Para comparar el rendimiento del algoritmo se utilizaron 3 elementos de comparación; la solución generada por la empresa, el método de la restricción (programación entera) y los resultados generados con el algoritmo Simulated Annealing multiobjetivo MOSA desarrollado por [17-18]. Los resultados de los experimentos se muestran en la siguiente sección.

## RESULTADOS

Con la finalidad de comparar las técnicas MOSA y MOSARTS de la manera más homogénea posible, se consideró fijar los mismos parámetros a ambas técnicas. De esta manera, se definió una temperatura inicial y final de 100 y 1 grado respectivamente, logrando así una comparación equitativa. Esto permite aislar el efecto asociado a los parámetros de las metaheurísticas y sólo evaluar el comportamiento asociado a su capacidad de búsqueda. Para el caso de MOSARTS fue necesario definir un parámetro adicional, correspondiente a la ponderación asociada a las memorias de corto y largo plazo. Dado que corresponde a sólo un parámetro, no se realizó un diseño de experimentos, como es tradicional en este tipo de estudios. En su defecto se fijaron 5 posiciones para los parámetros  $w_1$  y  $w_2$  y cada una fue replicada 10 veces con diferentes semillas para definir puntos de partida alternativos. Los resultados de este análisis se muestran en la tabla 1.

Tabla 1. Fijación Parámetros MOSARTS.

	$w_1=0.2$ $w_2=0.8$	$w_1=0.4$ $w_2=0.6$	$w_1=0.5$ $w_2=0.5$	$w_1=0.6$ $w_2=0.4$	$w_1=0.8$ $w_2=0.2$
Número Prom. de sols. no dominadas	4.8	5.6	6.1	5.9	8.2
Número de veces que encontró la mejor solución	2	4	3	1	0

La tabla muestra que los mejores resultados en términos de mejor frontera se encontraron en la configuración que otorga una ponderación de 0.4 a la memoria de largo plazo y un 0.6 a la de corto plazo, sin embargo esta configuración no genera el mayor número de soluciones no dominadas. Se aprecia que a medida que aumenta la ponderación de la memoria de largo plazo, el número de soluciones en la frontera aumenta, pero la calidad no es la mejor. El algoritmo de comparación MOSA también se replicó 10 veces. En forma adicional se modeló el problema como programación entera y se enfrentó el problema multiobjetivo a través de la técnica denominada el método de la restricción. Este método resuelve los extremos de la frontera de Pareto para luego resolver un problema de un solo objetivo pero incorporando el otro objetivo como restricción. Este proceso requiere resolver un nuevo problema para cada punto de la frontera de Pareto que se quiere generar. Este problema se implementó en el paquete de optimización CPLEX en

su versión 10.0, sin realizar ningún tipo de modificación del algoritmo de ramificación y acotamiento que tiene incorporado por defecto. Dado el tamaño del problema no fue posible lograr convergencia, por lo tanto se decidió correr cada modelo por un total de 8 horas y seleccionar la mejor solución encontrada en ese lapso de tiempo. Los resultados obtenidos se resumen en la tabla 2.

Tabla 2. Resultados de experimentos.

	Número Promedio de soluciones no dominadas	Número de veces que encontró la mejor solución
MOSA	6.8	2
e-Constraint	----	1
MOSARTS ( $w_1=0.4$ $w_2=0.6$ )	5.6	7
Manual	0	0

La figura 2 muestra el gráfico de las cuatro técnicas utilizadas con sus mejores resultados. Se aprecia que MOSA presenta un mayor número de soluciones dentro de la frontera que encontró, sin embargo existe una gran concentración de puntos en la zona media, lo cual indica una falta de distribución sobre la frontera. El gráfico muestra que MOSARTS encontró la mejor frontera de Pareto, es decir, una frontera que domina al resto de las fronteras encontradas por las otras técnicas. De igual manera es posible apreciar que existe una mejor distribución de puntos a lo largo de la frontera.

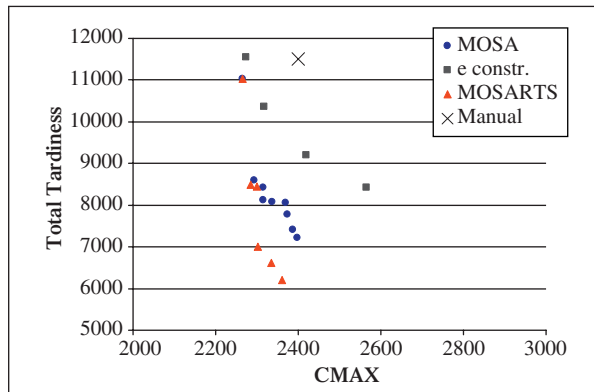


Figura 2. Gráfico de las fronteras de Pareto encontradas por las 4 técnicas.

### CONCLUSIONES

Este artículo presenta un enfoque metaheurístico basado en Simulated Annealing para enfrentar problemas combinatorios multiobjetivo. La gran diferencia que destaca a MOSARTS en relación a otras configuraciones de Simulated Annealing multiobjetivo radica en la utilización

de elementos de inteligencia asociados a memorias de largo y corto plazo. Estas memorias al balancearse en forma justa demostraron ser un elemento poderoso para enfrentar este tipo de problemas. MOSARTS encontró la mejor frontera de Pareto en comparación con las otras 3 técnicas utilizadas y además presentó una dispersión de puntos más distantes entre sí que MOSA. Este punto es de suma importancia, ya que en general en el campo de la optimización multiobjetivo, la construcción de una curva que contenga la mayor dispersión de soluciones posible representa un objetivo de interés. Finalmente, se pretende continuar esta investigación con la realización de una mayor cantidad de experimentos con diferentes números de trabajos y máquinas para poder generalizar los resultados encontrados hasta ahora. De igual manera se pretende comparar los resultados generados por MOSARTS con otras técnicas populares como son los algoritmos Genéticos Multiobjetivo.

También es de suma importancia destacar que todas las técnicas utilizadas son mejores que la solución manual generada por la empresa. En el caso de MOSARTS solo en el objetivo  $C_{max}$  fue posible mejorar en alrededor de un 10%, lo que se traduce en un aumento de la productividad en esa cantidad. Ahora si consideramos en objetivo tardanza, la mejora alcanza cerca del 50%, lo cual puede ser interpretado como satisfacción usuaria.

### AGRADECIMIENTOS

Este trabajo fue financiado por los Proyectos 052011 3/R de la Dirección de Investigación de la Universidad del Bío-Bío, Concepción-Chile y DIN 10/2005 de la Dirección de Investigación Universidad Católica de la Santísima Concepción, Concepción-Chile.

### REFERENCIAS

- [1] M. Ehrgott and X. Gandibleux (editores). "Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys". Kluwer Academic Publishers. Boston. 2002.
- [2] K. Deb. "Multi-Objective Optimization using Evolutionary Algorithms". John Wiley & Sons. ISBN 0-471-87339-X. Chichester, United Kingdom. 2001.
- [3] C. Coello, D. Van Veldhuizen and G.B. Lamont "Evolutionary Algorithms for Solving Multi-Objective Problems". Kluwer Academic Publishers. New York. Marzo 2002.



- [4] J. Schaffer. "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms". Thesis to obtain the degree of doctor. Vanderbilt University. 1984.
- [5] D. Goldberg. "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley Publishing Company. Reading, Massachusetts. 1989.
- [6] C.M. Fonseca and P.J. Fleming. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423. Morgan Kaufman Publishers. San Francisco, California. 1993.
- [7] N. Srinivas and K. Deb. "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms". Evolutionary Computation. Vol. 2 N° 3, pp. 221-248. 1994.
- [8] E. Zitzler and L. Thiele. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach". IEEE Transactions on Evolutionary Computation. Vol. 3 N° 4, pp. 257-271. 1999.
- [9] D. Van Veldhuizen and G. Lamont. "On Measuring Multiobjective Evolutionary Algorithm Performance". 2000 Congress on Evolutionary Computation. Vol. 1, pp. 204-211. Piscataway, New Jersey. IEEE Service Center. 2000.
- [10] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan. "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II". In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo and Hans-Paul Schwefel (editores). Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858. Springer. 2000.
- [11] J. Zydallis, D. Van Veldhuizen and G. Lamont. "A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II". In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 226-240. Springer-Verlag. Lecture Notes in Computer Science N° 1993. 2001.
- [12] E. Zitzler, M. Laumanns and L. Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm". In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (eds.) EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95-100, Athens, Greece. 2002.
- [13] E. Zitzler, K. Deb and L. Thiele. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results". Evolutionary Computation. Vol. 8 N° 2, pp. 173-195. 2000.
- [14] D.A. Van Veldhuizen. "Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations". PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
- [15] J.D. Knowles and D.W. Corne. "A Comparison of Diverse Approaches to Memetic Multiobjective Combinatorial Optimization". In Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, pp. 103-108. Las Vegas, Nevada. 2000.
- [16] P. Serafini. "Simulated Annealing for Multiple Objective Optimization Problems". In Proceedings of the Tenth International Conference of Multiple Criteria Decision Making, Taipei. Vol. 1, pp. 87-96. Julio 1992.
- [17] E. Ulungu, J. Teghem and P. Fortemps. "Heuristics for multi-objective combinatorial optimization by simulated annealing". In J. Gu, G. Chen, Q. Wei, and S. Wang, editors, Multiple Criteria Decision Making: Theory and Applications. Proceedings of the 6th National Conference on Multiple Criteria Decision Making, pp. 228-238. Windsor, United Kingdom. 1995.
- [18] E. Ulungu, J. Teghem and P. Fortemps. "MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems". Journal of Multi-Criteria Decision Analysis. Vol. 8 N° 4, pp. 221-236. 1999.
- [19] D. Nam and C. Park. "Multiobjective Simulated Annealing: A Comparative Study to Evolutionary Algorithms". International Journal of Fuzzy Systems. Vol. 2 N° 2, pp. 87-97. 2000.

- [20] M.P. Hansen. "Generating a Diversity of Good Solutions to a Practical Combinatorial Problem using Vectorized Simulated Annealing". Technical report. Institute of Mathematical Modelling, Technical University of Denmark. Denmark. August 1997.
- [21] P. Lucic and D. Teodorovic. "Simulated Annealing for the multi-objective aircrew rostering problem". *Transportation Research Part A*. Vol. 33, pp.19-45. 1999.
- [22] P. Czyzak and A. Jaskiewicz. "Pareto Simulated Annealing-a metaheuristic technique for multiple-objective combinatorial optimization". *Journal of Multi-Criteria Decision Analysis*. Vol. 7, pp. 34-47. 1998.
- [23] B. Suman and P. Kumar. "A survey of Simulated Annealing as a tool for single and multiobjective optimization". *Journal of the Operational Research Society*. Vol. 57 N° 10, pp. 1143-1160. 2006.
- [24] A. Nakib, H. Oulhadj and P. Siarry. "Non-supervised image segmentation based on multiobjective optimization ". *Pattern Recognition Letters*. Vol. 29 N° 2, pp. 161-172. 2007.
- [25] E. Aggelogiannaki and H. Sarimveis. "Simulated Annealing algorithm for prioritized multiobjective optimization-implementation in an adaptive model predictive control configuration". *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*. Vol. 37 N° 4, pp. 902-915. 2007.
- [26] M. Hapke A. Jaskiewicz and R. Slowinski. "Pareto Simulated Annealing for Fuzzy Multi-Objective Combinatorial Optimization". *Journal of Heuristics*. Vol. 6 N° 3, pp. 329-345. 2000.
- [27] G.T. Parks and A. Suppaitnarm. "Multiobjective optimization of PWR reload core designs using Simulated Annealing". *Mathematics & Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*. Vol. 2, pp. 1435-1444. Madrid, Spain. 1999.
- [28] A. Suppaitnarm, G.T. Parks, K. Shea & P.J. Clarkson. "Conceptual Design of Bicycle Frames by Multiobjective Shape Annealing". *Engineering Optimization*. Vol. 36 N° 2, pp. 165-188. 2004.
- [29] G. Rabadi, R.J. Moraga and A. Al-Salem. "Heuristics for the unrelated parallel machine scheduling problem with setup times". *Journal of Intelligent Manufacturing*. Vol. 17 N° 1, pp. 85-97. 2006.