

MÉTODO HÍBRIDO ENTRE EL ALGORITMO GENÉTICO DE CHU-BEASLEY Y SIMULATED ANNEALING PARA LA SOLUCIÓN DEL PROBLEMA DE ASIGNACIÓN GENERALIZADA

RESUMEN

Este artículo presenta una metodología híbrida entre el algoritmo genético desarrollado por Chu-Beasley adecuado al Problema de Asignación Generalizada y el algoritmo Simulated Annealing. Se propone generar la población inicial utilizando una metodología heurística constructiva basada en factores de sensibilidad y se modifica el modelo matemático tradicional del problema de asignación generalizada incorporando factores de penalización.

PALABRAS CLAVES: Algoritmos Genéticos, Problema de Asignación Generalizada, Optimización Combinatorial.

ABSTRACT

This paper presents a hybrid methodology between genetic algorithm, developed by Chu-Beasley to Generalized Assignment Problem, and Simulate annealing algorithm. Sensitivity factors are used to generate the initials conditions and penalty factors are used in mathematical model.

KEYWORDS: Genetic algorithm, Generalized Assignment, Combinatorial, Optimization

1. INTRODUCCIÓN

El problema de asignación generalizada (PAG) es un problema clásico de investigación de operaciones considerado como NP-completo. El problema consiste en encontrar una adecuada asignación de m tareas a n agentes de manera que se minimice el costo total de asignación.

En la literatura especializada se encuentran soluciones a este problema utilizando metaheurísticas como Búsqueda Tabú [6], las cuales son utilizadas para encontrar soluciones de buena calidad en problemas de gran tamaño. En el campo de la optimización clásica se destacan los algoritmos basados en Branch and Bound. Estas estrategias de solución son eficientes en problemas de pequeño y mediano tamaño [3].

En este artículo se utiliza como estrategia de solución al PAG una metodología híbrida entre el algoritmo genético (AG) de Chu-Beasley [4] y el algoritmo Simulated Annealing (SA) [1]. La estrategia consiste en abordar la etapa de mutación del AG a través del SA.

Adicionalmente, se proponen dos estrategias orientadas a mejorar la factibilidad y disminuir el esfuerzo computacional. La primera consiste en generar una población inicial que tenga en cuenta un análisis de sensibilidades usando un algoritmo heurístico constructivo. De manera que, si se compara con una población obtenida de forma aleatoria, el punto de búsqueda inicial será de mejor calidad y el esfuerzo computacional, para alcanzar una buena solución, será menor. La segunda estrategia consiste en incorporar en el

Fecha de Recepción: 31 Mayo de 2005
Fecha de Aceptación: 05 Septiembre de 2005

ELIANA M. TORO OCAMPO

Ingeniera Industrial.

Profesor

Estudiante Maestría IE, área de Investigación operativa.

Universidad Tecnológica de Pereira

eliana@ohm.utp.edu.co

MAURICIO GRANADA E.

Ingeniero Electricista, M.Sc

Profesor Auxiliar Programa

Ingeniería Eléctrica.

Universidad Tecnológica de Pereira

magra@utp.edu.co

modelo matemático factores que penalicen la infactibilidad.

Se emplean sistemas de prueba de la literatura especializada con el fin de comprobar la metodología propuesta, obteniéndose resultados de alta calidad los cuales son comparados con otras metodologías.

2. ASIGNACIÓN GENERALIZADA

El modelo matemático modificado que representa el problema de asignación generalizada usando factores de penalización es:

$$\begin{aligned} & \min \\ & \sum_{i=1}^n \sum_{j=1}^m C_{ij} X_{ij} + \lambda \sum_{i \in \left\{ \begin{array}{l} \text{restricciones} \\ \text{violadas} \end{array} \right\}} \left(\alpha_i \underbrace{\left(\sum_{j=1}^n A_{ij} X_{ij} - b_i \right)}_{\text{cantidad de recursos excedidos}} \right) \\ & \text{s.a.} \\ & \sum_{i=1}^n X_{ij} = 1 \\ & X \in \{0,1\} \end{aligned} \quad (1)$$

Donde; C_{ij} es el costo de designar la tarea j al agente i , m es el número de tareas y n el número de agentes, X_{ij} corresponde a los elementos de la matriz de alternativas, donde cada uno de estos elementos toman el valor de 0 o 1. Por ejemplo, si la tarea k es asignada al agente l

entonces el elemento $X_{kl}=1$, mientras que los demás agentes disponibles para realizar esa misma tarea tomarán un valor igual a cero ($X_{ki} = 0$ donde $i \neq l$). Lo anterior asegura que una tarea sea asignada sólo una vez a un único agente. A_{ij} representa los recursos consumidos por el agente i al realizar la tarea j , b_i representan la capacidad total de recursos del agente i , λ es el parámetro encargado de controlar el peso de la infactibilidad, de manera que $\lambda < 1$ hace que la penalización sea un porcentaje del costo total de los recursos excedidos. $\lambda = 1$ implica que se penalice la violación de recursos con el valor exacto del costo total de los recursos excedidos. $\lambda > 1$ incrementa la penalización y hace que el método elimine más rápidamente la alternativa infactible evaluada. α_i es el factor que relaciona costo por unidad de recurso del agente i y garantiza la correspondencia en unidades de costo en la función objetivo permitiendo obtener el costo total de los recursos excedidos. Por lo tanto, este parámetro corresponde a un multiplicador de Lagrange que se calcula de la siguiente manera:

$$\alpha_i = \frac{\sum_{j=1}^m C_{ij} X_{ij}}{b_i} \quad (2)$$

3. POBLACIÓN INICIAL

Se propone un algoritmo heurístico constructivo que permite la obtención de una población inicial de alternativas de asignación que tiene en cuenta el estudio de sensibilidades. Para esto, se define una matriz de sensibilidades S , la cual relaciona, elemento a elemento, la matriz de costos C con la matriz de recursos A de manera que mide la aproximación de la relación costo/recurso actual del agente i para realizar la tarea j con respecto a una relación costo/recurso ideal para realizar la misma tarea j , es decir:

$$S_{ij} = \frac{C_{ij}}{A_{ij}} - CA_{IDEAL\ j} \quad (3)$$

Donde $CA_{IDEAL\ j}$ es la relación entre el menor costo de asignación asociado a la tarea j y el menor consumo de recurso asociado a la misma tarea j , esto es:

$$CA_{IDEAL\ j} = \frac{\min_{i=1}^n (C_{ij})}{\min_{i=1}^n (A_{ij})} \quad (4)$$

De esta manera, los elementos de la matriz S más aproximados a cero serán los que cumplen con una mejor relación costo/recurso y los agentes asociados a dicho elemento tendrán una probabilidad mayor al momento de decidir la asignación de la tarea j . La asignación puede realizarse determinísticamente, es decir, asignando la tarea al agente con mayor probabilidad. O puede realizarse probabilísticamente.

4. SIMULATED ANNEALING

El algoritmo de SA puede interpretarse como iteraciones sucesivas del algoritmo de Metrópolis [1] en las que se evalúa y decreta de forma controlada la temperatura o parámetro de control. El algoritmo de Metrópolis simula el proceso físico basado en temperatura en el que los sólidos obtienen estados de baja energía o equilibrio térmico, este proceso es conocido como *recocido de sólidos*. Durante este proceso, la energía libre del sólido es minimizada.

En optimización combinatorial se desarrolla un proceso similar analizando el proceso de recocido de sólidos. Este proceso puede ser formulado como un problema que encuentra, entre muchas soluciones, aquella que tenga mínimo costo. Así se establecen las siguientes correspondencia: (i) Una alternativa de solución en un problema de optimización combinatorial es equivalente al estado de un sistema físico y (ii) el valor de la función objetivo es equivalente a la energía del estado. De esta forma es introducido un método de solución en el campo de la optimización combinatorial basado en la simulación del proceso físico de recocido de sólidos que se denomina "Simulated Annealing".

La idea original que dió lugar a esta metaherística se denomina algoritmo de Metrópolis, el que a su vez está basado en el método de Monte Carlo, con el cual se estudian las propiedades de equilibrio en el análisis del comportamiento microscópico de los cuerpos.

El algoritmo de metrópolis se basa en la técnica de Monte Carlo para generar una secuencia de estados del sólido en su proceso de enfriamiento. Este algoritmo consiste en que si se tiene un estado actual del sólido i con energía E_i , entonces el subsiguiente estado j con energía E_j es generado aplicando un mecanismo de perturbación consistente en provocar una pequeña distorsión en el estado actual. Si la diferencia de energía entre estados, $E_j - E_i$, es menor o igual a cero, el estado j es aceptado como el estado actual. Si la diferencia de energía es mayor a cero, el estado j es aceptado con cierta probabilidad, la cual esta determinada por:

$$e^{\left(\frac{E_i - E_j}{k_B T}\right)} \quad (5)$$

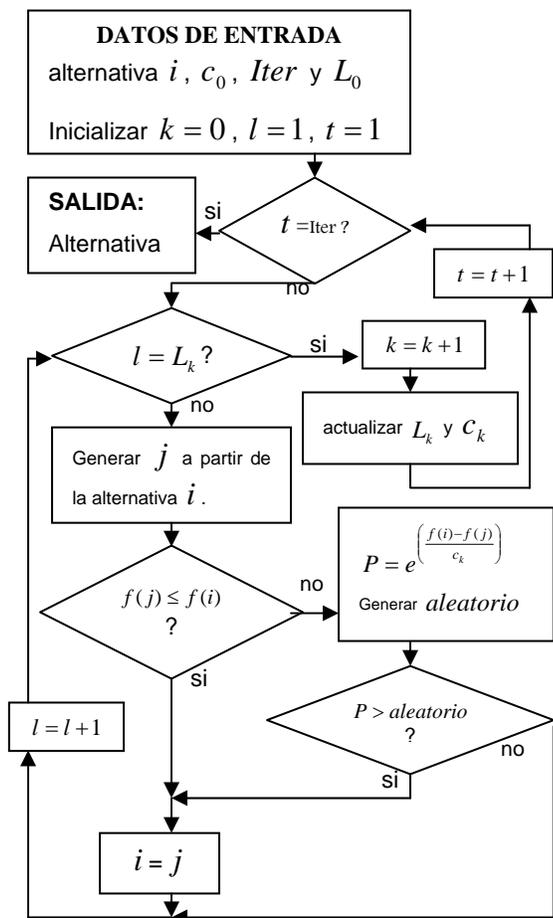


Figura 1. Algoritmo Simulated Annealing

Donde T denota la temperatura y K_B es una constante física conocida como constante de Boltzmann [1].

En optimización un movimiento será aceptado si su costo disminuye; o en caso contrario, si su probabilidad de aceptación es mayor que un número aleatorio, este tipo de estrategias permiten que el algoritmo escape de óptimos locales. En la medida en que evoluciona el proceso, disminuye la temperatura, se incrementa la longitud de la cadena y por lo tanto, disminuye la probabilidad de aceptar soluciones de peor calidad.

Los parámetros a definir en el algoritmo son los siguientes: (i) temperatura Inicial calculada al inicio del proceso, en la primera cadena y una única vez. Su valor depende principalmente del problema tratado y de la probabilidad de aceptación. Valores muy elevados, hacen que el proceso tenga un mayor esfuerzo computacional, valores bajos pueden hacer que el algoritmo quede atrapado en soluciones de baja calidad al inicio del proceso. (ii) La tasa de enfriamiento para la cual existen valores típicos propuestos en la literatura especializada entre [0.9 – 0.5]. (iii) La temperatura final proveniente de un adecuado criterio de parada. (iv) Una longitud de la cadena de Markov que permita al proceso alcanzar el

cuasi equilibrio en cada nivel de temperatura. Una manera simple de seleccionar la longitud de la cadena es hacerlo de acuerdo con el tamaño del problema. (v) Una tasa de incremento de la cadena para la cual existen valores típicos sugeridos en la literatura entre [1.0 – 1.2].

La estructura de vecindad y el mecanismo de generación de alternativas se introducen a través de las siguientes dos definiciones:

i) Si se tienen dos soluciones, i (alternativa actual) y j (alternativa i modificada), con valores de la función objetivo $f(i)$ y $f(j)$, respectivamente. Entonces, el criterio de aceptación determina si j es aceptado como alternativa actual. Para ello, se aplica la siguiente probabilidad de aceptación:

$$P_c \{ \text{aceptar } j \} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ e^{-\frac{f(i)-f(j)}{c_k}} & \text{si } f(j) > f(i) \end{cases} \quad (8)$$

c_k corresponde al parámetro de control.

ii) Una transición es una acción combinada que transforma la alternativa de solución actual. Dicha acción consiste de los siguientes dos pasos: (a) aplicación del mecanismo de generación y (b) aplicación del criterio de aceptación.

Definiendo c_k como el valor del parámetro de control y L_k como el número de transiciones generadas en la k -ésima iteración del algoritmo de Metrópolis, se tiene el algoritmo de enfriamiento simulado mostrado en la figura 1. Los valores adecuados de los parámetros c_0 y L_0 para el problema de asignación generalizada se discutirán más adelante.

5. ALGORITMO GENÉTICO

Las principales características del AG de Chu-Beasley consisten en mantener constante el tamaño de la población de alternativas de solución. De manera que en cada iteración se reemplaza una alternativa de la población usando un eficiente mecanismo de modificación de la misma. Dicho mecanismo busca beneficiar las alternativas menos infactibles y de mejor calidad. En cada iteración la población es reemplazada sistemáticamente por un único descendiente generado. Esta estrategia tiene la ventaja de permitir encontrar múltiples soluciones.

5.1 Codificación

La codificación utilizada para representar cada una de las alternativas de solución del PAG consiste en construir un vector de tamaño m . Por lo tanto, la k -ésima posición del vector representa la k -ésima tarea y el elemento

almacenado en esa posición corresponde al agente que realiza dicha tarea. La población de alternativas de solución se conforma por un número determinado de cromosomas como el mostrado en la figura 2.

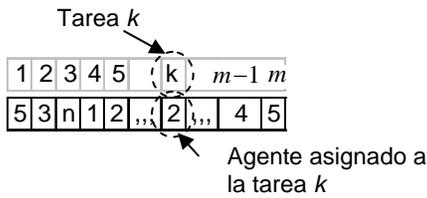


Figura 2. Codificación de Alternativas

5.2 Selección

En el proceso presentado por Chu-Beasley, son seleccionados 2 cromosomas padres para, posteriormente, ser recombinados de forma tal que, de los 2 hijos resultantes, sólo un hijo perdure en el proceso y pueda aspirar a ingresar a la población. En este proceso es necesario definir el número *k* de alternativas que serán escogidas aleatoriamente. Posteriormente, estas alternativas, compiten para seleccionar una alternativa padre. El proceso de selección utilizado debe realizarse dos veces de manera que se obtengan dos padres. se proponen dos algoritmos de competencia para la selección de dichos padres. El primero, denominado selección por ruleta, consiste en una escogencia basada en el inverso de valor de la función objetivo de cada alternativa ($1/Fobj_k$) donde las mejores tienen mayor probabilidad de ser seleccionadas como padre. Dicha probabilidad, asociada a cada alternativa, se calcula encontrando el porcentaje del valor anterior con respecto al valor que representa la función objetivo total el cual se calcula como:

$$FobjTotal = \sum_{i=1}^k \left(\frac{1}{Fobj_k} \right) \tag{9}$$

Se consideran los inversos, dado que se trata de un problema de minimización en donde se requiere asignar mayor probabilidad a las alternativas con menor costo de asignación.

El segundo algoritmo, denominado selección por torneo, es bastante simple y consiste en escoger la mejor alternativa de las *k* seleccionadas. El éxito de aplicar esta metodología radica en escoger adecuadamente un valor de *k* que se ajuste a cada problema en particular teniendo en cuenta el tamaño y complejidad del problema, así como el tamaño de la población inicial. Un valor de *k* muy alto y la generación de una población inicial pequeña puede, eventualmente, hacer elitista el proceso de selección y ocasionar convergencias locales.

5.3 Recombinación

Con los dos padres seleccionados, el siguiente paso consiste en combinarlos de forma tal que se obtenga solamente un descendiente. Lo anterior representa una diferencia significativa con respecto al método tradicional de AG y se traduce, en este caso, en una estrategia eficiente de búsqueda local.

En el proceso de recombinación es necesario definir el número *p* de puntos de recombinación. Dichos puntos son escogidos de forma aleatoria sobre el cromosoma. Posteriormente, se combinan las características de los padres haciendo un cruzamiento de las porciones de cromosoma existentes entre cada punto de recombinación como se muestra en la figura 3 para un cromosoma de ocho elementos y tres puntos de recombinación.

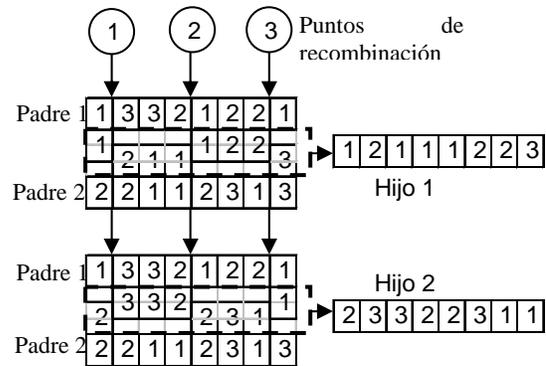


Figura 3. Ejemplo de recombinación de 3 puntos

El resultado de la recombinación produce dos hijos de los cuales uno es desechado de forma aleatoria, por ruleta o por torneo.

5.4 Mutación usando simulated annealing

El proceso de mutación se encuentra fuertemente ligado al concepto de vecindad. Así, una mutación drástica puede alejar el proceso de la región de búsqueda actual y producir oscilaciones alrededor de soluciones de buena calidad, es decir, se produce diversificación. Por otra parte, una mutación débil eventualmente puede ocasionar convergencias a soluciones locales de mala calidad, es decir se produce intensificación. Por lo tanto, el proceso de mutación permite una amplia gama de propuestas. El cromosoma hijo obtenido en el proceso de recombinación es sometido al proceso de mutación. El proceso de mutación es una subrutina que utiliza el algoritmo mostrado en la figura 1. En el proceso, se invierte un alto esfuerzo computacional en encontrar una modificación sugerida de la alternativa actual. Esto implica que el proceso consume mucho más tiempo computacional en cada iteración. Sin embargo, cada mutación mejora drásticamente la incumbente, por lo que el número de iteraciones totales generalmente es menor. Por lo tanto,

el esfuerzo computacional total del proceso puede ser comparable con el obtenido usando la mutación tradicional, siempre y cuando se garantice un adecuado cálculo de los parámetros L_0 y c_0 .

5.4.1 Cálculo de la temperatura C_k y la longitud de la cadena L_k

Una adecuada estrategia de enfriamiento requiere de un cálculo cuidadoso de estos parámetros. Para ello, es necesario especificar una secuencia finita de valores en el parámetro de control c_k , de manera que se establezca:

- **Un valor inicial de la temperatura C_0 .** Este valor inicial deberá, en teoría, ser lo suficientemente grande como para garantizar que un porcentaje establecido (generalmente el 80 %) de las modificaciones propuestas sean aceptadas. Para lograr esto, se comienza con un valor positivo pequeño de C_0 y se multiplica iterativamente por un factor constante mayor que 1 hasta que se establezca una *relación de aceptación* $X(C_0)$ lo más cercana posible a 1, es decir:

$$X(c_0) = \frac{\text{Número de Modificaciones Aceptadas}}{\text{Número de Modificaciones Propuestas}} \Big|_{c_0} \quad (10)$$

$$X(c_0) - 1 \leq \varepsilon$$

Donde ε es un valor tan pequeño como se desee. Este cálculo se realiza una única vez para todo el proceso.

- **Una función de decremento** que permita disminuir el valor del parámetro de control para simular el enfriamiento. Una forma frecuente de hacerlo es usando:

$c_{k+1} = \alpha c_k$ donde $k = 1, 2, \dots$, y α es generalmente un valor entre 0.8 y 0.99.

- **Una temperatura final**, especificada por un *criterio de parada* el cual puede ser: (i) la mutación termina si el valor de la función objetivo de la alternativa que está siendo modificada no cambia durante un número determinado de cadenas consecutivas. (ii) la mutación termina si el número de modificaciones aceptadas es menor a un valor dado.

Adicionalmente, es necesario especificar un número finito de transiciones de manera que se tenga:

- **Una longitud finita de la cadena de Markov L_k .** La longitud de la cadena de Markov se basa en el requerimiento de que en cada valor de C_k el equilibrio térmico sea alcanzado. Para lograr esto, la cadena de Markov deber ser larga para valores pequeños de C_k , es decir $L_k \rightarrow \infty$ para $c_k \downarrow 0$. El criterio, utilizado en este trabajo, para escoger el valor de la cadena se basa en: $L_k = \mu(m \times n) = \mu(\text{Tareas} \times \text{Agentes})$ donde μ es un parámetro de calibración, que permite ampliar o reducir

la cadena. Otra estrategia de enfriamiento para calcular los parámetros L_k y C_k bastante utilizada, se fundamenta en la metodología presentada en [2] denominada *Estrategia de Enfriamiento en un Tiempo Polinomial* la cual no se discutida en este documento.

5.5 Disminuir infactibilidad

El modelo, tal como se plantea en este trabajo, lleva en cuenta la infactibilidad de forma implícita al considerar factores de penalización. Adicionalmente, el algoritmo que se propone incluye un proceso que busca disminuirla gradualmente. Para alternativas infactibles, el proceso consiste en reasignar una tarea perteneciente a un agente violado a otro agente con capacidad para realizarla. Una alternativa es infactible cuando uno o más de sus agentes violan los límites máximos de consumo de recursos. Es importante que el agente que asume la nueva tarea no pase a violar el límite máximo de recursos. La primera reasignación que disminuya la infactibilidad y no viole ninguna restricción es la que se toma como definitiva. La infactibilidad se cuantifica a través de la siguiente expresión:

$$\sum_{i \in \left\{ \begin{array}{l} \text{restricciones} \\ \text{violadas} \end{array} \right\}} \left(\alpha_i \left(\underbrace{\sum_{j=1}^n A_{ij} X_{ij} - b_i}_{\text{cantidad de recursos excedidos}} \right) \right) \quad (11)$$

Este proceso puede ser realizado consecutivamente dos o más veces para eliminar más rápidamente las configuraciones infactibles.

5.6 Modificar la población actual

El algoritmo genético completo, después de generada la población inicial, es el siguiente:

Repetir un número determinado de iteraciones los pasos del 1 al 5.

1. Se obtienen 2 alternativas padre por Selección de la población actual.
2. Se obtiene una alternativa hijo aplicando Recombinación a los padres obtenidos en el paso anterior.
3. Se obtiene una alternativa modificada aplicando Mutación.
4. Si la configuración es infactible se mejora la infactibilidad y se obtiene una alternativa menos infactible, de lo contrario, ir al paso 5.
5. Si la alternativa resultante de aplicar los pasos anteriores no se encuentra en la población, entonces aplicar estrategia de modificación de la población, sino, volver al paso 1.

Para modificar la población se propone la siguiente estrategia:

(i) Si la alternativa actual es infactible y a su vez es menos infactible que la peor infactible de la población, entonces reemplazar la peor infactible en la población actual. (ii) Si la configuración es factible y existe por lo menos una infactible en la población actual, entonces reemplazar la más infactible. (iii) Si la configuración es factible y toda las alternativas de la población actual son factibles, entonces reemplazar la alternativa con peor función objetivo por la alternativa actual. Lo anterior se realiza sólo si la alternativa actual es de mejor calidad que la peor de la población.

La estrategia de modificación de la población actual se realiza cambiando sólo una alternativa por iteración y teniendo en cuenta que no se admiten alternativas repetidas. Lo anterior evita convergencias prematuras y asegura una exploración detallada de la región de soluciones. Adicionalmente se pueden obtener múltiples soluciones de un mismo problema.

Esta estrategia busca preservar las mejores alternativas, asegurando factibilidad y optimalidad. Estas características constituyen la principal diferencia con respecto al algoritmo propuesto por Chu-Beasley, en el cual la alternativa más infactible es reemplazada. A diferencia de los algoritmos genéticos tradicionales, no se modifica la población de forma aleatoria.

6. RESULTADOS OBTENIDOS

Tipo	n	m	Mejor Solución Chu-Beasley	Mejor Solución AG-Híbrido	Número de Soluciones Encontradas
A	5	100	1698	1698	32
A	20	200	2339	2339	18
B	5	100	1843	1843	10
B	5	200	3553	3553	15
C	5	100	1931	1937	5
C	10	200	2814	2814	10
D	5	100	6373	6360	2
D	10	200	12601	12563	1
E	5	100	12680	12690	1
E	10	200	23307	23315	1

Tabla 1. Resultados Obtenidos

En la tabla 1 se comparan los resultados obtenidos con las mejores soluciones conocidas publicadas en [4]. Los casos de prueba pueden ser descargados de :
<http://mscmga.ms.ic.ac.uk/jeb/orlib/gapinfo.html>
<http://www-or.amp.i.kyoto-u.ac.jp/~yagiura/gap/>.

Estos problemas típicos han sido clasificados en 5 grupos (A-B-C-D-E) según su nivel de dificultad. Cada uno de estos grupos contiene 6 problemas según su tamaño. De manera que existen 3 problemas con $m = 100$ y $n \in \{5, 10, 20\}$ y otros 3 problemas con $m = 200$ y

$n \in \{5, 10, 20\}$. Los problemas tipo A son considerados los más sencillos y los tipo E los más complejos.

7. CONCLUSIONES Y RECOMENDACIONES

El algoritmo propuesto presenta resultados interesantes si se considera que permite obtener múltiples soluciones para un mismo problema. Así mismo, la estrategia que se propone en este documento para modificar la población actual y para realizar la mutación (SA) presenta un alto desempeño ya que se obtuvieron respuestas mejores que las obtenidas por Chu-Beasley [4].

La cuantificación de la infactibilidad se realiza de forma diferente a como se plantea en [4]. En este trabajo se consideran factores de penalización y el proceso de disminución de infactibilidad. Por lo tanto, es posible hacer perdurar por más o menos tiempo las alternativas infactibles si se le da un valor adecuado al parámetro λ . Lo anterior puede mejorar la exploración del método y adquiere mayor importancia en problemas de alta complejidad en los cuales el espacio de soluciones es reducido.

9. BIBLIOGRAFÍA

- [1] AARTS, E., KORST, J. Simulated Annealing and Boltzmann Machines, John Wiley & Sons, New York, 1990
- [2] AARTS, E.H.L, AND P.J.M. VAN LAARHOVEN, A new Polynomial time Cooling schedule, Proc. IEEE Int. Conf. On Computer-Aided Design, Santa Clara, 1985
- [3] BACK T. Evolutionary Algorithms in Theory and Practice. Oxford University Press, 1996.
- [4] BEASLEY, J.E. E CHU, P. C. A Genetic Algorithm for the Generalized Assignment Problem. Computers Operations Research, 24(1), pp 17-23, 1997.
- [5] GRANADA M., TORO E. M., TABARES P. Método de Colonia de Hormigas Aplicado a la Solución del Problema de Asignación Generalizada. Revista Tecnura No 15, Universidad Distrital F.J.C., II-2004.
- [6] TORO E. M., GRANADA M. Solución al Problema de la Asignación Generalizada Usando el Método de Búsqueda Tabú. Revista Scientia et Technica (24), 61-67, U.T.P., Colombia 2004.