



Dyna

ISSN: 0012-7353

dyna@unalmed.edu.co

Universidad Nacional de Colombia

Colombia

MORENO, JULIAN; RIVERA, JUAN CARLOS; CEBALLOS, YONY FERNANDO
AGRUPAMIENTO HOMOGÉNEO DE ELEMENTOS CON MÚLTIPLES ATRIBUTOS MEDIANTE
ALGORITMOS GENÉTICOS

Dyna, vol. 78, núm. 165, febrero-marzo, 2011, pp. 246-254

Universidad Nacional de Colombia

Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=49622372026>

- 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

AGRUPAMIENTO HOMOGÉNEO DE ELEMENTOS CON MÚLTIPLES ATRIBUTOS MEDIANTE ALGORITMOS GENÉTICOS

HOMOGENEOUS GROUPING FOR MULTI-ATTRIBUTE ELEMENTS USING GENETIC ALGORITHMS

JULIAN MORENO

Escuela de Sistemas, Universidad Nacional de Colombia - Sede Medellín, jmoreno1@unalmed.edu.co

JUAN CARLOS RIVERA

Departamento de Matemáticas, Universidad Eafit, jriversa6@eafit.edu.co

YONY FERNANDO CEBALLOS

Departamento de Ciencias Básicas, Universidad de Antioquia, fceball@udea.edu.co

Recibido para revisar Febrero 17 de 2009, aceptado Septiembre 14 de 2009, versión final Septiembre 17 de 2009

RESUMEN: Este artículo describe el problema general de agrupamiento, particularmente aquel en el que se busca conformar grupos de igual tamaño y equitativos respecto a más de un atributo, como un problema de optimización multi-objetivo, cuya solución por medio de una búsqueda exhaustiva no siempre será conveniente dada la explosión combinatoria que puede presentarse. Como alternativa a esta situación, se propone un método basado en algoritmos genéticos donde las soluciones posibles se codifican en estructuras tipo cromosoma a manera de matrices y donde por medio de un proceso iterativo en el que intervienen los operadores genéticos de selección, cruce y mutación, se guía el proceso de búsqueda hasta dar con una solución satisfactoria.

PALABRAS CLAVE: Agrupamiento, Optimización, Algoritmos genéticos

ABSTRACT: This paper describes the grouping general problem, particularly when more than one attribute is considered and fair groups with same size are needed, as a multi-objective optimization problem whose solution using an exhaustive search is not always feasible due to the combinatory explosion that may be present. As an alternative to this situation a genetic algorithms based method is proposed where the possible solutions are codified in chromosome like structures using matrixes and through an iterative process where the selection, crossover and mutation genetic operators are used in order to guide the searching process until reaching a satisfactory solution.

KEYWORDS: Grouping, Optimization, Genetic algorithms

1. INTRODUCCIÓN

El agrupamiento de elementos es un problema combinatorio general que consiste en la repartición de un total de elementos entre un número definido de grupos, generalmente del mismo tamaño, de tal manera que se satisfaga una cierta condición. Aunque a primera vista parezca muy simple, la complejidad de este problema se focaliza principalmente en dos aspectos. El primero se refiere a la condición que debe ser satisfecha, la cual

en el caso más común se trata de obtener grupos “equitativos” u homogéneos considerando una cierta medida de valor para cada elemento. Para ejemplificar un caso muy simple, supóngase que en un salón de clase se desea formar varios grupos de estudio. En este caso, un método sencillo de alcanzar cierta homogeneidad (académicamente hablando) es ordenar los

estudiantes de mayor a menor según su promedio acumulado y empezar a asignar uno a cada grupo de manera secuencial. Sin embargo, ¿que sucede cuando de cada elemento no se tiene en cuenta solamente un atributo si no varios y se desea que cada grupo sea equitativo considerándolos todos? Peor aún, ¿que pasa cuando tales atributos no son proporcionales entre sí? En estos casos el criterio de repartición ya no es trivial, si no que requiere de alguna búsqueda inteligente que permite encontrar una solución que satisfaga la condición requerida.

El segundo aspecto es la explosión combinatoria que va de la mano con el número de elementos totales que se tengan y la cantidad de grupos que quieran formarse. De manera general el número de grupos diferentes de q elementos que pueden obtenerse a partir de un conjunto total de p elementos ($q \leq p$), considerando relevante el ordenamiento de los grupos, es:

$$\binom{p}{q} = \frac{p!}{(p-q)!q!} \quad (1)$$

Así por ejemplo si se desea repartir 100 elementos en grupos de a 20, este valor ascendería aproximadamente a $5,36E+20$ posibles combinaciones, lo cual hace que encontrar la mejor solución a partir de una búsqueda exhaustiva sea poco factible en muchos casos.

Al considerar estos dos aspectos en conjunto es fácil intuir que este problema puede ser tratado como una optimización multi objetivo, donde cada objetivo consiste en alcanzar el mayor nivel de similaridad posible entre el promedio de cada grupo respecto a cada atributo y el promedio de la totalidad de los elementos.

Una forma de atacar este problema como se mencionó anteriormente es mediante una búsqueda exhaustiva, lo cual dependiendo de los valores de p y q no siempre será posible dadas las limitaciones de cómputo. En estos casos métodos de búsqueda heurística pueden ser una buena alternativa pues, aunque estos métodos no garantizan hallar la solución óptima, generalmente si encuentran una que sea satisfactoria, empleando para ello un esfuerzo de cómputo considerablemente menor. Entre estos métodos se pueden encontrar [1-3]:

búsqueda local, búsqueda tabú, algoritmos genéticos, siendo este último el objeto de estudio de este trabajo.

El resto de este artículo está organizado de la siguiente manera: en la siguiente sección se presenta una breve introducción a los algoritmos genéticos y su aplicación en problemas combinatorios. En la sección 3 se describe el método propuesto, mientras que en la sección 4 se muestran los resultados obtenidos mediante varios casos de prueba. Finalmente en la sección 5 se presentan las conclusiones.

2. ALGORITMOS GENÉTICOS

Los algoritmos genéticos - AG fueron descritos por Holland [4] y son considerados como una familia de modelos computacionales inspirados por los principios de evolución de Darwin. Son generalmente asociados con optimizadores de funciones, aunque el espectro de problemas en los que pueden ser aplicados es bastante grande [5]. La característica común de estos algoritmos es que codifican las potenciales soluciones del problema que atacan mediante una estructura de datos, generalmente un vector, a manera de cromosoma y aplican operadores de recombinación buscando preservar la información crítica que guía a una solución satisfactoria [6].

Con el fin de brindar una idea más clara del esquema general de estos algoritmos se presenta la figura 1. Aquí se puede observar que se parte de una población inicial de individuos, típicamente aleatorios, donde un individuo se entiende como una solución posible (no necesariamente buena). Cada individuo se representa entonces con un cromosoma compuesto por genes donde cada gen hace referencia a una parte o secuencia de tal solución. Luego dichos individuos son evaluados mediante una determinada función de aptitud y se aplican unos operadores genéticos para obtener una nueva población (siguiente generación). El objetivo de estos operadores es preservar los cromosomas, o

parte de ellos, que representen mejores soluciones siguiendo para esto los principios de selección natural, de manera que aquellos individuos más aptos tienen más posibilidades de preservar sus genes sobre aquellos que no. Una explicación más detallada de estos procesos, así como su aplicación en el problema de estudio, se encuentra en la siguiente sección.

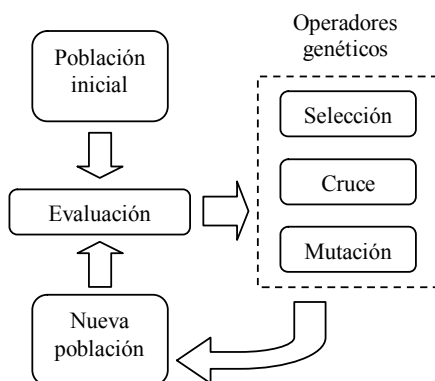


Figura 1. Esquema general de AG
Figure 1. GA general schema

Respecto a su aplicación, los algoritmos genéticos han sido empleados, particularmente durante las dos últimas décadas, en una amplia gama de problemas de optimización combinatoria que incluyen el TSP (Travel Salesman Problem), KP (Knapsack Problem), secuenciamiento y programación de tareas (scheduling), enrutamiento de vehículos, entre otros [7-12], lo cual los convierte en un buen candidato para resolver el problema de agrupamiento planteado en este artículo.

3. FORMULACIÓN

En esta sección se describe la formulación matemática y algorítmica del método propuesto, desde la representación de los elementos (los que desean agruparse), de las soluciones (agrupaciones factibles) y sus medidas de aptitud, hasta los operadores empleados para la aplicación del algoritmo genético.

3.1 Representación de los elementos

Dado que una de las premisas fundamentales de este trabajo es que los elementos que se desea agrupar pueden tener más de un atributo, estos pueden ser representados por medio de un vector de la siguiente manera, siendo M el número de atributos:

$$E_n = \{Atr_1, Atr_2, \dots, Atr_M\} \quad (2)$$

Así por ejemplo, dos elementos diferentes considerando 5 atributos pueden visualizarse en la figura 2.

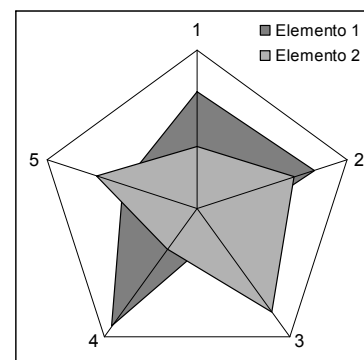


Figura 2. Ejemplo de dos elementos
Figure 2. Two elements example

Esta representación requiere que todo atributo m ($1 \leq m \leq M$) sea un escalar en un rango definido. Esto no quiere decir que atributos categóricos no puedan ser considerados, pues en este caso solamente se necesitaría un proceso previo de discretización. Por ejemplo, si un atributo puede tomar los valores “alto”, “medio”, “bajo” y “muy bajo”, estos podrían cambiarse por 1, 2, 3 y 4 respectivamente.

A partir de esto, y considerando que cada atributo puede estar en una escala diferente, el total de elementos puede representarse mediante una matriz de $M \times N$ como el ejemplo que se muestra en la tabla 1.

Una vez se tienen los datos de esta manera, estos deben ser escalados a un rango común con el fin que no se presenten perturbaciones en el cálculo de la función objetivo.

Una forma simple de hacer esto, de manera que todos los datos queden en el rango 0 - 1, es aplicando la fórmula:

$$Z^* = \frac{Z - Z_{\min}}{Z_{\max} - Z_{\min}} \quad (3)$$

Siendo Z_{\max} y Z_{\min} los valores máximo y mínimo del atributo correspondiente.

Tabla 1. Representación de un conjunto total de elementos

Table 1. Representation of an elements total set

ID	Atr ₁	Atr ₂	...	Atr _M
1	80	0.4		15
2	30	0.73		-8
...				
N	55	1.12		23

3.2 Representación de los individuos

Como se mencionó previamente, un individuo en el contexto de los algoritmos genéticos es entendido como una solución factible del problema analizado representada mediante una estructura de datos a manera de cromosoma. En el caso del agrupamiento, un individuo corresponde a una colección determinada de G grupos cada uno con hasta N/G elementos, siendo N el total de elementos. En la mayoría de trabajos con algoritmos genéticos, la estructura de datos empleada es un vector donde cada posición corresponde a un gen de la solución. En el método presentado en este trabajo se propone utilizar una matriz donde el número de filas corresponde al número de grupos deseado G y el de columnas al tamaño máximo de cada grupo N/G . De esta manera cada gen que compone el cromosoma contiene el identificador de un elemento, y su posición dentro de la matriz define el grupo al que pertenecería. Esta representación, además de la simplicidad que supone y de su obvia interpretación, facilita el uso del operador genético de cruce propuesto más adelante.

Para este problema, al igual que en otros como los mencionados en la sección 2 (TSP, KP, secuenciamento y programación de tareas, etc.), un cromosoma no puede tener genes repetidos, lo que significa que un individuo (solución) factible es aquel en el que cada elemento está en una única posición del cromosoma. Así por ejemplo si se tiene

un total de 20 elementos y se desea formar 4 grupos, cada uno contendría exactamente 5 elementos. En este caso un posible individuo, si los elementos son numerados consecutivamente, podría ser como se presenta en la tabla 2.

Tabla 2. Representación de un individuo

Table 2. Individual representation

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

3.3 Medida de aptitud

Dado que el objetivo al que apunta este trabajo es obtener grupos homogéneos respecto a la totalidad de los elementos, es necesario definir una medida de esta homogeneidad. Una manera de hacerlo se describe a continuación.

Primero se calcula el promedio de cada atributo de la totalidad de los elementos:

$$PT = \{\overline{Atr_1}, \overline{Atr_2}, \dots, \overline{Atr_M}\} \quad (4)$$

Luego para cada grupo g (fila) de cada individuo se calcula el promedio de cada atributo. Como cada individuo i se representa como una matriz, llámese X^i , dichos promedios pueden obtenerse como:

$$P^i_g = \{\overline{X^i_{g,1}}, \overline{X^i_{g,2}}, \dots, \overline{X^i_{g,M}}\} \quad (5)$$

Posteriormente se calcula la sumatoria entre la diferencia al cuadrado respecto a los M atributos entre cada grupo del individuo i y el total de los elementos:

$$D_i = \sum_{g=1}^G \left[\left(\overline{Atr_1} - \overline{X^i_{g,1}} \right)^2 + \left(\overline{Atr_2} - \overline{X^i_{g,2}} \right)^2 + \dots + \left(\overline{Atr_M} - \overline{X^i_{g,M}} \right)^2 \right] \quad (6)$$

$$= \sum_{g=1}^G \left\| PT - P^i_g \right\|^2$$

A menor sea el valor de esta métrica (con un mínimo de 0), más similar será en promedio cada uno de los grupos respecto al total de los elementos. Para explicar más claramente esta medida, así como los conceptos explicados hasta este momento, considérese el siguiente ejemplo donde se tienen 4 elementos y 2 atributos como se muestra en la tabla 3.

Tabla 3. Elementos de ejemplo
Table 3. Example elements

ID	Atr1	Atr2
1	30	0,52
2	40	0,26
3	20	0,78
4	10	1,04

Luego de escalar estos valores según el procedimiento descrito al final de la sección 3.1 se obtiene la tabla 4.

Tabla 4. Valores escalonados
Table 4. Scaled values

ID	Atr1	Atr2
1	0,66	0,33
2	1	0
3	0,33	0,66
4	0	1

Supóngase que se desean formar 2 grupos, cada uno con dos elementos. Dos posibles individuos para este caso se muestran en la tabla 5.

Tabla 5. Individuos de ejemplo
Table 5. Example individuals

Individuo 1		Individuo 2	
1	2	1	3
3	4	2	4

Al aplicar (4) se obtiene que $PT = \{0,5, 0,5\}$, mientras que al calcular $P_{g,i}$ según (5) a partir de las tablas 4 y 5 se obtiene:

$$P^1_g = \begin{Bmatrix} 0,83 & 0,16 \\ 0,16 & 0,83 \end{Bmatrix} \text{ y } P^2_g = \begin{Bmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{Bmatrix}$$

Finalmente, calculando las medidas de aptitud según (6) se obtienen $D1 = 2,94$ y $D2 = 0$. En este caso se puede observar que el agrupamiento representado por el individuo 2 cumple perfectamente con el criterio de homogeneidad ya que cada grupo respecto a cada atributo es igual al promedio del total de elementos. Si bien en este ejemplo se

alcanza un óptimo global (la función de aptitud del mejor individuo fue 0) debido a que en este caso existe una clara proporcionalidad en los valores de los atributos, cabe recordar que esta situación no es usual en la mayoría de los problemas de agrupamiento multi atributo, y de allí la utilidad del método propuesto en este artículo.

3.4 Población inicial y evolución

En el ejemplo representado en la tabla 3 se muestra una conformación de grupos trivial: asignar cada elemento en orden a un grupo según el identificador que tengan. Esto es, que los primeros N/G elementos (en ese caso 5) pertenezcan al grupo 1, los siguientes N/G al 2 y así sucesivamente. Si bien esta conformación es válida, la idea de la población inicial es generar k individuos de manera aleatoria, utilizando la representación de matriz descrita en la sección 3.2 y cumpliendo la restricción que cada elemento debe estar en una y solo una de las posiciones de la matriz.

Una vez se obtiene la población inicial, y de acuerdo al esquema representado en la figura 1, se procede a llevar a cabo el proceso de evolución en el que se pasa de una generación a otra empleando los operadores genéticos descritos a continuación hasta obtener una medida de aptitud deseada o hasta que se alcance un total de h generaciones.

3.5 Operadores genéticos

Los 3 operadores genéticos básicos son [4]: selección, cruce y mutación. El primero, como su nombre lo indica, consiste en seleccionar los individuos más aptos, es decir con mejor función de aptitud, para que sus cromosomas o parte de ellos se conserven en la siguiente generación (bajo el principio de sobre vivencia del más fuerte o más adaptado). El segundo consiste en generar individuos “hijos” a partir de padres aptos, lo cual se logra combinando de alguna manera los cromosomas de dichos padres (bajo el principio de conservación de los genes

aptos). El último consiste en la cambio de uno o más genes de los individuos de la nueva generación (bajo el principio que dichos cambios a nivel de cromosoma pueden ser favorables para la población).

3.5.1 Selección

Una manera de llevar a cabo la selección de individuos es mediante el mecanismo denominado “rueda de ruleta” [13], en el cual los individuos de una generación pueden ser imaginados como porciones de una ruleta como se muestra en la figura 3, donde el área de cada porción A es proporcional a la función de aptitud f de dicho individuo X

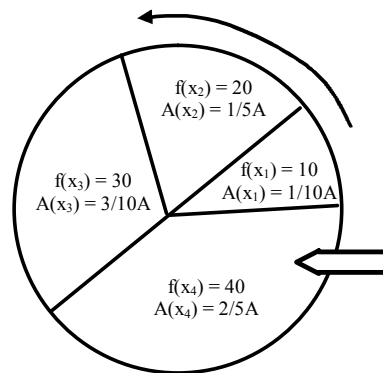


Figura 3. Ejemplo de ruleta
Figure 3. Roulette wheel example

Una vez definidas las porciones, o probabilidades de selección de cada individuo, la rueda se gira y donde se detenga dicho individuo es seleccionado. Este procedimiento se repite hasta que una proporción α de la generación sea seleccionado y sus cromosomas son “clonados” a la siguiente generación.

Si bien la figura 3 representa el caso en el que la función de aptitud busca ser maximizada, un procedimiento análogo puede usarse en el caso que se busque la minimización.

3.5.2 Cruce

Para llevar a cabo el cruce de genes entre cromosomas de varios individuos para producir uno o varios hijos existe una amplia gama de operadores de cruce, los cuales en su mayoría consideran la representación vectorial de los cromosomas. Algunos de los operadores más comunes en

problemas donde no se presentan genes repetidos son: cruce de dos puntos - 2X, cruce de orden linear - LOX, cruce mapeado parcialmente - PMX, cruce de ciclo - CX, operador C1, operador NABEL, cruce multi-padres - MPX, cruce de la subsecuencia común más larga - LCSX [14].

El operador de cruce propuesto en este artículo es una modificación del operador C1, el cual escoge un punto de cruce (aleatorio o fijo) entre los cromosomas de los padres, combina el primer segmento del primer padre con el segundo segmento pero en el orden en que los genes correspondientes aparezcan en el segundo padre y viceversa. La modificación requerida para poder aplicarse a la representación matricial consiste en que el punto de cruce, definido de manera aleatoria, se emplea como un corte a lo largo de la columna correspondiente, tal como se muestra en la figura 4. De esta manera, si por ejemplo el punto de cruce entre 2 individuos X^1 y X^2 es j ($0 \leq j \leq G/N$), los elementos $X^1_{i,b}$ ($j+1 \leq b \leq G/N$) se reordenarían según aparezcan en X^2 generando así el primer hijo, y de manera análoga se haría con X^2 para generar el segundo.

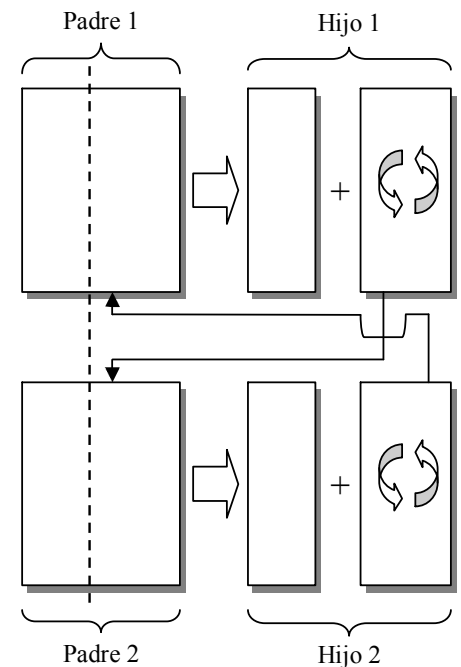


Figura 4. Operador de cruce
Figure 4. Crossover operator

En el método propuesto, y con el fin que el número de individuos entre generaciones permanezca constante, el restante $(1 - \alpha)$ porcentaje de cada nueva generación se obtiene mediante la operación de cruce descrita previamente seleccionando los padres por medio del mecanismo de ruleta.

3.5.3 Mutación

El proceso de mutación se lleva a cabo una vez se tienen definidos los individuos de una nueva generación, esto es, una vez se ha realizado la selección y el cruce. Los individuos a mutar, al igual que los genes que mutan son seleccionados de manera aleatoria mediante dos parámetros θ y ϕ . Así por ejemplo si estos parámetros toman los valores 0,15 y 0,05 significa que cada individuo tendrá una probabilidad de mutar de 0,15 y, para un individuo seleccionado para ser mutado, cada gen de su cromosoma tendrá una probabilidad de mutar de 0,05.

Dada la naturaleza del problema, la mutación de un gen consiste en el intercambio del valor de dicho gen por el valor de otro. Sin embargo, considerando la representación matricial empleada es necesario hacer la salvedad que el gen con el que se realiza el intercambio no puede estar situado en la misma fila pues el cambio no tendría incidencia alguna (el orden al interior de un grupo no tiene relevancia).

4. RESULTADOS

La validación del método propuesto fue llevada a cabo mediante varias pruebas, 3 de las cuales se presentan a continuación. En la primera prueba se simuló de manera aleatoria 50 elementos considerando 4 atributos. En esta prueba se definió como el número de grupos deseado 10, con lo que el número de elementos por grupo es 5. En este caso el número posible de combinaciones que podrían presentarse (agrupamientos diferentes) sería de:

$$\binom{50}{5} = \frac{50!}{(50-5)! \cdot 5!} = 2'118.760$$

Para el primer operador genético, la selección, se definió que un porcentaje α de 40% de individuos de la población se clonaría a la siguiente, mientras que

el operador de cruce se aplicó al 60% restante. El operador de mutación se aplicó a toda la población de cada generación con una probabilidad de mutación de un individuo θ de 0,2 y una probabilidad para cada gen ϕ de 0,15. Para cada generación se definió un número de individuos k de 80 y el criterio de parada del algoritmo fue hasta que se obtuviera un valor de la función de aptitud inferior 0,01 o hasta alcanzar $h = 1000$ generaciones. Bajo estos parámetros la ejecución del algoritmo programado en Java tardó 6,3 segundos (en un computador con procesador Intel Core 2 Duo de 2,4Ghz y 2Gb de RAM) y el mejor individuo obtenido tuvo una función de aptitud de 0,05911.

Una segunda prueba fue llevada a cabo con los mismos parámetros con la única diferencia que el número de grupos deseados fue de 5, con lo que el número de elementos por grupo sería de 10 con lo que el número posible de agrupaciones es de: 10.272'278.170. En este caso la ejecución del algoritmo tardó los mismos 6,3 segundos y el mejor individuo obtenido tuvo una función de aptitud de 0,007814.

Para la última prueba se consideraron igualmente 4 atributos pero se aumentaron los valores del número de elementos a 300 y de grupos a 15, con lo que el número de elementos por grupo es de 20 y el número de agrupaciones posibles asciende a 7,50043E+30. En este caso la ejecución del algoritmo tardó 1 minuto 44 segundos y el mejor individuo obtenido tuvo una función de aptitud de 0,03154.

La evolución de la medida de aptitud del mejor individuo durante las 1000 generaciones para las tres pruebas se muestra en la figura 5, evidenciando la convergencia del algoritmo.

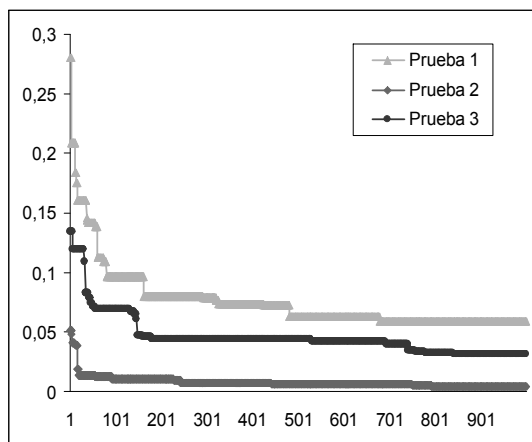


Figura 5. Evolución de la medida de aptitud

Figure 5. Fitness measure evolution

Estos resultados demuestran la efectividad del algoritmo planteado, el cual por tratarse de un método heurístico de búsqueda, no garantiza alcanzar el óptimo global pero sí un valor cercano a pesar de la simplicidad de su formulación y de su baja demanda de recursos computacionales (tiempo y memoria).

5. CONCLUSIONES

Considerando que el problema de obtener grupos homogéneos (equitativos) a partir de un conjunto de elementos donde se tienen en cuenta no sólo uno sino varios atributos, es difícil de resolver por métodos analíticos o de búsqueda exhaustiva debido a la explosión combinatoria que puede llegar a presentarse dependiendo del número de elementos y de grupos, un método de búsqueda heurística parece ser un buen candidato para solucionarlo.

Bajo esta premisa en este artículo se propone una medida de homogeneidad a partir de la diferencia cuadrada media entre el conjunto total de elementos y cada uno de los grupos. Adicionalmente, se propone un método basado en algoritmos genéticos donde los individuos (posibles soluciones) se codifican por medio de matrices que representan sus cromosomas y, mediante un proceso iterativo basado en los principios de evolución de las especies, se van obteniendo soluciones cada vez mejores hasta alcanzar un estado estable. Tal proceso involucra el uso de una serie de operadores

genéticos conocidos como selección, cruce y mutación, cada uno de los cuales es descrito en detalle en este artículo, así como su adaptación para trabajar en la forma matricial propuesta.

Con los resultados obtenidos a través de diversas pruebas se pudo comprobar la utilidad del método ya que logra obtener grupos bastante homogéneos (considerando la medida de aptitud propuesta) para múltiples atributos, incluso cuando el número de combinaciones posibles es muy elevado, sin que esto implique un elevado tiempo de cómputo.

Por último es necesario agregar que los campos de aplicación de este método son variados y aún están por explorar. Sin embargo, como ejemplo específico, cabe señalar que en la actualidad está siendo analizado para la conformación de grupos de trabajo en entornos colaborativos, donde se busca que dichos grupos estén “bien repartidos” considerando varias habilidades de los estudiantes como rendimiento académico, aptitudes colaborativas y perfiles de trabajo.

REFERENCIAS

- [1] GLOVER, F and KOCHENBERGER, G. Handbook of Metaheuristics. Kluwer Academic Publishers, 2003.
- [2] REEVES, C. Modern Heuristic Techniques for combinatorial Problems, John Wiley & Sons, New York, 1993.
- [3] RESENDE, M. and PINHO DE SOUSA, J. Metaheuristics: Computer Decision-Making. Kluwer Academic Publishers, Dordrecht, 2004.
- [4] HOLLAND, H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, 1992. First Published by University of Michigan Press 1975.

- [5] WHITLEY, D. A Genetic Algorithm Tutorial. Technical Report, CS-93-103. Department of Computer Science, Colorado State University. 1993.
- [6] GOLDBERG, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley: Reading, MA, 1989.
- [7] ALBA, E. and DORRONSORO, B. Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms. In Proceedings of the 4th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP 2004, pages 11–20, 2004.
- [8] GREFENSTETTE J. Genetic algorithm for the traveling salesman problem. Proceedings of the International Conference of Genetic Algorithm and their Applications, pp. 160–165. 1995.
- [9] KHURI, S. BÄCK, T. and HEITKÖTTER, J. The zero/one multiple knapsack problem and genetic algorithms. In Proceedings of the 1994 ACM Symposium on Applied Computing, pages 188–193, March 6–8, 1994.
- [10] MURATA, T., ISHIBUCHI, H. and TANAKA, H. Multi-objective genetic algorithm and its applications to flowshop-scheduling. Comput. Ind. Eng, 30, 957–968. 1996.
- [11] PONGCHAROENA, P., HICKS, C. BRAIDEN, P. and STEWARDSONB, D. Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products. International Journal of Production Economics. Volume 78, Issue 3, 311-322. 2002.
- [12] WHITLEY, D., STARKWEATHER, T. and D’ANN, F. Scheduling problems and traveling salesman. In Proceedings of the Third International Conference on Genetic Algorithms and their Applications, Arlington, VA, USA, 1989, pp. 133–140.
- [13] WEISE, T. Global Optimization Algorithms - Theory and Application. University of Kassel. 2008.
- [14] HEJAZI, R. and SAGHAFIAN, S. Flowshop-scheduling problems with makespan criterion: A review. International Journal of Production Research. v43. 2895-2929. 2005.