

Algoritmo Genético para el problema de la Calendarización de horarios: “Una nueva representación del individuo”

Melissa P. Castilla Pelayo Jacinto Pool Balam

Joel Trejo Sanchez

UNIVERSIDAD DEL CARIBE

Ingeniería en Telemática

Departamento de Ciencias Básicas e Ingeniería

25 de noviembre de 2008

Resumen

En este trabajo se describe una nueva representación para los individuos de un Algoritmo Genético aplicado al problema de la calendarización de horarios. El caso de estudio fue el programa de estudios de Ingeniería en Telemática de la Universidad del Caribe.

1. Introducción

En general, se puede describir la calendarización de horarios como el problema de asignar ciertos recursos como: salón, profesor y asignatura, sujetos a ciertas restricciones, dentro de un limitado número de periodos de tiempo y colocados con el fin de satisfacer un conjunto de objetivos lo mejor posible [1, 2]. Debido a su complejidad computacional se dice que pertenece al grupo de problemas N-P complejos [3, 4] por que no hay un método para resolverlo en un tiempo razonable (polinomial) conocido. En términos básicos un problema es de este tipo si es muy difícil de resolver o requiere demasiado tiempo y esfuerzo usando métodos exactos tal como es el caso de evaluar todos los posibles horarios[5].

Debido a la naturaleza de la calendarización de horarios, ésta se puede ver como un problema de optimización por que se busca minimizar la violación de restricciones y el tiempo necesario para la elaboración de los horarios de clases. Éste problema ha sido tratado de diferentes maneras, tanto así que Burke y Petrovic [6] clasifican las técnicas usadas de acuerdo a diferentes criterios: *aproximación basada en programación matemática, aproximación basada en coloración de grafos, métodos de agrupación, aproximaciones basados en restricciones, métodos meta-heurísticos, aproximaciones de múltiple criterio y aproximaciones de razonamiento basado en casos*. Dentro de estas agrupaciones se puede mencionar a los Algoritmos Genéticos (AG) y sus hibridaciones, Tabú Search, Memetics, Optimización por colonia de hormigas, Coloración de grafos y otros más listados en [1].

En este trabajo se presenta una propuesta de representación del horario (cromosoma) usado por el AG durante la calendarización de horarios de clases. El método usado se basa en algoritmos genéticos, así que estas posibles soluciones son llamadas cromosomas. Los AG son herramientas muy poderosas para resolver problemas de optimización, los cuales modelan los principios de la evolución [7]. Son capaces de encontrar globalmente soluciones óptimas incluso en los espacios de búsqueda más complejos. Operan sobre una población de posibles soluciones codificadas (individuos), las cuales son seleccionadas según su aptitud para ser las bases de la próxima generación de soluciones, esta nueva población se obtiene por combinación (cruza) o alterando (mutación) los individuos actuales [8], éste proceso se repite una y otra vez hasta encontrar la solución óptima.

La representación del cromosoma para los individuos de la población de posibles soluciones sobre la que va trabajar el AG es fundamental para el óptimo desempeño del mismo. Es por eso que al implementar un AG, primero se define la representación del cromosoma de los individuos, la función objetivo obtendrá la aptitud realizando evaluaciones sobre el cuerpo de cada individuo. La eficiencia del AG esta influenciado en mayor o menor medida en la facilidad que se tiene para evaluar los individuos, éste tema se tratará con más detalle en la sección 3.1.

2. Análisis de datos

Se ha tomado como caso de estudio al programa de estudios Ingeniería en Telemática de la Universidad del Caribe. Es importante mencionar que existen academias¹ internas que agrupan a profesores con dominio sobre un área en particular. De manera que cuando el encargado hace el horario, se asegura que los elementos profesor y asignatura de las tuplas(profesor, salón, asignatura) compartan el mismo perfil² en el horario de clases. Esta agrupación de elementos en perfiles es usada en el algoritmo propuesto en una fase de preprocesado que se menciona en la sección 3.3. Los elementos de la tupla cambian debido a la nueva representación del individuo que se propone. Los principales elementos para la representación del cromosoma de los individuos del AG son:

Asignaturas : Éstas se caracterizan por tener un perfil asociado al profesor que la imparte, pertenece a un periodo escolar en particular, y están clasificadas como elección libre y básicas. Tienen un máximo de cuatro horas a la semana.

Por ejemplo: Algoritmos y estructura de datos, es del ciclo 1, clasificado como básica, su perfil es Programación e ingeniería de software y tiene 3 horas a la semana.

Profesores : Éstos pueden ser de tiempo completo o de asignatura (laboran por horas), pertenecen a una o más academias. El perfil del profesor es importante en el proceso de asignación de una asignatura.

Por ejemplo: M.C Nancy Aguas García, es PTC y pertenece a la academia de Programación e ingeniería de software.

Salones : Éstos pueden ser laboratorios para clases prácticas o aulas de clases teóricas.

Horas : Los horarios se asignan de 7:00 horas a 22:00 horas, excepto de 14:00 a 16:00 hrs, dando un total de 13 horas asignables por día.

Días : Son 5 días a la semana, de lunes a viernes.

¹Las academias son Redes y Telecomunicaciones, Programación e ingeniería de software, Hardware y software de base y Matemáticas.

²El perfil indica la academia a la que pertenece la asignatura o el profesor en cuestión.

2.1. Análisis de restricciones

La asignación de periodos de tiempo, salones de clase y profesores esta sujeta a *restricciones* que dependen de la naturaleza de la institución y sus prioridades [9].

Se denominan *restricciones* a aquellas condiciones o políticas que se deben satisfacer para crear un horario de clases. Éstas están sujetas a condiciones físicas, subjetivas o normativas, ya sea del profesor, el salón, el tiempo, la asignatura o la universidad en general.

Las restricciones se clasifican en débiles o fuertes [9] de acuerdo al grado de importancia que representan para crear un horario de clases admisible para la Universidad, a continuación se listan y clasifican las necesarias.

2.1.1. Restricciones débiles

- Los profesores de asignatura pueden dar un mínimo de 1 y un máximo de 3 asignaturas por ciclo.
- Los profesores de tiempo completo pueden impartir un mínimo de 2 y máximo de 4 asignaturas.
- Las asignaturas de tipo prácticas serán asignadas a los laboratorios.
- Si una asignatura es tanto teórica como práctica le será asignada tanto un salón como un laboratorio.

2.1.2. Restricciones fuertes

- Los profesores de asignatura sólo pueden dar clase en horarios determinados.
- Los profesores de asignatura tienen preferencia en la asignación de horarios.
- Los profesores sólo pueden impartir asignaturas relacionadas con su perfil.
- Un salón sólo puede estar asignado a un profesor y a una asignatura en un periodo de tiempo determinado.

- Las asignaturas que sean de tipo teóricas serán asignadas en aulas de clase convencionales.
- Las asignaturas con tres horas a la semana estarán divididas en dos días, el primero de una hora y el segundo dos horas consecutivas o viceversa.
- Las asignaturas de cuatro horas a la semana estarán divididas en dos días, donde cada uno tiene dos horas consecutivas.
- Los grupos con identificador impar deben ser asignados a un horario en la mañana, y los grupos con identificador par deben ser asignados a un horario en la tarde.
- Un profesor no puede estar en dos lugares al mismo tiempo.

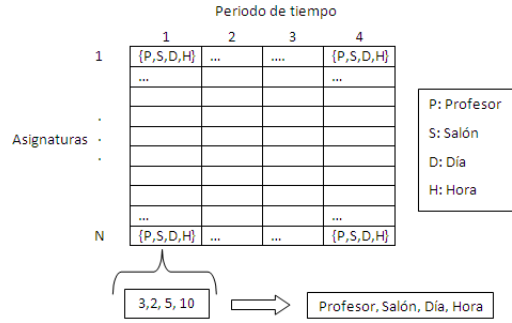
3. Descripción de la solución propuesta

El algoritmo utilizado es el básico para un Algoritmo Evolutivo el cual puede ser revisado en [10] y sus correspondientes operadores se describen en 3.3.

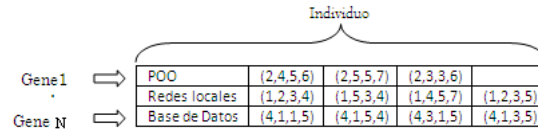
3.1. Codificación del individuo

La codificación del individuo es uno de los factores de mayor importancia para asegurar la rápida y correcta convergencia del AG. Para mantener la codificación de la información de la manera más simple se recurrió a utilizar claves numéricas que representen los datos a codificar. La representación del cromosoma que se propone en este trabajo se diseñó con el propósito de satisfacer los criterios que Beligiannis et al. [8] hace:

- Debe ocupar el menor espacio posible en relación con la información que codifica.
- Ser fácil de preservar durante el proceso de reproducción (selección, cruza y mutación) tanto como sea posible las características de los horarios que cumplan las restricciones fuertes.



Cuadro 1: Representación del individuo



Cuadro 2: Ejemplo de un individuo con información codificada

La codificación del cromosoma se describe como una matriz $C_{N,k}$ donde N es el # de asignaturas y k es cuatro (Cuadro 1). Ésta tiene cierta similitud con la utilizada para la representación del horario de clases en el proceso hecho por humanos. Pero en este caso, cada fila es un gene que indica la asignatura C_i a calendarizar y cada columna representa un periodo de tiempo de una hora en la cual puede ser asignable una asignatura, son cuatro columnas por que éste es el número de horas máxima para una asignatura. Cada C_{ij} contiene una tupla (P, S, D, H) donde el profesor P es un valor fijo para cada C_i , S representa el salón de clases, D es el día y H es una hora. Dado que una asignatura solo puede ser impartida por un solo profesor, en cada periodo de tiempo de la asignatura siempre tendrá al mismo profesor. Además, como existen asignaturas con tres horas a la semana, algunos genes tendrán un periodo de tiempo vacío en el cromosoma. Por ejemplo, al codificar la información, en alguna posición $C_{1,j}$ puede estar asignada la tupla $(2,5,5,10)$, la cual se interpreta como la asignatura 1 (Programación Orientada a Objetos) se asigna el profesor 2 (M.C. Flavio Reyes) en el salón 5 (C-06) el día 5 (viernes) en la hora 10 (6:00-7:00). En el cuadro 2 se muestra un ejemplo de un individuo para esta representación. Los valores para cada elemento de la tupla pueden ser obtenidos mediante las tablas de hashing como las que se muestran en las tablas de la figura 1.

Clave	Horario de clase	Clave	Profesor
1	7:00 – 8:00	1	M.C. Nancy Aguas
2	8:00 – 9:00	2	M.C. Flavio Reyes
.	.	.	.
.	.	.	.
.	.	.	.
13	21:00 – 22:00	M	M.C. Joel Trejo

a) b)

Clave	Salones	Clave	Día de clase
1	C-04	1	Lunes
2	C-05	2	Martes
.	.	.	.
.	.	.	.
.	.	.	.
S	C-15	5	Viernes

c) d)

Figura 1: Tablas de hashing

3.2. Función objetivo

La función objetivo es un factor importante en los AG. Éste se encarga de evaluar los individuos de la población y obtener su aptitud. Esta aptitud dice al AG que tan bien el cromosoma resuelve el problema determinado.

Para este caso, la función objetivo consta de evaluar la violación de restricciones tomando en consideración si son débiles o fuertes. Cada restricción tiene un costo, para las restricciones débiles vale uno y para las fuertes vale tres. La aptitud del cromosoma se obtiene sumando los costos de las restricciones violadas y ésta es inversamente proporcional a la suma de las violaciones, es decir, los cromosomas más aptos serán los que posean el menor número de violaciones. A continuación se describe:

- Sea D_r el conjunto de restricciones débiles con costo 1, sección 2.1.1.
- Sea F_r el conjunto de restricciones fuertes con costo 3, sección 2.1.2.
- Sea l el # de restricciones débiles.
- Sea L el # de restricciones fuertes.

1. Se define $\phi : C \times D_r \longrightarrow \{1, 0\}$, de tal manera que:

- $\phi(C_{n,k}, d) = 1$ si $C_{n,k}$ no cumple con d .

- $\phi(C_{n,k}, d) = 0$ si $C_{N,k}$ cumple con d .

2. Se define $\varphi : C \times F_r \longrightarrow \{1, 0\}$, de tal manera que:

- $\varphi(C_{n,k}, f) = 1$ si $C_{N,k}$ no cumple con f .
- $\varphi(C_{n,k}, f) = 0$ si $C_{N,k}$ cumple con f .

De esta forma finalmente la función objetivo se define:

$$\mathbf{FO}(C_{n,k}) = \sum_{q=1}^L [\phi(C_{n,k}, d_q)] +$$

$$\sum_{z=1}^l [\varphi(C_{n,k}, f_z) \times 3].$$

3.3. Los operadores Genéticos

Los operadores genéticos son los encargados del proceso de reproducción del AG, éstos se describen a continuación.

3.3.1. Operador de selección

Eiben y Smith [10] describen al operador de selección como un método para elegir a los cromosomas que se utilizarán como padres para crear nuevas descendencias, este método hace referencia a la selección natural. El método de selección usado es selección por torneo. Éste se eligió porque no necesita conocer a la población globalmente sino el rango de dos individuos, se basa mayormente en la aptitud relativa más que en la absoluta. Eiben y Smith [10] mencionan que la probabilidad de un individuo de ser seleccionado esta influenciada por las siguientes condiciones:

- Su posición en la población.
- El tamaño del torneo.
- La probabilidad p que el individuo más apto del torneo sea seleccionado.
- Si los individuos son escogidos con o sin remplazo.

El número de padres a seleccionar es de seis, de una población de doce, y el tamaño del torneo es de cinco elementos con selección sin reemplazo.

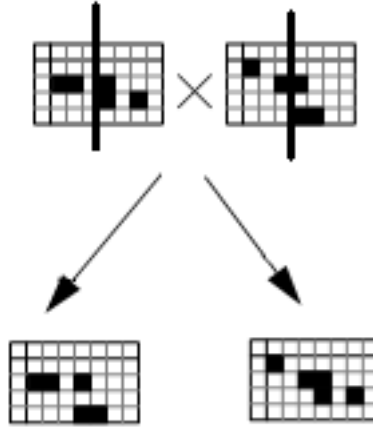


Figura 2: Ejemplo de cruza entre dos padres

3.3.2. Operador de cruza

El operador de cruza permite crear nuevos individuos basándose en la información de dos cromosomas padres, ésta hace referencia a la procreación de individuos [10].

El método de cruzamiento usado es de un punto [10]. En ésta, se elige un valor entre 1 y N-1, donde N es el número de genes(asignaturas), entonces se divide a los padres tomando como referencia de división del cromosoma el valor elegido, a continuación se intercambian las partes finales de los padres en ese punto(Figura 2). El tamaño de la descendencia es de seis.

3.3.3. Operador de mutación

El operador de mutación modifica los genes del cromosoma de los individuos hijos en base a una probabilidad [10]. Éste operador hace que la actual población varíe por medio de la descendencia.

3.3.4. Elitismo

Para el AG propuesto, este operador preserva el mejor cromosoma de cada generación, ya que en caso de que no se crearán cromosomas más aptos en la próxima generación, al menos se mantiene al individuo con mayor aptitud de la generación actual.

4. Resultados

Para probar el AG ingresaron: 13 Asignaturas, 8 Profesores y 6 salones. A partir de los experimentos realizados, se listan los parámetros del AG que dieron mejores resultados fueron:

1. Tamaño de población con 12 individuos.
2. Todos los padres se cruzan.
3. Probabilidad de mutación 0.05.
4. Elitismo para 1 individuo.
5. Número de generaciones 10,000.

Se puede observar en la figura 3 que para los parámetros definidos para AG, intenta converger, aunque tiene pequeñas variaciones que no permiten se defina correctamente la gráfica. Además se puede notar que a partir de la generación 2500, la aptitud promedio queda atrapado entre 22 y 26, este comportamiento continua hasta que se producen todas las generaciones.

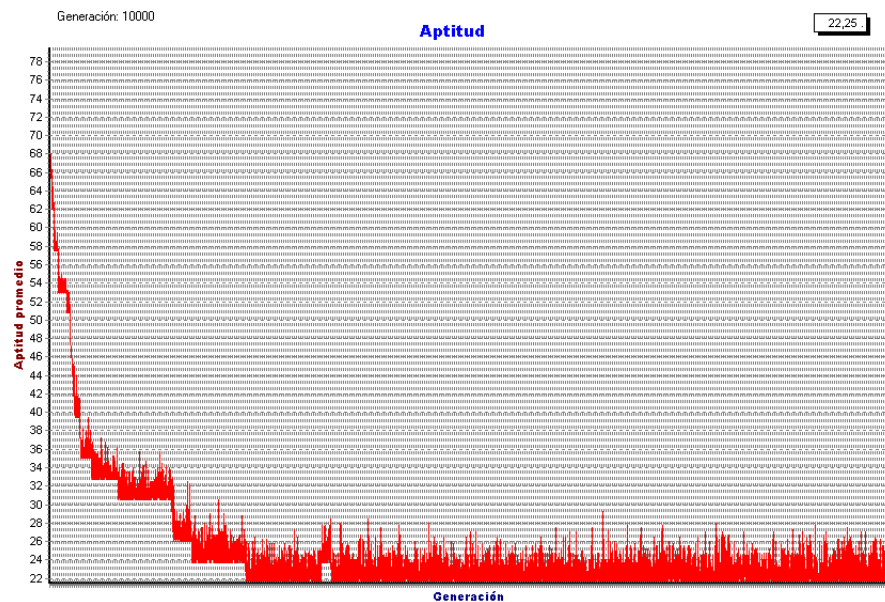


Figura 3: Gráfica de aptitud promedio para 10000 generaciones

5. Conclusiones

La representación presentada para el AG facilitó en gran medida la implementación del AG, esto se debe a la función objetivo evalúa de las restricciones de manera más rápida. A partir de las pruebas realizadas, se observó que si el tamaño de la población es mayor a 20, la aptitud promedio de cada generación varía demasiado y no se asegura su convergencia. Además, la probabilidad de mutación debe estar entre 0.01 y 0.05 para que el AG tenga un comportamiento más o menos convergente. Los resultados descritos anteriormente demuestran que la representación del cromosoma facilita su implementación, pero es necesario encontrar los parámetros adecuados del AG para asegurar su convergencia.

De acuerdo a la literatura revisada y a la experiencia adquirida, se afirma que el método utilizado cae en la categoría de los *métodos meta-heurísticos y aproximaciones basados en restricciones* porque el espacio de búsqueda es recorrido por un algoritmo genético y la aptitud de cada posible solución es obtenida revisando la violación de restricciones fuertes y/o débiles. Para más detalles consultar a Rossi-Doria et al. [11]. Se observó que no es posible generalizar el problema de la calendarización de horarios en una sola solución que lo resuelva para todas las instituciones educativas, ya que las restricciones se definen de manera diferente para cada una de ellas.

6. Trabajos futuros

Es necesario contrastar la propuesta de representación y el algoritmo con otros autores que han trabajado con AG para observar su desempeño y comportamiento. Por otro lado, se deben definir las probabilidades de cruce y mutación para los cuales el AG converja correctamente. Además, éste proyecto es la base para la construcción de un software calendarizador de horarios.

Referencias

- [1] S. Petrovic and E. Burke, “University timetabling,” in *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (J. Leung, ed.), Chapman Hall/CRC Press, 2004.

- [2] A. Wren, “Scheduling, timetabling and rostering - a special relationship?,” pp. 46–75, 1996.
- [3] S. Gyori, Z. Petres, P. Baranyi, and A. R. Vrkonyi-kczy, “Genetic algorithms in timetabling. a new approach,” 2001.
- [4] T. Cooper and J. Kingston, “The complexity of construction problems,” in *Practice and theory of automated timetabling: Selected papers from the first international conference* (E. Burke and P. Ross, eds.), vol. 1153, (Berlin, Heidelberg), pp. 183–295, Springer-Verlag, 1996. Lecture of notes in computer science.
- [5] A. Eckersley, *Novel Knowledge Based and Heuristic Approaches to University Timetabling*. PhD thesis, University of Nottingham, December 2007.
- [6] E. K. Burke and S. Petrovic, “Recent directions in automated timetabling,” *European Journal of Operational Research, EJOR*, vol. 140, no. 2, pp. 266–280, 2002.
- [7] D. L., “Handbook of genetic algorithms,” Van NOstrand Reinhold, 1991.
- [8] B. et al., “Applying evolutionary computation to the school timetabling problem: The greek case,” *Computers & Operations Research*, vol. 10, July 2006.
- [9] M. Rahoual, R. Saad, and U. M. Bougara, “Solving timetabling problems by hybridizing genetic algorithms and tabu search,” in *PATAT 2006* (E. Burke and R. H., eds.), pp. 467–472, PATAT, 2006.
- [10] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 3th ed., 2003.
- [11] O. Rossi-doria, M. Sampels, M. Birattari, M. Chiar, M. Dorigo, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Sttzle, “A comparison of the performance of different metaheuristics on the timetabling problem,” in *In: Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT)*, pp. 329–351, 2002.