

Organización de Asignaturas en una Facultad mediante Algoritmos Genéticos

Fernando Nellmédin y Mariano Peyregne

Universidad Nacional del Litoral, Facultad de Ingeniería y Ciencias Hídricas

Resumen—El problema de acomodar los horarios de una facultad, llamado *TimeTabling Problem* en inglés, es una difícil tarea que se debe realizar en cualquier facultad o universidad. Se tienen un conjunto de recursos limitados tales como la cantidad de profesores, alumnos o aulas, y se los debe asignar a cada una de las asignaturas que se cursan en la institución de manera de minimizar el impacto negativo tanto en los alumnos como en los docentes. Se quieren evitar situaciones tales como que se cursen simultáneamente materias del mismo año o de años adyacentes, ya que con frecuencia hay estudiantes que pueden asistir a asignaturas de distintos años. En este paper abordamos el *Timetabling Problem* utilizando un Algoritmo Genético y modelando las restricciones en una simple función de fitness. Si bien no consideramos todas las requisitos que participan en este problema, encontramos resultados satisfactorios con respecto al programa semanal obtenido. Primero presentamos el modelado del problema, luego los operadores definidos para el algoritmo genético y el fitness. Por último se presentan los resultados obtenidos y algunos ejemplos.

Palabras clave—*timetabling problem*, restricciones, algoritmo genético, optimización

I. INTRODUCCIÓN

La distribución de los horarios en que se dictan las asignaturas de la facultad se debe actualizar todos los años e implica un trabajo iterativo hasta que se cumplan todas las restricciones, o al menos se minimicen la cantidad de conflictos. Entre los requisitos a cumplir se encuentran los inherentes a los planes de las carreras, tales como que no se asignen materias del mismo año al mismo horario, o incluso de años consecutivos, ya que muchos alumnos asisten a asignaturas de distintos años por problemas de correlatividades. Asimismo, otras limitaciones surgen por la disponibilidad de aulas (y sus correspondientes equipamientos y límite de alumnos) y los docentes asignados a cada clase. Sobre este último, con frecuencia surge que cada profesor impone una restricción del tipo: “sólo de mañana” o “los lunes y martes únicamente”. De la definición de todas estas limitaciones, vemos que encontrar una distribución de clases durante la semana se torna una tarea casi imposible para cualquier ser humano.

En este trabajo, nos enfocamos en solucionar el *Timetabling Problem* (TTP) para un conjunto limitado de restricciones. Además, simplificamos la definición del problema.

Especializamos el trabajo en nuestra propia carrera: Ingeniería en Informática. Por lo tanto, modelamos las asignaturas de los 5 años de la misma. También se resumieron las clases de cada materia a 3: una de teoría de dos horas, y dos de práctica de tres horas cada una (a las que los alumnos deben acudir a ambas). En total hay 20 materias que se dictan en un semestre, de 3 clases cada una, lo que suma un total de 60 clases a ubicar. Un ejemplo

para las primeras asignaturas se ve en la Tabla I. Luego se estudiará el caso más general.

TABLA I
ASIGNATURAS MODELADAS COMO CLASES

Asignatura	Clase	ID
Cálculo II	Cálculo II (Teoría, 2 horas)	1
Cálculo II	Cálculo II (Práctica 1, 3 horas)	2
Cálculo II	Cálculo II (Práctica 2, 3 horas)	3
Física I	Física I (Teoría, 2 horas)	4
Física I	Física I (Práctica 1, 3 horas)	5
Física I	Física I (Práctica 2, 3 horas)	6

No se tuvo en consideración la distribución de los profesores ni sus restricciones añadidas, por lo que se asume que cada docente tiene a su cargo una sola clase y puede darla en cualquier franja horaria. En nuestra institución, como en muchas, se dicta más de una carrera. En lugar de modelar la distribución de clases y aulas para todas las carreras, se tomó un camino alternativo: se limitarán las aulas ocupadas por nuestra carrera, de manera que sólo se puede ocupar determinado número de aulas en simultáneo.

Por el lado de las restricciones, elegimos las siguientes a satisfacer:

- *Limitación de aulas*: No se pueden utilizar más de 4 aulas al mismo tiempo
- *Asignaturas en un mismo día*: No se desea que la misma asignatura, aún de clases distintas, se dicte dos veces en el mismo día.
- *Solapamiento en años adyacentes*: No se deben superponer las clases de asignaturas de un año, con las clases del año anterior o siguiente, para dar la posibilidad a los alumnos de cursar ambos semestres.
- *Superposición en un mismo año*: No es posible que se dicten dos clases del mismo año en el mismo momento, ya que imposibilita a los estudiantes a hacerse presente en ambas clases.

Con respecto a la distribución de horarios, se divide cada día, de lunes a viernes, en 5 módulos o franjas horarias. A cada módulo se le asigna un ID único, que servirá para identificarlo en el algoritmo genético. En la Fig. 1 se aprecian los módulos y el ID asignado a cada uno. El uso de estos identificadores se explicará en la próxima sección.

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
8 a 10	0	5	10	15	20
10 a 13	1	6	11	16	21
14 a 16	2	7	12	17	22
16 a 18	3	8	13	18	23
18 a 21	4	9	14	19	24

Fig. 1. Módulos definidos para la semana

II. ALGORITMO GENÉTICO

Para la solución del problema, se utilizó un AG en el que se utilizó un fenotipo como se muestra en la Fig. 2. De acuerdo a los ID definidos para cada clase en la Tabla I, se le asigna una posición específica en el fenotipo a cada una. Así, la clase con identificador igual a 1, será la primera en el fenotipo. Lo mismo se aplica a ID = 2, que estará ubicada segunda, y lo mismo con el resto. En cada posición del fenotipo, se almacena un número que referencia en qué bloque está situada. Siguiendo con el ejemplo de la Fig. 2, se aprecia que la clase 1 se dictará en el módulo 12, que al consultar la Fig. 1 vemos que corresponde a los miércoles a las 14 horas. A partir de la representación fenotípica, se codifica en un genotipo binario de 300 bits de extensión, ya que se toman 5 bits para cada elemento del fenotipo (que está en el rango 0-24).

12	13	16	1	3	...	20
----	----	----	---	---	-----	----

Fig. 2. Representación fenotípica

A. Inicialización

Se parte de una población inicial aleatoria. Para generarla, se toman las clases de a una y se le asigna un bloque aleatorio en el rango 0-24. De esta forma, se comenzará con muchos individuos con una evaluación de fitness pobre, ya que no se espera que cumplan ninguna restricción al inicio, pero todos serán válidos en el sentido de que no se asignarán bloques fuera del rango 0-24. Esta propiedad se mantendrá durante todo el algoritmo.

B. Selección de individuos para la siguiente generación

La población siempre se mantiene en el mismo tamaño. En cada generación, se deben elegir a los individuos que formarán parte de la generación siguiente. Se utilizó el método de selección por ventanas. A los individuos elegidos se les aplicará, si corresponde, las cruza y mutaciones que se describen en la próxima sección. Además, a través de elitismo, siempre se pasa el mejor individuo (según su fitness) a la próxima generación. Por último, se elige un término de brecha generacional del 10% del tamaño de la población. Por lo tanto, sin importar el fitness, se elige aleatoriamente cierta cantidad de individuos que serán integrantes del próximo conjunto. De esta manera, nos aseguramos que siempre haya diversidad en el conjunto de búsqueda.

C. Operadores genéticos

Se definieron dos operadores de mutación y uno de cruza. Estos trabajan en representación genotípica, pero cuidan de realizarse en posiciones que se corresponden con el fenotipo, de manera de no generar individuos inválidos luego de la operación. Cada uno de estos operadores tiene una probabilidad de aplicarse definida al inicio, por lo que puede pasar que ambas mutaciones se apliquen y la cruza no, o viceversa o cualquier otra combinación.

C.1. Cruza: Se utiliza una cruza de dos puntos. Se cuenta con dos padres y se generan aleatoriamente dos posiciones en el genotipo de ambos. Luego se intercambia el contenido genético entre éstos. Las posiciones elegidas se corresponden con los límites de cada elemento del fenotipo, de manera de no generar individuos con bloques de horarios inválidos.

C.2. Mutación por movimiento: Se genera una posición en el fenotipo al azar, que identificará a un bloque donde está asignada cierta asignatura. A continuación se buscará un módulo que se encuentre libre y que sea del mismo tamaño (2 o 3 horas según corresponda). Si se encuentra un espacio libre (después de un número limitado de iteraciones), se moverá la clase al nuevo bloque. En la Fig. 3 se ve un ejemplo.

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
8 a 10	Mat	Química			
10 a 13		POO	APS	Mat 2	PDS
14 a 16			IComp		
16 a 18					
18 a 21					

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
8 a 10	Mat	Química		IComp	
10 a 13		POO	APS	Mat 2	PDS
14 a 16					
16 a 18					
18 a 21					

Fig. 3. Mutación por movimiento

C.3. Mutación por permutación: Se toma una posición aleatoria del fenotipo. A partir de ésta, y dependiendo de si es de 2 o 3 horas, se genera una nueva posición aleatoria del mismo horario. Para continuar, se intercambian las posiciones de ambas, de manera de que donde se ubicaba la primera, ahora se ubique la segunda y viceversa. En la Fig. 4 mostramos un ejemplo ilustrativo.

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
8 a 10	Mat	Química			
10 a 13		POO	APS	Mat 2	PDS
14 a 16			IComp		
16 a 18					
18 a 21					

Horario	Lunes	Martes	Miércoles	Jueves	Viernes
8 a 10	Mat	IComp			
10 a 13		POO	APS	Mat 2	PDS
14 a 16			Química		
16 a 18					
18 a 21					

Fig. 4. Mutación por permutación

D. Función de fitness

La función de fitness que evalúa la calidad de la solución representada por cada individuo, debe tener en cuenta qué tanto cumple con las restricciones definidas. Para ello, el fitness está formado por 4 términos, uno por cada requisito a satisfacer. Se probaron dos tipos de funciones de fitness, una aditiva y otra multiplicativa. Primero se expondrán cada uno de los términos y luego la relación entre éstos.

D.1. Sobrepasso de Aulas: Cuenta la cantidad de veces que se supera la capacidad de aulas definida. Si se cuenta con 4 aulas, contará para cada bloque horario, la cantidad de clases que se dictan simultáneamente. Si se sobrepasa el límite de 4, se sumará 1 en este contador.

D.2. Repeticiones de Asignaturas: Cuenta las veces que se da más de una vez cierta asignatura en el mismo día. Si el individuo establece, por ejemplo, que la teoría y la práctica 1 se dan en el mismo día, sumará 1 en este contador.

D.3. Solapamiento de años adyacentes: Se cuentan la cantidad de oportunidades en las que clase de años adyacentes se dictan en el mismo módulo. Por ejemplo, este contador sumará 1 cada vez que el individuo represente que un curso de segundo año se solape con cualquier clase de primer o tercer año.

D.4. Superposición de clases: Se aumentará el contador de esta restricción cuando dos clases del mismo año se asignen al mismo bloque horario, ya que significará que un alumno debería estar al mismo tiempo en dos clases distintas.

Una vez definido los términos de la función de fitness, procedemos a enunciar su forma. Primero presentamos la función costo aditiva en la Ecuación 1:

$$fitness = \frac{a_1}{p+1} + \frac{a_2}{q+1} + \frac{a_3}{r+1} + \frac{a_4}{s+1} \quad (1)$$

donde los a_i representan el peso o importancia que se le da a cada término del fitness y se definen según la Tabla II. Por otra parte, p, q, r y s representan los contadores, datos que se enuncian en la Tabla III.

TABLA II
VALORES DE LOS PARÁMETROS DEL FITNESS SUMA

Parámetro	Valor
a_1	0.05
a_2	0.05
a_3	0.35
a_4	0.55

TABLA III
NOMENCLATURA DE LOS VALORES DEL FITNESS

Contador	Nombre
p	Repeticiones de asignaturas en un mismo día
q	Sobrepaso de aulas
r	Solapamiento de años adyacentes
s	Superposición de clases

A continuación, enunciamos la forma del fitness producto en la Ecuación 2

$$fitness = \frac{1}{(p+1)^{b_1}} \frac{1}{(q+1)^{b_2}} \frac{1}{(r+1)^{b_3}} \frac{1}{(s+1)^{b_4}} \quad (2)$$

donde p, q, r y s son como en 1 y los b_i representan la importancia de cada término y cuyos valores se muestran en la Tabla IV.

Como vemos, en ambos casos se le da mayor importancia al término de superposición, ya que éste es la mayor de las restricciones que debemos satisfacer, sin ella, la tabla de horarios obtenida sería de poca utilidad para cualquier alumno. El segundo término en importancia es el solapamiento entre asignaturas de años adyacentes, ya que es una de las condiciones más deseables de una tabla

TABLA IV
VALORES DE LOS PARÁMETROS DEL FITNESS PRODUCTO

Parámetro	Valor
b_1	1
b_2	1
b_3	2
b_4	5

horaria. El resto de las restricciones tienen valores bajos, ya que no son tan importantes o se cumplen por defecto al satisfacer las importantes.

III. RESULTADOS

IV. CONCLUSIONES

A. Trabajos futuros

B. Detalles técnicos

REFERENCIAS

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.
- [2] O. Roeva, *Real-World Applications of Genetic Algorithms*. InTech, 2012.
- [3] A. Colomi, M. Dorigo, y V. Maniezzo, "A genetic algorithm to solve the timetabling problem," *Computational Optimization and Applications*, 1993.
- [4] Y. Enzhe y S. Ki-Seok, "A genetic algorithm for a university weekly courses timetabling problem," *International Transactions in Operational Research*, no. 9, pp. 703–717, 2002.
- [5] D. Datta, D. K., y C. Fonseca, "Solving class timetabling problem of IIT kampur using multi-objective evolutionary algorithm," 2006.
- [6] F. Martinez Ruiz, E. García Sánchez, J. Muñoz Arteaga, y C. Castañeda Ramírez, "Timetabling académico usando algoritmos genéticos y programación celular," 2006.
- [7] C. Pérez de la Cruz y J. Ramírez Rodríguez, "Un algoritmo genético para un problema de horarios con restricciones especiales," *Revista de Matemática: Teoría y Aplicaciones*, no. 18, pp. 215–229, 2011.