

Herramientas heurísticas para la asignación óptima de horarios de clase

Heuristic tools for optimal assignment of class schedules

Jaillivi Marín Lozada^{1*}; Diana Lorena Hoyos B^{1*}; César A. Peñuela^{2*}; Jhon Jairo Santa Chávez^{2*}.

¹Estudiantes Facultad de Ingeniería Comercial. *jmarin.comercial@unilibrepereira.edu.co, *dlhoyos.comercial@unilibrepereira.edu.co.

²Docentes Universidad Libre Seccional Pereira. *capenuela@unilibrepereira.edu.co, *jjsanta@unilibrepereira.edu.co.

Fecha de recepción del artículo: 15/06/2013 Fecha de aceptación del artículo: 16/07/2013

Resumen

En este trabajo, se formula una metodología basada en técnicas heurísticas y en la técnica de Optimización por Colonia de Hormigas, para resolver el problema de asignación adecuada de salones. Para alcanzar este objetivo, se describe inicialmente la problemática y se desarrolla el modelo matemático para garantizar, que a cada evento programado, se le asignen los salones adecuados, así como los horarios de asistencia de los estudiantes matriculados. La eficiencia computacional del algoritmo propuesto, se pone a prueba usando instancias conocidas en la literatura técnica. Finalmente, se proponen trabajos futuros para mejorar los resultados logrados en este trabajo y así mismo, visualizar su viabilidad de implementación en las diferentes seccionales de la Universidad Libre.

Palabras clave

Asignación de horarios de clase, Metaheurísticas, Optimización por colonia de hormigas Técnicas de optimización.

Abstract

This paper proposes a methodology to solve the problem of proper allocation of classrooms. A

methodology is based on heuristics techniques as well as the use of the metaheuristic Ant Colony Optimization. To achieve this goal, it is initially described the problem. Thus, the mathematical model is developed in order to ensure that each scheduled event is assigned to an adequate classroom. Additionally, the attendance timing of the students enrolled to each event is determined. The computational effort of the proposed algorithm is tested using known instances in the literature. Finally, we propose future work to improve the results achieved in this work and visualize himself viability of the implementation of the methodology in the Libre University of Pereira.

Keywords

Ant colony system, Metaheuristics, Optimization techniques.

1. Introducción

1.1 Generalidades

En cada ciclo de formación de los estudiantes, una institución educativa debe determinar los espacios que deben ser usados para impartir asignaturas y/o realizar eventos. Para ello, la institución requiere de una infraestructura amplia, compuesta por las sa-

las de clases necesarias, para que tanto el cuerpo docente, como los estudiantes, se sientan en un ambiente propicio para el desarrollo de las actividades educativas. Desde el punto de vista operativo de la institución educativa, resulta de alto interés la búsqueda de una operación económica eficiente de su infraestructura, lo cual implica, que los salones existentes, sean usados en su totalidad durante cada periodo de tiempo considerado en el ciclo de aprendizaje. Sin embargo, para alcanzar este objetivo, también deben ser considerados un alto número de variables que deben ser satisfechas, como por ejemplo: el número de eventos a programar, el total de estudiantes adscritos a la institución, el número de estudiantes matriculados en cada evento, el total de salas disponibles y las características que permiten el desarrollo de ciertos tipos de eventos (laboratorios, salas de informática, capacidad, entre otros).

La asignación adecuada de los espacios (salones), determina igualmente los horarios en los cuales cada estudiante debe presentarse para asistir a las materias o eventos que se programaron al inicio del ciclo académico. En este punto, las preferencias de los estudiantes también conforman una serie de limitaciones, las que deben ser tenidas en cuenta, con el fin de brindar a los estudiantes un cronograma claro y simple, que permita la distribución adecuada de sus labores académicas. El no asistir a eventos programados en el último bloque de tiempo de cada día, así como el no recibir demasiados eventos consecutivos, son ejemplos clásicos de preferencias de los estudiantes para recibir sus horarios.

De esta manera, se define el problema de asignación de horarios de clase como un problema de optimización, en el cual se buscan determinar los espacios necesarios para generar la carga académica de todos los estudiantes; esto queda sujeto a un conjunto de restricciones que son producto de un limitado valor de recursos disponibles para cada bloque de tiempo [1]. Problemas de ese tipo son el centro de atención de la comunidad científica en los campos de la investigación operativa y la inteligencia artificial. Con el fin de integrar a los grupos investigadores y avanzar en el tema, se creó el

concurso “*International Timetabling Competition*” [2], donde se aborda el problema, en dos categorías: programación de exámenes y programación de salones. Para el caso de la programación de salones, las restricciones impuestas son de dos tipos: por un lado, se encuentran las restricciones de estricto cumplimiento, denominadas “duras”; y de otra parte, se definen las restricciones de cumplimiento opcional, denominadas “blandas”. Las restricciones blandas o suaves, son deseables; pero en soluciones reales, pueden llegar a ser difíciles de satisfacer completamente. Un análisis más detallado del tipo de restricciones que se pueden suponer en la programación puede ser consultada en [3].

Las restricciones duras se refieren a:

- Un estudiante no debe asistir a dos eventos diferentes a la misma hora (cruce de horarios para los estudiantes)
- Un salón de clase debe ser apto para el desarrollo normal de cualquier evento que se le asigne (idoneidad del salón)
- Un salón de clase no puede ser sede simultánea para dos eventos diferentes (cruce de salones)

Por su parte, las restricciones de tipo blandas, son las enumeradas a continuación:

- A un estudiante no se debe le deben programar eventos a la última hora del día.
- Los estudiantes no deben asistir a más de dos eventos en forma consecutiva.

Una gran variedad de métodos se han descrito en la literatura para resolver el problema de asignación de horarios, siendo en cada caso puestos a prueba en instancias teóricas y prácticas. Los métodos usados se pueden dividir en cuatro tipos: métodos secuenciales [4-6], métodos de cluster [7,8], métodos basados en restricciones [9], y los métodos metaheurísticos [10]. Durante las últimas dos décadas, se ha dado especial énfasis a la implementación de metaheurísticas, tales como: Recocido Simulado, Búsqueda Tabú, Genéticos, métodos Híbridos, entre otros. Por otro lado, un desarrollo basado en la

metaheurística Colonia de Hormigas puede resolver el problema de asignación óptima de horarios de clase. Esta metodología de solución, ha mostrado amplias cualidades en otros problemas típicos de la literatura especializada, caracterizados por su alta explosión combinatorial, tales como: Asignación de Tareas, Ubicación de Bancos de Condensadores, entre otros [11-13]

En este trabajo, se propone la técnica de Colonia de Hormigas para resolver el problema de programación óptima de horarios de clase; aplicando la metodología en casos de prueba de gran tamaño, como los propuestos en la página oficial de International Timetabling Competition [2].

1.2 Modelo matemático

Para el manejo de los datos del problema de programación óptima de horarios de clase, la información suministrada por la institución es almacenada en matrices y vectores, de forma que la información sea fácilmente accesada y manipulada. Para el presente trabajo, las matrices son definidas para almacenar:

- Información de las características requeridas por los eventos
- Información de las características que brinda cada salón
- Capacidad de los salones
- Información de los estudiantes que están matriculados en cada evento

La información de las características requeridas para cada evento, se almacena en la matriz *eventos_características* (**EC**). En dicha matriz, el elemento EC_{ij} es ajustado en el valor uno (“1”), en caso de que el evento i requiere que el salón que se le asigne cumpla con el requisito j . En caso contrario, dicho elemento es ajustado con el valor cero (“0”). Análogamente, la información de las características que tienen los salones, son almacenadas en la matriz *salones_características* (**SC**). En ese caso, el elemento SC_{ij} es ajustado en el valor uno (“1”), en caso de

que el salón i tenga la característica j . En caso contrario, dicho elemento es ajustado con el valor cero (“0”). Adicionalmente, para identificar la capacidad de cada salón, se define el vector *capacidad_salones* (**CapS**). En dicho vector, el elemento $CapS_i$ almacena el número de alumnos que pueden ingresar al salón i . A partir de las matrices **EC** y **SC**, y del vector **CapS**, se puede crear la *matriz eventos_salones* (**ES**), la cual se emplea para definir las aulas que pueden ser efectivamente asignadas a cada evento. En base a lo anterior, el elemento ES_{ij} debe tomar el valor uno (“1”) en caso que el evento i pueda ser programado en el salón j ; en caso contrario, tendrá valor cero (“0”). Por tanto, esa matriz, resume las condiciones características y de capacidad de los salones, frente a los eventos.

La información de los estudiantes que están matriculados en cada evento, se representa en la *matriz estudiantes_eventos* (**EstEv**). En ese caso, el elemento $EstEv_{ij}$ toma el valor uno (“1”) si el estudiante i asiste al evento j , o toma el valor cero (“0”), en caso contrario.

La programación de los eventos se puede representar por una *matriz evento_hora_salon* (**EHS**), la cual se forma con dos columnas, y por tantas filas, como número de eventos considerados en el problema. Así, para cada evento, la primera columna almacena la variable que determina la hora en la que se programa tal evento; mientras que la segunda columna, almacena el salón donde se realiza. Se puede formar la matriz *horario_eventos* (**HE**), que contiene un número de filas igual al número de eventos, y tantas columnas como bloques de tiempo considerados (45 para las instancias analizadas en los sistemas de prueba). La matriz **HE** es usada para almacenar el valor de la variable binaria, que determina la hora en que se programa un evento determinado. Por lo tanto, HE_{ij} tiene valor uno (“1”) en caso que el evento i sea programado en la hora j , o tiene valor cero (“0”), en caso contrario. El horario de los estudiantes se puede generar a partir de las matrices **EHS** y **EstEv**, con lo que se pueden conocer la hora y el salón programados para cada estudiante, según los eventos en que se haya matriculado.

Para facilitar la comprobación de las restricciones blandas, se forma la matriz *horario_estudiante* (**HEst**), la cual posee un número de filas igual al número de estudiantes matriculados, y tantas columnas como bloques de tiempo. En ese caso, la matriz **HEst** almacena en la posición (i,j) un uno ("1") en caso que el estudiante i tenga programado un evento en la hora j , o un valor cero ("0"), en caso contrario.

2. Metodología

2.1 Algoritmo de solución

El algoritmo por colonia de hormigas, es una metaheurística capaz de encontrar soluciones de buena calidad a problemas de optimización altamente complejos. Se basa en la habilidad que poseen las hormigas naturales, para encontrar el alimento a partir de individuos relativamente simples, pero con una estructura social altamente eficiente. La base en ambos sistemas (natural y artificial), es la comunicación indirecta entre todos los individuos a partir de rastros de feromonas. El algoritmo, construye una alternativa de solución agregando paso a paso un elemento i de entre un conjunto de alternativas N_i^k , conocido como vecindad del elemento k . Cada elemento tiene consigo una probabilidad p_i de ser elegido, dada por:

$$p_i = \frac{\tau_i^\alpha \cdot \eta_i^\beta}{\sum_{l \in N_i^k} \tau_l^\alpha \cdot \eta_l^\beta} \quad (1)$$

Donde τ_i es una cantidad numérica almacenada en una matriz de feromonas y define el grado de participación del elemento i , en alternativas de solución de buena calidad, encontradas con anterioridad. El factor η_i es un valor dado por un análisis de sensibilidad del sistema, y define el impacto que produce el elemento i , al ser adicionado a la alternativa que se construye. Los parámetros de inicialización α y

β , definen el grado de importancia de la información utilizada.

El proceso para construir una alternativa de solución, se repite hasta completar varios individuos. Cada uno de ellos, produce un incremento o un decremento de feromona sobre los elementos que los forman, en forma proporcional a la calidad de la solución. Con base a esa información adicional, se genera una nueva población con mejores características.

Por otro lado, para evitar el estancamiento del algoritmo en alternativas de baja calidad, el sistema Colonia de Hormigas, efectúa un proceso de evaporación en dos etapas distintas. La primera, conocida como evaporación local, se realiza durante la construcción de cada solución. La segunda, conocida como evaporación global, se ejecuta luego de evaluar la función objetivo de todas las alternativas de la población.

2.2 Codificación del problema

Para la formación de las alternativas de solución, se recurre a dos tipos matrices de feromonas. La primera, relaciona la deseabilidad aprendida de seleccionar el salón i , para dictar el evento j ; por tanto es de dimensiones $S \times E$ (donde S es el número total de salones y E la cantidad de eventos del problema). La segunda matriz, relaciona la deseabilidad aprendida de seleccionar el bloque de tiempo i , para dictar el evento j , por tanto es su dimensiones $BT \times E$ (donde BT , es el número total de bloques de tiempo).

El orden en el que se otorga la asignación, depende del número de salones donde es factible que un evento se desarrolle; dando prioridad a los eventos con un menor número de opciones factibles. La probabilidad de escoger el salón i , para el evento j , se calcula por medio de la ecuación (1), usando como información heurística la ecuación (2).

$$\eta_{ij} = f * S_a^{NES} \quad (2)$$

En la ecuación (2), S_a es un parámetro continuo en el intervalo $[0.9, 1.0]$, con el cual se decrementa la probabilidad de escoger el salón i , por la asignación de otros eventos con anterioridad; mientras que la variable NES_i indica el número de eventos asignados al salón i , durante la alternativa de solución en construcción. La variable f_j toma el valor de uno (1.0) si el salón i es apto para el evento j , y cero (0), en caso contrario.

La probabilidad de que un evento j , se dicte en el bloque de tiempo i , se calcula por medio de la ecuación (1); pero en este caso, la información heurística se determina por la ecuación (3).

$$\eta_{ij} = (1 - Asig) * Cr^{NC} * Ec^{NCont} \quad (3)$$

Donde $Asig$ es un vector que caracteriza a todos los bloques de tiempo posibles. En cada posición, toma el valor de uno ("1") si el salón ha sido programado con anterioridad; o el valor de cero ("0"), en caso contrario. Cr y Ec son factores que reducen la probabilidad de aceptar el bloque de tiempo i , en caso de que se produzca un cruce de horario, en el primer caso, o un evento continuo en el segundo caso. NC , es el número de estudiantes a los cuales se les asignó previamente asistir a un evento distinto al que se está programando durante el bloque de tiempo i . $NCont$, es el número de eventos continuos que se producen en los horarios de los estudiantes matriculados al evento, j si el bloque de tiempo i es aceptado.

Cada vez que a un evento se le asigna un salón o un bloque de tiempo durante la construcción de la alternativa de solución, se disminuye levemente la cantidad de feromona que caracteriza al elemento adicionado. La evaporación global se aplica al final de cada iteración y afecta todos los elementos de todas las alternativas, excepto a la que posea las mejores características. Para efectos del problema, se define que una alternativa supera otra, si se cumple que:

- Posee un menor grado de infactibilidad, o
- Posee el mismo valor de infactibilidad, pero un menor número de restricciones blandas insatisfechas.

El factor de evaporación global (ρ), se calcula por medio de la siguiente ecuación:

$$\rho = t \cdot \left(1 - 0.99^{(a+b)}\right) \quad (4)$$

Donde la variable a , determina la diferencia entre el número de restricciones duras violadas por la alternativa actual y las restricciones violadas por la mejor alternativa encontrada durante todo el algoritmo. En forma análoga es calculado b , pero teniendo en cuenta las restricciones blandas. En base a lo anterior, los depósitos de feromona se calculan de acuerdo a la ecuación (5).

$$\Delta = t \cdot 0.99^{(R_{duras} + R_{blandas})} \quad (5)$$

Donde Δ , es el incremento en cada elemento de las matrices de feromonas que pertenecen a las alternativas de solución habilitadas para los depósitos. R_{duras} , es el número de restricciones duras que origina la alternativa de solución que es evaluada; mientras que, de forma análoga, $R_{blandas}$, representa el número de restricciones blandas. El parámetro de calibración t , equivale a un 5% del valor de la feromona inicial en caso de evaluar la mejor alternativa de la iteración, y de un 10% de la feromona inicial en caso de evaluar la incumbente.

3. Resultados y análisis

Se desarrolló un programa computacional usando lenguaje Python, con el fin de implementar la metodología propuesta para la solución del problema de programación óptima de horarios de clase usando, Colina de Hormigas. El programa fue probado con tres instancias del problema propuestas en la *International Timetabling Competition* [2]. En cada ins-

tancia, se deben programar 400 eventos en 10 salones de clase; en una institución que cuenta con 200 estudiantes. Por su parte, los salones de clase son definidos a partir de diez características diferentes, que pueden o no, dar cabida a la selección de un determinado evento. Para cada instancia, se efectuaron 300 iteraciones, usando 10 hormigas; consiguiendo en todo caso, soluciones factibles (Cumplimiento de restricciones duras).

En la instancia 1, el valor de la función objetivo es de 386; que corresponde a la siguiente secuencia de restricciones blandas insatisfechas, en los horarios de los estudiantes: 209 eventos continuos, 147 eventos en la hora nueve del día y 30 eventos de asistencia única por día.

En la instancia 2, el valor de la función objetivo es de 365, la que corresponde a la siguiente secuencia de restricciones blandas insatisfechas en los horarios de los estudiantes: 188 eventos continuos, 147 eventos en la hora nueve del día, 30 eventos de asistencia única por día.

En la instancia 3, el valor de la función objetivo es de 434, que corresponde a la siguiente secuencia de restricciones blandas insatisfechas en los horarios de los estudiantes: 207 eventos continuos, 206 eventos en la hora 9 del día, 21 eventos de asistencia única por día.

Tabla 1. Resumen de resultados

Número de estudiantes con:	Caso de Prueba		
	1	2	3
Eventos continuos	209	188	207
Eventos finales	147	121	206
Eventos únicos	30	30	21

4. Conclusiones

Se ha mostrado que la metaheurística de optimización por Colonia de Hormigas, resuelve el problema de asignación óptima de salones, encontrando una solución factible y de buena calidad; considerando el tamaño y complejidad de las instancias asumidas.

Durante el desarrollo de la aplicación y puesta a prueba del programa computacional, se observó que resulta conveniente asignar inicialmente los salones a los eventos y posteriormente los bloques de tiempo. Adicionalmente, es crítica la asignación en primera instancia de los eventos que cuentan con un reducido conjunto de salones aptos, con lo cual se reduce drásticamente la infactibilidad.

Estudiando el comportamiento del aplicativo con los casos de prueba, se observó que una vez alcanzada la solución factible, es difícil que esta sea mejorada, lo cual denota una convergencia acelerada del algoritmo. Lo anterior, se debe a la definición adoptada para comparar alternativas y a que siempre se almacena como incumbente, la que produzca la menor infactibilidad. De esta manera, se renuncia al interés sobre cualquier solución con bajas restricciones blandas, que puede resultar tan importante como una solución factible. Se pueden almacenar varias incumbentes, siguiendo el criterio de que cada una de ellas, minimice una de las restricciones blandas; con lo que se obtiene mayor diversidad de las configuraciones.

Como trabajos futuros, se propone implementar una técnica híbrida entre Colonia de Hormigas y SimulatedAnnealing de forma que con este último, se obtengan soluciones iniciales diferentes y de buena calidad, para que con Colonia de Hormigas se realice la exploración del vecindario. Además, se deben considerar aspectos adicionales como: el desplazamiento de los estudiantes de un salón a otro, el que se tengan eventos que duren más de una hora y materias que deban distribuirse en diferentes días de la semana. También se puede considerar el problema de asignación de carga docente, resolviéndolo simultáneamente, con el de programación de clases.

La propuesta final de este proyecto, se encamina a su aplicación en las diferentes seccionales de la Universidad Libre. Para esto, se debe generar un modelo matemático suficientemente amplio, que albergar la complejidad numérica que representa el alto número de estudiantes matriculados en la Uni-

versidad, durante cada período académico. De esta forma, se pretende minimizar el esfuerzo humano que actualmente se requiere para asignar los horarios de clase al inicio de cada semestre; y, por otro lado, incrementar la eficiencia de la infraestructura física de las seccionales, evitando, o por lo menos aplazando, inversiones en adecuación o construcción de nuevas salas.

Referencias

1. Wren A. Scheduling, timetabling and rostering – A special relationship In: Burke and Ross, pp. 46–75, 1996.
2. International Timetabling Competition. Disponible en página de internet: <http://www.idsia.ch/Files/ttcomp2002/>
3. Burke, E.K.; Elliman, D.G.; Ford, P.; Weare, R.F.; Examination timetabling in British Universities – A survey. In: Burke and Ross, pp. 76–92, 1996.
4. Carter, M.W.; Laporte, G.; Recent developments in practical examination timetabling. In: Burke and Ross, pp. 3–21, 1996.
5. Carter, M.W.; Laporte, G.; Recent developments in practical course timetabling. In: Burke and Carter, pp. 3–19, 1996.
6. Werra; An introduction to timetabling. *European Journal of Operational Research*, 19, pp. 151–162, 1985.
7. White, G.M.; and Chan P.W.; Towards the construction of optimal examination timetables. *INFOR* 17, pp. 219–229, 1979.
8. Fisher, J.G.; Shier, D.R.; A heuristic procedure for large-scale examination scheduling problems. Technical Report 417, Department of Mathematical Sciences, Clemson University.
9. White G.M.; Constrained satisfaction, not so constrained satisfaction and the timetabling problem. In: A Plenary Talk in the Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling, University of Applied Sciences, Konstanz, August 16–18, 2000, pp. 32–47.
10. Burke, E.K.; Newall, J.P.; Weare, R.F.; 1996b. A memetic algorithm for University exam timetabling. In: Burke and Ross, pp. 241–250, 1996.
11. Domingo M.; Stützle T.; *Ant Colony Optimization*, A Bradford Book, Massachusetts Institute of Technology, 2004.
12. Granada M.; Toro E.M.; Franco J.F.; Programación óptima de horarios de clase usando un algoritmo memético. *Scientia et technical* Año XII no. 30 Mayo de 2006.
13. Burke; McCollum; Meisels; A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research* 176, 2007.