

El problema de horarios y cursos basado en currículum

**Ing. Jorge Carlos González González
M.C. Alma Edith Martínez Licona
Universidad Autónoma Metropolitana
Unidad Iztapalapa**

Los problemas de horarios surgen en diversos campos donde se tenga la necesidad de administrar diversos recursos (infraestructura, personas, tiempo, etc.) e.g. hospitales (nurse rostering problem) [1], compañías (personnel scheduling) [2], instituciones educativas, por mencionar algunos.

Particularmente para las instituciones universitarias es un problema de interés, debido a que cada nuevo periodo escolar, deben de generar o reajustar sus horarios para ofrecer una serie de asignaturas y eventos en los salones que poseen durante el periodo escolar. Es en este tipo de escenario donde entra el problema de horarios y cursos basados en currículum [3].

El problema de horarios y cursos basado en currículum (por sus siglas en inglés, CBCT) abre una oportunidad como un acercamiento a las dificultades que tienen las instituciones educativas para resolver los problemas relacionados con la generación de calendarios. El esquema CBCT fue propuesto por la Competencia Internacional de Horarios (por sus siglas en inglés, ITC) en el año 2007, y es hasta la fecha un problema que continúa siendo estudiado [4].

El planteamiento propuesto por CBCT no es el primer esquema para abordar los problemas de calendarización para instituciones educativas [5] [6], pero se diferencia de otros modelos debido a que su enfoque se encuentra hacia el lado de la institución, generando el horario con base en los cursos que deben ser ofertados en el ciclo escolar, a diferencia de otros acercamientos como lo podría ser el problema de UCTP (por sus siglas en inglés: el problema de horarios y cursos universitarios), el cual propone la elaboración de horarios con relación a los alumnos y sus cursos [7].

La idea esencial de CBCT, es el alojar un conjunto de lecturas de cursos (la lectura en el contexto del problema es el equivalente a una clase) en diferentes periodos de tiempo, mientras se respetan una serie de restricciones de carácter duras y blandas que deben ser cumplidas al momento de proponer un horario [8] [9].

Variables

El problema de horarios y cursos basado en currículum, maneja diversas variables que representan recursos que una institución educativa debe administrar: profesores, cursos, salones, tiempo, etc., por lo que, para una clara ilustración del problema, se hará una descripción de cada una [10].

La forma de dividir el tiempo puede ser manejado con las siguientes variables: día y hora. Los días representan la semana de trabajo de la institución educativa (típicamente 5 a 6 días), mientras que las horas son los periodos de actividad que tiene la institución para realizar las labores [4] [9].

Los cursos son las materias que oferta la institución educativa en cada nuevo periodo escolar. Cada curso consta de un número fijo de lecturas que deben ser programadas en distintos periodos de tiempo, además cada lectura es atendida por un número definido de estudiantes y es impartida por un profesor. Como última característica, cada curso tiene un número mínimo de días en los que debe de ser extendidas las clases en el calendario [4] [8].

Los salones son las estancias donde se ofrecen las clases, de acuerdo al planteamiento del problema, cualquier cuarto puede albergar a cualquier lectura. Cada habitación cuenta con diferente número de sillas, lo que constituye la propiedad

principal de las aulas [8] [9].

Los profesores son las personas que ofrecen los cursos que propone la institución educativa. Cada maestro puede tener uno o más cursos asociados. Como última propiedad, existen profesores que no estarán disponibles a determinadas horas dentro del calendario [7] [10].

La última variable del problema es la currícula, la cual se define como un grupo de cursos que poseen la propiedad de que, al tomar cualquier par del conjunto, tienen un grupo de estudiantes que cursan ambas materias [4] [7].

Restricciones

El problema CBCT posee una serie de limitaciones que se deben considerar al momento de proponer un horario. Las restricciones se pueden dividir en dos categorías: restricciones duras y restricciones blandas (también llamadas restricciones suaves) [9].

Una restricción dura se define como un requisito que debe ser cumplido de manera obligatoria. Las restricciones duras son las que hacen que una solución sea válida si cumple con todas las limitantes, y en caso contrario, no es viable y carece de valor para el problema. Particularmente para CBCT, se tiene las siguientes restricciones duras [4]:

- 1) Todas las lecturas pertenecientes a los cursos deben de ser asignadas en el calendario.
- 2) Dos lecturas no pueden ocurrir en el mismo periodo de tiempo y salón.
- 3) Lecturas pertenecientes a una currícula o impartidas por el mismo profesor,

deben de ser asignadas en periodos diferentes.

- 4) Si un profesor no está disponible para impartir una lectura de un curso en un periodo de tiempo específico, entonces, no es posible asignar la lectura a esa hora.

Una restricción blanda es un requisito de carácter opcional, por lo que no es necesario cumplir con todas al momento de proponer una solución. Sin embargo, cada restricción blanda no satisfecha genera un impacto sobre el valor del calendario al ser evaluado por la función objetivo. Por lo que cumplir con un mayor número de restricciones blandas ofrece al calendario un mejor valor y por tanto un candidato superior a ser empleado [4]. Para el problema CBCT se tienen las siguientes restricciones suaves:

- 1) Para cada lectura; el salón asignado debe contar con número de asientos igual o mayor a los estudiantes.
- 2) Las lecturas de un curso deben ser extendidas en un mínimo de días.
- 3) Las lecturas pertenecientes a la misma currícula, deben de estar adyacentes una de otra.
- 4) Todos los cursos deben de estar en un sólo salón.

Evaluación

Dado que es posible obtener más de un calendario que intente resolver el problema CBCT, se ve la necesidad de poder calificar la utilidad que tiene cada horario propuesto. Por lo anterior descrito la ITC2007 proporciona una serie de pautas que dictan cómo se debe realizar la valoración de una solución [4].

Dependiendo de la representación de la solución, es posible que la forma de realizar la evaluación cambie, sin embargo, en esencia lo que se realiza, es la ponderación de restricciones duras y restricciones blandas que no son satisfechas en el horario propuesto (Fórmula 1) [11].

$$CostoTotal = \left[\sum_{i=1}^{RD} P_D \cdot CostoD_i \right] + \left[\sum_{j=1}^{RB} CostoB_j \right]$$

Ecuación 1. Evaluación de una solución.

Donde:

RD: Son todas las restricciones duras a revisar en el calendario.

RB: Son todas las restricciones blandas a revisar en el calendario.

P_D: Es la penalización por romper una restricción dura.

CostoD_i: Es el costo asociado por romper una restricción dura.

CostoB_j: Es el costo asociado por romper una restricción dura.

Propuestas para abordar el problema de horarios y cursos basados en currículum

El primer intento para abordar el problema CBCT, fueron los algoritmos propuestos en la primera competencia del año 2007. El ganador fue Tomás Müller, quien utilizó una estrategia híbrida donde se implementaban diferentes algoritmos que trabajaban en conjunto de los que destacan: Gran Diluvio, Recocido Simulado y Ascenso de la Colina [4].

Sin embargo, desde la publicación de los resultados iniciales de la competencia han surgido diferentes propuestas para abordar el problema CBCT, con resultados variables, en la siguiente sección se hará

mención de algunos de los algoritmos más recientes para hacer frente a la tarea.

Recocido simulado

La metaheurística llamada recocido simulado fue propuesta por Kirkpatrick [12], está inspirada en un método empleado en la metalurgia denominado recocido. La técnica del recocido de metales consiste en elevar la temperatura del elemento y después de manera lenta, bajar la temperatura propiciando un arreglo de moléculas equilibrado y generando un estado de energía mínima [13].

De manera general, la metaheurística emula el proceso del recocido, al comienzo del algoritmo se inicia con una solución a una temperatura muy alta, lo cual, produce candidatos en los que se generan grandes cambios en su “estructura”. Conforme la temperatura desciende, los cambios son cada vez más finos hasta llegar a la temperatura de enfriado, simulando la configuración de baja energía [14] [15].

Gran diluvio

El algoritmo de gran diluvio es una estrategia de búsqueda propuesta por el investigador Duek, el cual funciona utilizando tres componentes principales: nivel de agua, ratio de decaimiento y solución esperada [16].

La técnica funciona de manera similar a recocido simulado, aceptando soluciones mejores a la actual y con la posibilidad de permitir soluciones peores. A diferencia de recocido simulado que se inspira de la técnica metalúrgica del recocido de metales, gran diluvio hace una analogía del ascenso y descenso del nivel de agua durante una lluvia prolongada [13][17].

Algoritmos genéticos

Los algoritmos genéticos fueron propuestos por John Holland los años 70s [18]. Reciben el nombre de algoritmo genético debido a la forma en que trabajan. En su diseño tienen mecanismos internos denominados operadores genéticos [18], los cuales intentan emular el proceso de selección natural, la modificación genética, la evolución de las especies entre otros, con el fin de mejorar la población con el pasar de las generaciones [19].

A diferencia de otras heurísticas como la escalada de la colina donde una solución inicial es mejorada en cada iteración [14] [15], los algoritmos genéticos producen un conjunto de posibles candidatos en cada iteración que realizan [20].

La capacidad de generar soluciones distintas desde el comienzo de la ejecución, permite al algoritmo el no estancarse de manera prematura en un óptimo local, además de generar una exploración más amplia del campo solución, debido a que idealmente cada solución se encuentra en regiones diferentes [20].

Hiper-heurísticas

Las hiper-heurísticas buscan ser una técnica de mayor grado de generalización que las metaheurísticas. En los algoritmos hiper-heurísticos se plantea obtener resultados consistentes en diferentes instancias de diversos problemas, en lugar de obtener resultados excelentes en pocas instancias de un solo problema de dominio [1] [21].

En el esquema básico de las hiper-heurísticas existen dos componentes principales: el nivel alto y el nivel bajo. En el nivel inferior se encuentra las heurísticas de bajo nivel, las cuales trabajan directamente con el problema. Mientras que en el nivel

superior se encuentra una estrategia que guía el proceso de búsqueda. Por último, existe el componente denominado barrera de dominio, el cual tiene el objetivo de separar ambos niveles. así como ser el medio de intercambio de información entre los mismos [22][23].

Bibliografía

[1] N. Pillay, R. Qu, “Hyper-Heuristics: Theory and Applications”, Springer, Natural Computing Series, 2018.

[2] K. Smith-Miles, L. Lopes, “Measuring instance difficulty for combinatorial optimization problems”, *Computers & Operations Research*, vol. 39, pp. 875–889, 2012.

[3] C. K. Teoh, A. Wibowo, M. S. Ngadiman, “Review of state of the art for metaheuristic techniques in Academic Scheduling Problems”, *Artificial Intelligence Review*, vol. 44, pp. 1–21, 2015.

[4] Competencia Internacional de Horarios, problema de horarios 2007 y cursos basados en currículum, <http://www.cs.qub.ac.uk/itc2007/curriculumcourse/course...>

[5] V. Králev, R. Králeva, “A local search algorithm based on chromatic classes for university course timetabling problem”, *International Journal of Advanced Computer Research*, Vol 7, 2016.

[6] M. Alzaqebah, S. Abdullah, “Hybrid bee colony optimization for examination timetabling problems”, *Computers & Operations Research*, vol. 54, pp. 142–154, 2015.

[7] N. Pillay, E. Özcan, “Automated generation of constructive ordering heuristics for educational timetabling”, *Annals of*

- Operations Research, vol. 275, pp. 181–208, 2019.
- [8] N. C. F. Bagger, M. Sørensen, T. R. Stidsen, “Benders’ decomposition for curriculum-based course timetabling”, *Computers and Operations Research*, vol. 91, pp. 178–189, 2018.
- [9] M. Lindahl, A. J. Mason, T. Stidsen, M. Sørensen, “A strategic view of University timetabling”, *European Journal of Operational Research*, vol. 266, pp. 35–45, 2018.
- [10] T. Chong Keat, H. Haron, A. Wibowo, M. S. Ngadiman, “A Heuristic Room Matching Algorithm in Generating Enhanced Initial Seed for the University Course Timetabling Problem”, *Research Journal of Applied Sciences, Engineering and Technology*, vol. 10, pp. 882–889, 2015.
- [11] H. Babaei, J. Karimpour, A. Hadidi, “A survey of approaches for university course timetabling problem”, *Computers & Industrial Engineering*, Vol. 86, pp 43–59, 2015.
- [12] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, “Optimization by simulated annealing”, *Science*, vol. 220, issue 4598, pp. 671–680, 1983.
- [13] I. Boussaïd, J. Lepagnot, P. Siarry, “A survey on optimization metaheuristics”, *Information Sciences*, vol. 237, pp. 82–117, 2013.
- [14] T. Song, S. Liu, X. Tang, X. Peng, M. Chem, “An iterated local search algorithm for the University Course Timetabling Problem”, *Applied Soft Computing*, Volume 68, pp. 597–608, 2018.
- [15] M. Kalender, A. Kheiri, E. Özcan, E. K. Burke, “A greedy gradient-simulated annealing selection hyper-heuristic”, *Soft Computing*, vol. 17, pp. 2279–2292, 2013.
- [16] G. Dueck, “New optimization heuristics: the great deluge algorithm and the record-to-record travel”, *Journal of Computational Physics*. Vol. 104, pp. 86–92, 1993.
- [17] C. Weng Fong, H. Asmuni, B. McCollum, P. McMullan, S. Omatu, “A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems”, *Information Sciences* vol. 283, pp. 1–21, 2014.
- [18] L. Araujo, C. Cervigón, “Algoritmo evolutivos: un enfoque práctico”, RA-MA, México, 2009.
- [19] J. B. Matias, A. C. Fajardo, R. M. Medina, “Examining Genetic Algorithm with Guided Search and Self-Adaptive Neighborhood Strategies for Curriculum-Based Course Timetable Problem”, *IEEE, Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, pp. 1–6, 2019.
- [20] S. Sivanandam, & S. Deepa, “Introduction to Genetic Algorithms”, Springer., 2008.
- [21] N. R. Sabar, M. Ayob, G. Kendall, R. Qu, “A Dynamic Multiarmed Bandit-Gene Expression Programming Hyper-Heuristic for Combinatorial Optimization Problems”, *European Journal of Operational Research*, *IEEE Transactions on Cybernetics*, Volumen 45, Número 2, pp. 217–228, 2015.
- [22] N. R. Sabar, M. Ayob, G. Kendall, R. Qu, “Automatic Design of a Hyper-

Heuristic Framework With Gene Expression Programming for Combinatorial Optimization Problems”, *IEEE Transactions on Evolutionary Computation*, Volumen 19, Número. 3, pp. 309-325, 2015.

[23] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, R. Qu, “Hyper heuristics: a survey of the state of the art”, *Journal of the Operational Research Society*, vol. 64, pp. 1695–1724, 2013.