UN MODELO DE PROGRAMACION ENTERA BASADO EN PATRONES PARA LA ASIGNACION DE SALAS DE CLASES PARA UNA FACULTAD DE MEDICINA

Jaime Miranda
Facultad de Economía y Negocios – Universidad de Chile
Diagonal Paraguay 256 – Santiago – Chile
jmirandap@fen.uchile.cl

Pablo Andrés Rey
Facultad de Ingeniería – Universidad Diego Portales
Av. Ejército 441 – Santiago – Chile
pablo.rey@udp.cl

RESUMEN

Este trabajo presenta un modelo de programación entera para la asignación de salas de clases de una Facultad de Medicina. En particular, el problema de programación enfrentado se diferencia de los problemas usuales de asignación de salas de clases ya que los requerimientos de salas varían de acuerdo a la sesión de clases y éstos no se repiten regularmente. De esta manera, la asignación debe contemplar distintos tipos de salas y abarcar el semestre de clases completo. La definición de las variables del modelo considera la asignación simultánea de un grupo o patrón de salas de clase. La aplicación a un caso real muestra que el modelo permite encontrar asignaciones que cumplen todas las condiciones impuestas dentro de un tiempo computacional reducido.

PALAVARAS CHAVE. Planificación educativa. Asignación de recursos. Programación entera.

Área principal: EDU – PO na Educação

ABSTRACT

This paper presents an integer programming model for allocating classrooms for a medical school. The planning problem faced is different from the standard problems classroom assignment since room requirements could vary on different session and will not be repeated regularly. Consequently, the assignment must provide for different rooms and comprise the entire semester of classes. The definition of the variables of the model considers the simultaneous assignment of a group or pattern of rooms. The application to a real case shows that the model allows to find assignments that meet all the conditions imposed in a reduced computational time.

KEYWORDS. Educational planning. Resource assignment. Integer programming.

Main area: EDU – OR in Education

1. Introducción

Desde hace décadas, las instituciones de educación superior han utilizado sistemas de apoyo a la decisión para un espectro amplio de tareas administrativas, como por ejemplo inscripción de cursos, gestión de registros de alumnos, contabilidad y finanzas, entre otras. Otra actividad recurrente dentro de cualquier institución de educación corresponde la programación de las actividades que se realizan durante un semestre particular y en qué lugar se llevarán a cabo.

En su forma más simple, la programación horaria de cursos asigna a cada curso a un bloque horario y sala de clase considerando una serie de requerimientos y restricciones que permiten el normal funcionamiento de estas instituciones de educación superior (Burke,2002). Dada la naturaleza combinatorial del problema es de difícil resolución, por lo que los enfoques manuales basados en la prueba y error son ineficientes en escenarios caracterizados por una enorme cantidad de cursos y salas de clase. Este último factor realza la necesidad de contar con enfoques de solución que permitan determinar buenas soluciones. No obstante, el problema de la programación horaria de cursos no es un problema nuevo existiendo varios trabajos al respecto en la literatura de investigación de operaciones (Schaerf 1999; Burke, 2002).

En algunos casos, los problemas de programación de horarios y asignación de salas de clases se manejan de manera simultánea (Stallaert, 1997; MirHassani, 2006). Sin embargo, en varias situaciones, los horarios de los cursos son construidos con antecedencia, y en una segunda etapa se busca una asignación de salas de clase que se adecue al horario predefinido. Este es el caso cuando el horario de clases se replica de un periodo al siguiente pero varía el número de alumnos que toman los cursos o cuando, como en nuestro caso, el proceso de planificación está descentralizado y los agentes responsables de la programación de horarios y asignación de salas de clases son diferentes y no se retroalimentan.

En este artículo se presenta un modelo de programación entera para resolver el problema de asignación de salas de clase a cursos con horarios predefinidos. Cada sesión de un curso tiene diferentes requerimientos de sala de clase. Un requerimiento corresponde a una cantidad de salas de clase con su respectivo tipo. Una característica del modelo es el tipo de variable de decisión que utiliza, la cual asigna simultáneamente para todas las sesiones de un curso que solicitan un mismo requerimiento un único conjunto de salas de clase.

El resto del artículo se estructura de la siguiente manera. En la Sección 2, se realiza una breve revisión bibliográfica sobre problemas similares al abordado aquí. La Sección 3 presenta en detalle el problema de decisión analizado y el proceso de toma de decisiones en que se enmarca. En la Sección 4 se formula el modelo de programación entera propuesto. La Sección 5 presenta el caso de prueba y los resultados de los experimentos computacionales. Finalmente, las Secciones 6 y 7 presentan, respectivamente, una breve discusión sobre el uso adicional del modelo para la detección de conflictos en los requerimientos y las conclusiones del trabajo.

2. Revisión bibliográfica

Los problemas de programación de horarios y asignación de salas de clases no son nuevos y han sido ampliamente estudiados en la literatura (Schaerf, 1999; Lewis, 2008). Uno de los enfoques de solución más comúnmente utilizados para modelar el problema se basan en modelos de programación lineal entera (Stallaert,1997; Daskalaki y Birbas, 2005; Miranda, 2010). Dentro de este contexto, el trabajo presentado por (De Werra,1985) es uno de los seminales de esta área, el cual muestra una serie de modelos de optimización lineal que han

servido de inspiración para una serie de aplicaciones reales. Un conjunto de enfoques de solución que se basan en modelos de programación lineal entera se presentan en los siguientes trabajos (Daskalaki et al., 2004; Daskalaki y Birbas, 2005; Dimopoulou y Miliotis, 2001; MirHassani, 2006), los cuales muestran los beneficios de la aplicación de este tipo de modelo en diversas universidades.

Específicamente, el problema de asignación de salas de clases fue resuelto por primera vez, de acuerdo con los autores, en (Glassey y Mizrach, 1986). En ese trabajo se presenta un sistema de apoyo a las decisiones basado en un modelo de programación entera. En el caso estudiado se programan las clases de una semana que luego se replican en todo el periodo. El modelo asigna individualmente las clases y contempla que todas las sesiones ocupen la misma sala de clase. Debido al gran tamaño del problema para el caso real con la tecnología de la época, los autores presentan también un procedimiento heurístico con buenos resultados. La heurística propuesta es un procedimiento goloso que va seleccionando y asignando progresivamente los cursos aún no asignados, en orden decreciente del número de alumnos.

Carter y Tovey (1992) estudian la complejidad computacional de diferentes variantes del problema de asignación de salas de clases. Los problemas estudiados en este trabajo tampoco consideran la repetición de la sala asignada para todas las clases a programar. Varios problemas de los estudiados están la categoría de complejidad NP-Hard entre el que se encuentra el llamado *interval classroom assigment problem*. El problema que estudiamos en este trabajo es una generalización de ese problema, por lo tanto también es un problema NP-Hard.

Más recientemente, Kingston (2012) analiza un problema similar en el contexto de escuelas de nivel medio en Australia. El autor propone varios algoritmos combinatoriales basados en asignaciones parciales e intercambios. Kolen et al. (1997) es *survey* reciente sobre la clase más general de problemas de asignación por intervalos, que incluye algunas aplicaciones a problemas de asignación de salas.

3. El problema estudiado

El problema abordado en este trabajo consiste en la asignación de salas para las clases de una Facultad de Medicina. Para esta asignación se consideran predefinidos y conocidos los horarios de clases y los requerimientos de salas de clase (tipo y cantidad) para cada sesión de los cursos. La capacidad de las salas y las cantidades de alumnos inscritos en cada curso son contemplados en la etapa previa de definición de los requerimientos, por lo que no son tratados de manera explícita en el problema considerado.

De acuerdo a la actividad, los cursos requieren diferentes tipos y cantidades de salas. Por ejemplo, para las clases magistrales pueden ser dictadas para todos los alumnos de un curso en conjunto en un auditorio de gran capacidad, mientras que para las clases de laboratorio o evaluaciones, los alumnos son separados en pequeños grupos en varias salas. Además, las actividades y, por lo tanto, los requerimientos de salas de clases no se repiten periódicamente todas las semanas. Esto último conlleva la necesidad de, por un lado encontrar una asignación de salas de clase para todas las clases de un periodo completo y no para una semana base que se replica y, por el otro lado, que considere distintos combinaciones de salas de clase. Estas características diferencian el problema modelado de la mayoría de los trabajos publicados en el área.

Dentro de las condiciones que desea respete una asignación, encontramos la condición que todos las sesiones en que se realiza la misma actividad a lo largo del semestre se asignen al mismo conjunto de salas de clase. Esta condición no se respeta en muchas oportunidades debido a que las asignaciones se van haciendo dinámicamente y a veces, las salas de clase utilizadas la

oportunidad anterior ya han sido programados para otro curso. En caso de no poder asignarse salas del tipo requerido, existen reglas que indican qué salas pueden ser utilizadas para sustituir las requeridas. En el caso práctico resuelto, no fue necesario recurrir a esta posibilidad.

Si bien, el problema planteado consiste principalmente en encontrar una asignación factible de salas, en la práctica hay ciertas asignaciones que son preferibles a otras. Este hecho tiene relación con diversos factores como preferencias por el uso de determinadas salas para ciertas actividades o por la utilización en ciertos horarios que se verán reflejadas en la función objetivo del modelo.

Descripción del proceso de programación de salas de clase

El proceso actual para la asignación de salas de clase se detalla en la Figura 1. Este proceso comienza cuando un Coordinador por Nivel (CN) de cada carrera revisa los horarios de los semestres anteriores para ver la factibilidad repetirlos en el semestre actual. Después de revisar los horarios, el CN le propone a los profesores de su nivel los potenciales horarios que podrían tener sus cursos en un semestre particular. Si los profesores están de acuerdo con la propuesta de horarios se envía la solicitud de salas de clase a la Unidad de Gestión de Aulas (UGA). En caso contrario, se modifican los horarios y se genera una nueva propuesta. Estas modificaciones en la programación de horarios de las sesiones de un curso se realizan hasta que el staff de profesores está completamente de acuerdo con los horarios establecidos. Después de la aceptación de los horarios por parte del *staff* de profesores, cada Coordinador por Nivel envía el requerimiento de salas de clase para todas las sesiones del semestre. Esta información se envía en formato impreso, a través de correos y, en pocas ocasiones, en un formulario de uso interno.

Luego de enviar los requerimientos de salas el CN, la Unidad de Gestión de Aulas estudia este requerimiento y, genera una programación preliminar de salas de clase. Si esta programación es factible se genera la programación final de salas de clase. En caso contrario, se realizan ajustes manuales hasta que sea factible la programación de salas de clase evitando los topes horarios dentro de lo posible.

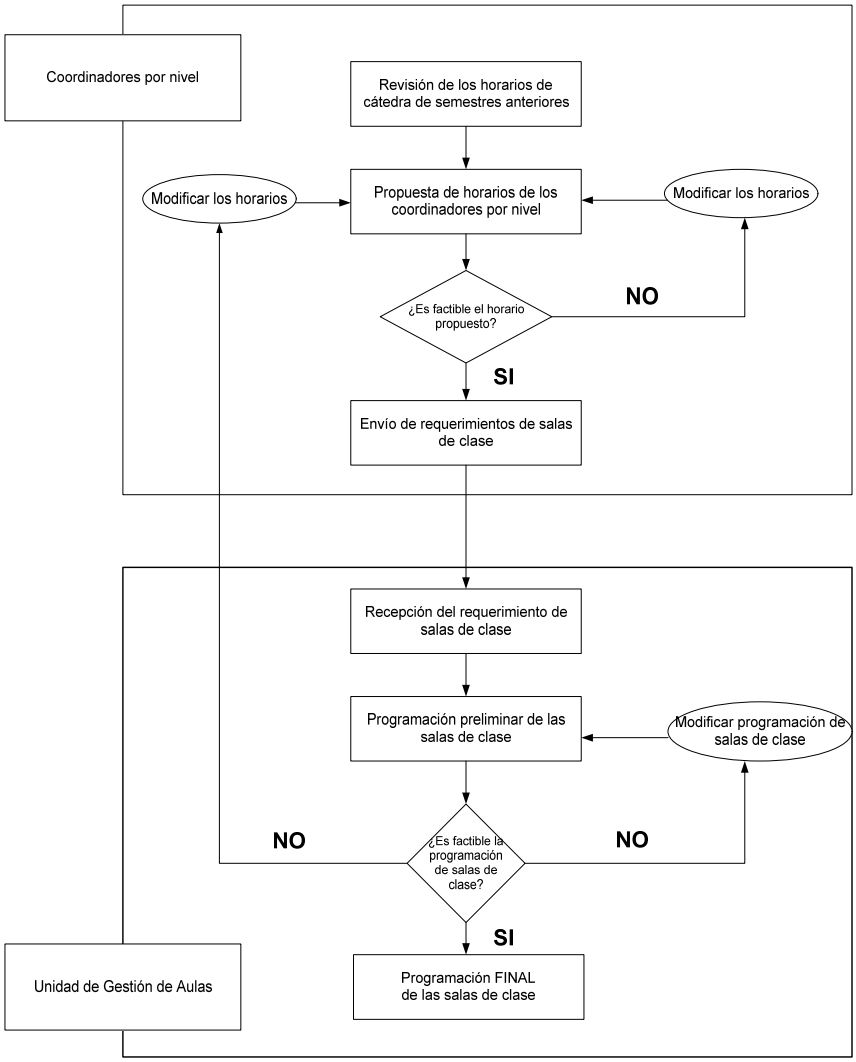

Figura 1. Etapas del proceso de asignación de salas de clase de la Facultad de Medicina.

4. Modelo de programación entera basado en patrones de salas

La idea central del modelo es asignar simultáneamente un conjunto de salas de clase para todas las sesiones de un curso que tengan el mismo requerimiento. Para definir apropiadamente las variables y condiciones del modelo, definimos los conceptos de *patrón de salas* y *familias de patrones*.

Consideremos un curso cualquiera. Este curso puede tener requerimientos por cantidades o tipos diferentes de salas de clase. Por ejemplo, puede requerir en un horario particular un auditorio de 200 personas para una clase y en otro horario, 4 salas de clase de 50 alumnos para otra actividad. Cada combinación posible "tipo de sala"–"cantidad de salas" la llamamos un tipo de requerimiento o *familia de patrones* (de salas). En el ejemplo, se consideran dos tipos de requerimientos: ("Auditorio",1) y ("Salas de 50 asientos", 4).

El conjunto de todas las familias de patrones o tipos de requerimiento se denota por F. Cada tipo de requerimiento f puede ser satisfecho por diferentes combinaciones de distintas salas del tipo apropiado. Cada una de estas combinaciones es un patrón de la familia f.

El conjunto de todos los patrones de la familia f es $B(f)$ y el conjunto $B = \bigcup_{f \in F} B(f)$ es el conjunto de todos los patrones. La familia a la que pertenece el patrón b es $f(b)$.
Sea $F(i)$ el conjunto de las familias de patrones de salas que requiere el curso i a lo largo del semestre. La idea es asignarle a cada curso un patrón de cada uno de los que necesita, esto es, un patrón de cada familia $f \in F(i)$.

Con esta notación, la formulación del modelo es la siguiente.

Variables

Se define un conjunto de variables binarias que indican si un patrón se asigna a un determinado curso. Específicamente,

$$x_{ib} = \begin{cases} 1 & \text{si al curso } i \text{ se le asigna el patrón } b, \\ 0 & \text{en caso contrario.} \end{cases}$$

Estas variables están definidas sólo para los patrones de las familias en $F(i)$.

Restricciones

El modelo tiene dos conjuntos de restricciones: el primer conjunto de restricciones (1), asegura que todos los cursos tengan las salas que requieren, mientras que las desigualdades (2) del segundo grupo, garantizan que las salas de clase no son utilizadas simultáneamente por más de una actividad. La condición que todas las asignaciones correspondientes a un tipo de requerimiento se hagan a las mismas salas de clase, se obtiene por la definición de las variables a partir patrones.

1. Se asignan los patrones necesarios

$$\sum_{b \in B(f)} x_{ib} = 1 \qquad \forall i \in I, f \in F(i). \tag{1}$$

Recordemos que $B(f)$ es conjunto de patrones de la familia f. Entonces, el lado derecho cuenta el número de patrones de una determinada familia f que se asignan al curso i. La ecaución impone que sea exactamente uno para cada tipo de requerimiento $f \in F(i)$. De esta manera, quedan bien definidas cuáles salas de clase se asignan para cada requerimiento de cada curso, y con la información de en qué horarios los cursos tienen cada requerimiento (que se maneja implícitamente) se puede reconstruir todas las asignaciones a lo largo del semestre.

2. No hay tope de salas

$$\sum_{i \in I} \sum_{f \in F(i)} \sum_{b \in B(f)} \sum_{(b,i) \in L(s,t,w)} x_{ib} \leq d_{tw}^{s} \qquad \forall s \in S, t \in T, w \in W. \tag{2}$$

En la desigualdad (2): $L(s,t,w) \subseteq B \times I$ es el conjunto de pares (patrón b, curso i) tales que i tiene una clase en el módulo t de la semana w que tiene un requerimiento de salas de clase de patrón del tipo de b y s es una sala de clase que compone el patrón b ($s \in b$).

Se define d_{tw}^{s} como un parámetro que vale 1 si la sala s está disponible en el módulo (t,w), y vale 0, si no. De esta manera, el lado izquierdo de la desigualdad cuenta el número de veces que es utilizada la sala de clases s en el módulo (t,w). Dependiendo del valor del lado derecho, d_{tw}^{s}, fuerza a que la sala de clase no se utilice si no está disponible ($d_{tw}^{s} = 0$) o que sea asignada a lo más a una actividad, si la sala de clase está disponible ($d_{tw}^{s} = 1$).

Función objetivo

Los criterios son incluidos por el medio de premios y penalidades en las asignaciones individuales, que inducen en el proceso de solución, a buscar asignaciones globalmente más atractivas en la práctica. En el modelo propuesto la función objetivo está compuesta por dos tipos de términos que permiten incorporar y balancear distintos criterios:

a) Penalidades o premios por asignación de patrón a curso: Permiten modelar reglas de asignación de salas de clase por especialidad o para cursos particulares, incentivar o desincentivar asignaciones en franjas horarias o días predefinidos. Las prohibiciones de asignaciones o uso de determinadas salas de clase no se manejan por este mecanismo sino que se introducen como restricciones.

b) Penalidades por uso de salas: Permiten modelar preferencias en el orden en que son asignadas ciertas salas de clase equivalentes (mayor peso a las menos deseadas de ser usadas). Además, permite incluir la intención de dejar ciertas salas bloqueadas en todos los módulos de manera más flexible que simplemente bloquear un conjunto de salas de clase predefinidas, al dejar al modelo seleccionar cuáles salas no utilizar.

5. Resultados computacionales

Se construyó un caso de prueba a partir de las actividades académicas programadas durante el semestre Otoño 2011. Se (consideran) dictan 9 carreras: Enfermería, Fonoaudiología, Kinesiología, Medicina, Nutrición, Obstetricia, Tecnología Médica y Terapia Ocupacional con un total de más de 2.500 alumnos. Las salas de clases fueron divididas en 15 grupos dependiendo de su capacidad, ya que las características tecnológicas y físicas de las salas de clase del mismo tamaño eran similares. Las salas de clase específicas de una especialidad, principalmente, laboratorios, no se consideran en el análisis. La asignación de estas salas no reviste mayor dificultad ya que son utilizadas por un grupo reducido de cursos.

Se resolvieron problemas de optimización separados para programar cada conjunto de requerimientos por tipo sala de manera independiente. La primera mitad de la Tabla 1 muestra el resumen de las características de cada instancia. En la cuarta columna, “Requerimientos” se indica el número de combinaciones diferentes de salas de clase del tipo que solicitan. En estos requerimientos pueden estar consideradas varias solicitudes diferentes del mismo curso para distintos tipos de sesiones. Se puede observar que hay una gran variación en las características de la problemática para los distintos tipos de salas: mientras que hay casos simples con pocos requerimientos, también hay casos que incluyen una gran cantidad de requerimientos y una mayor cantidad de salas.

Resultados

La Tabla 1 muestra las características de las instancias analizadas y las características de los modelos de programación entera resultantes. La última columna presenta los tiempos computacionales de resolución.

Para los casos TIPO 11 y TIPO 13, no fue posible resolver computacionalmente los modelos construidos considerando todos los patrones existentes. Para estos casos se definen un conjunto reducido de patrones. Los patrones seleccionados privilegian combinaciones atractivas en la práctica (salas de clase en el mismo piso o edificio) pero incluyen algunas combinaciones aleatorias para dar más flexibilidad al modelo. Se realizaron experimentos limitando el número de patrones por familia a un máximo entre 200 y 2000, y se observó que con menos de 1000 patrones en algunas selecciones no encontraba una solución factible para el problema, mientras que con más de 1000 patrones, los tiempos de ejecución crecían rápidamente y la calidad de las soluciones no cambiaba. Por estos motivos, para la construcción de los modelos de estas dos instancias, se utilizó un límite máximo de 1000 patrones de cada familia.

Instancia	Número de salas	Capacidad	Requerimientos	Variables	Restricciones	Tiempo de solución (segundos)
TIPO 1	1	600	3	29	29	0.09
TIPO 2	2	230	51	1435	1385	0.2
TIPO 3	3	192	50	2205	2105	0.31
TIPO 4	1	120	24	766	766	0.13
TIPO 5	10	100	199	13400	9719	488.37
TIPO 6	3	90	90	2973	2793	0.28
TIPO 7	4	70	58	2978	2798	0.27
TIPO 8	8	50	56	6312	5840	0.54
TIPO 9	8	45	101	6952	5581	1.38
TIPO 10	9	40	126	11205	7587	121.62
TIPO 11	24	30	154	163648	11988	13263
TIPO 12	2	24	35	1332	1297	0.15
TIPO 13	17	16	101	127898	11746	2852
TIPO 14	5	12	19	2505	2424	0.4
TIPO 15	5	10	11	1450	1396	0.16

Tabla 1. Características de las instancias, de los modelos resueltos y sus tiempos de resolución computacional.

Como se observa, los tiempos necesarios para resolver cada instancia varían notablemente y hay una relación directa entre este tiempo computacional y el tamaño del modelo, el que a su vez tiene relación con el número de requerimientos y de salas de clase disponibles del caso correspondiente.

En todos los casos que era posible, se consiguió mantener la misma combinación de salas de clase para todas las ocurrencias del mismo requerimiento. Esto se logra, asignando salas del requerido y sin la necesidad de usar salas de mayor capacidad a las solicitadas. Los pocos casos en que no fue posible asignar la misma sala, ocurren porque los requerimientos en sí son incompatibles. Estos casos fueron identificados resolviendo el modelo modificado con variables de holgura en las asignaciones de salas como se describe en la Sección 6. Esto presenta una gran

mejora respecto lo conseguido por la asignación manual. Como ilustración, en la Figura 2 se muestra el número de bloques horarios que se asigna a un curso salas diferentes para el mismo requerimiento de acuerdo a la programación manual. Este ejemplo corresponde a un caso en que el modelo encuentra una asignación que repite las salas de acuerdo a lo requerido.

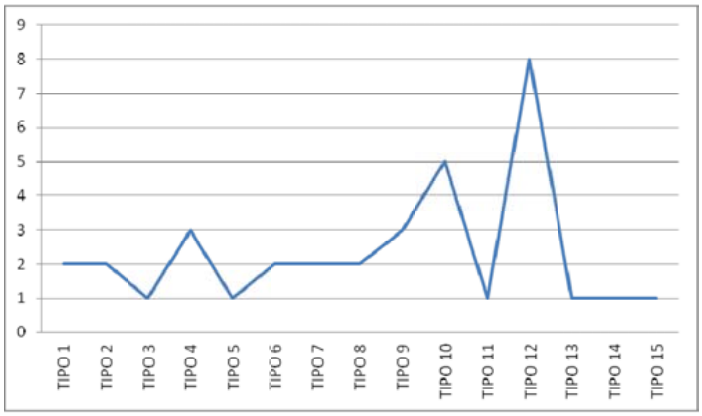

Figura 2. Asignaciones de salas de clases diferentes para el mismo curso, por tipos de sala de clase según asignación manual.

Otra mejora en la asignación construida con el uso del modelo es que, en algunos casos se consigue programar todos los requerimientos sin utilizar todas las salas disponibles. Por ejemplo, para el caso TIPO11, se dispone de un total de 24 salas (y un requerimiento simultáneo máximo de 19 salas) y son utilizadas sólo 21 de estas salas. Esto significa que hay 3 de las salas disponibles (un 12%) que quedan liberadas completamente para otros usos.

6. Discusión

El modelo presentado maneja la factibilidad de las asignación de salas de clase de manera estricta por el medio de restricciones. En ciertos casos, se los puede flexibilizar utilizando variables de holgura que permitan violar algunas condiciones, y utilizar estas variables para penalizar estas violaciones. Esta incorporación puede ser útil para identificar infactibilidades en los modelos con un conjunto reducido de patrones. En ese caso, se puede determinar si no es posible satisfacer los requerimientos por lo que estos deben ser modificados o relajados, o se deben incorporar más patrones en el análisis. Las variables de holgura positivas en la solución, indican los horarios y tipos de salas de clase que no pueden ser satisfechos y dan una guía de qué tipos de patrones es necesario agregar. Este modelo, se resuelve iterativamente hasta encontrar solución factible o tener prueba que no existe.

Las variables de holgura h_{stw} (no negativas) se introducen en las desigualdades (2) para permitir que una sala de clase pueda ser asignada más de una vez, señalando un conflicto en ese tipo de salas de clase con los requerimientos en ese bloque horario. Estas variables son también penalizadas en la función objetivo para inducir que sólo sean activadas en caso inevitable. Las restricciones (2) del modelo de la Sección 4, son sustituidas por las desigualdades:

$$\sum_{i\in I}\sum_{f\in F(i)}\sum_{b\in B(f)}\sum_{(b,i)\in L(s,t,w)} x_{ib} \le d^{s}_{tw} + h_{stw} \qquad \forall s \in S, t \in T, w \in W. \tag{2'}$$

7. Conclusiones

Se propone un modelo de programación entera basado que asigna de manera simultánea todas las salas de clase requeridas para un conjunto de clases a lo largo del semestre para los cursos de una Facultad de Medicina. Las características del problema considerado determinan que la programación deba ser hecha para el semestre completo y que se asignen diferentes combinaciones de salas de clase para cada curso. El modelo se aplica para un caso de prueba con datos reales obteniéndose buenos resultados.

Se observa que puede haber una gran diferencia en las dificultades de las asignaciones dependiendo del número de salas de clase disponibles y la complejidad de los requerimientos a atender. En muchos casos en los que se dispone de pocas salas de clase de un determinado tipo, es relativamente fácil encontrar una asignación factible o detectar los conflictos. Sin embargo, para los casos más complejos, el apoyo del modelo de optimización es fundamental para encontrar asignaciones que mantuvieran las salas de clase por todo el semestre y que eliminaran los conflictos.

Respecto de la detección de conflictos entre los requisitos, el modelo con variables de holgura en las asignaciones de salas de clase fue de mucha utilidad. El uso de este modelo de optimización no sólo sirvió para identificar los conflictos, sino que también permitió determinar asignaciones que minimizaran el número de sesiones que deben ser asignadas sin respetar el patrón. En el caso considerado, estos conflictos estaban provocados por ciertos requerimientos puntuales no regulares de algunos cursos.

En la aplicación práctica, el uso de este modelo podría permitir identificar estos conflictos antes de consolidar la programación de los cursos, dando la oportunidad al coordinador del curso correspondiente de revisar los requerimientos de su asignatura antes del inicio del semestre si deseara evitar el uso de salas diferentes.

Finalmente, el modelo consigue determinar asignaciones que utilizan la menor cantidad de salas posibles permitiendo liberar recursos de salas de clase de manera permanente para otras actividades.

Referencias

Al-Yakoob, S. y Sherali, H. (2007), A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations, *European Journal of Operational Research*, 180, 1028–1044.

Avella, P. y Vasil'ev, L. (2004), A computational study of a cutting plane algorithm for university course timetabling, *Journal of Scheduling*, 8, 497–514.

Beyrouthy, C., Burke, E., McCollum, B., McMullan, P. y Parkes, A.J. (2010), University space planning and space–type profiles, *Journal of Scheduling*, 13, 363–374.

Burke, E. (2002), Recent research directions in automated timetabling, *European Journal of Operational Research*, 140, 266–280.

Carter, M. y Tovey, C. (1992), When is the classroom assignment problem hard?, *Operations Research*, 40, S28–S39.

Daskalaki, S. y Birbas, T. (2005), Efficient solutions for a university timetabling problem through integer programming, *European Journal of Operational Research*, 160, 106–120.

Daskalaki, S., Birbas, T. y Housos, E. (2004), An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research*, 153, 117–135.

Dimopoulou, M. y Miliotis, P. (2001), Implementation of a university course and examination timetabling system, *European Journal of Operational Research*, 130, 202–213.

Glassey, C.R. y Mizrac, M. (1986), A decision support system for assigning classes to rooms, *Interfaces*, 92–100.

Kingston, J. (2012), Resource assignment in high school timetabling, *Annals of Operations Research*, 194, 241–254.

Kolen, A., Lenstra, J.K., Papadimitriou, C. y Spieksma, F. (2007), Interval scheduling: A survey, *Naval Research Logistics*, 54, 530–543.

Lewis, R. (2008), A survey of metaheuristics-based techniques for university timetabling problems, *OR Spectrum*, 30, 167–190.

Miranda, J. (2010), eClasSkeduler: a course scheduling system for the Executive Education Unit at the Universidad de Chile, *Interfaces*, 40, 196–207.

MirHassani, S. (2006), A computational approach to enhancing course timetabling with integer programming, *Applied Mathematics and Computation*, 175, 814–822.

Schaerf, A. (1999), A survey of automated timetabling, *Artificial Intelligence Review*, 13, 87–127.

Stallaert, J. (1997), Automated timetabling improves course scheduling at UCLA, *Interfaces*, 27, 67–81.