

Un modelo de programación entera para el problema de horarios universitarios

Francisco, I.; Marcial Romero, J. R.; Hernández-Servín, J. A.

Información del Artículo	Resumen
Recibido: 20-mar-2023 Aceptado: 28-may-2023	Se presenta un modelo de programación entera para resolver un caso de estudio para el problema de horarios universitarios de la Facultad de Ingeniería de la Universidad Autónoma del Estado de México. El modelo toma en consideración las preferencias de días y horas que el personal académico puede elegir para impartir clases dentro de una escala de 0 a 5, donde 0 representa la mayor preferencia y 5 representa la menos preferida para dar clases. Se presentan dos formulaciones para la generación de horarios con clases en períodos consecutivos, estas formulaciones se comparan con las propuestas de otros autores.
Palabras clave: Programación entera, Timetabling, Períodos consecutivos	

1. Introducción

El proceso de generación de horarios escolares se conoce como Carga Horaria o Timetabling, se dice que un horario es eficiente si presenta la menor cantidad de solapamientos entre clases y si además cumple con una serie de condiciones impuestas por la institución académica [1].

La generación de horarios puede variar dependiendo del nivel educativo y de las condiciones que se deben cumplir, es por ello que en [2] se propone la siguiente clasificación: - Carga horaria escolar: es la programación semanal de todas las clases de una escuela, evitando que los profesores tenga dos clases o más al mismo tiempo. - Carga horaria universitaria: es la programación semanal de todas las clases de un conjunto de cursos de alguna universidad, minimizando los conflictos entre clases que tienen alumnos en común. - Carga horaria de exámenes: esta es la programación de un conjunto de exámenes de una universidad, evitando programar al mismo tiempo dos exámenes con alumnos en común, además se debe dar el tiempo necesario entre exámenes para que los alumnos tengan un periodo de descanso y preparación.

Se sabe que el problema de la carga horaria es NPCompleto [1] [2] [3], este es el motivo por el cual encontrar una solución sin conflictos puede llevar un tiempo exponencial, ya que para ello se requiere examinar las combinaciones posibles entre los conjuntos de asignaturas, profesores, alumnos, salones y periodos de tiempo, para obtener aquella combinación en la cual no exista ningún conflicto o en su caso aquella que contenga la menor cantidad. Debido a esto, el problema de la carga horaria se ha abordado desde diferentes áreas y enfoques, se ha tomado como un problema de búsqueda, ya que el objetivo es encontrar una solución que cumpla con las condiciones impuestas por la institución, para ello se ha aplicado técnicas como búsqueda tabú, temple simulado, algoritmos genéticos, y búsquedas basadas en heurísticas [2], las cuales garantizan encontrar una solución cercana a la óptima. Otro enfoque que se ha dado, es tratar al problema de la carga horaria como un problema de optimización, ya que no únicamente se busca encontrar una

solución, sino aquella que tenga la menor cantidad de conflictos, por lo cual se puede abordar por medio de Programación Lineal y resolver utilizando los algoritmos desarrollados para el área de Investigación de operaciones como Algoritmo de Ramificación y Acotamiento [4], Algoritmo de Ramificación y Corte [4] [5].

Para probar la propuesta que se presenta en este artículo, se ha tomado como caso de estudio la Facultad de Ingeniería (FI) de la Universidad Autónoma del Estado de México (UAEM) que ofrece las carreras de Ingeniería Civil, Ingeniería en Computación, Ingeniería en Electrónica, Ingeniería Mecánica y Sistemas Energéticos Sustentables. El proceso de generación de horarios escolares consiste en lo siguiente, cada profesor de cada carrera plantea el horario en que desea impartir sus clases, estos horarios se compilan por cada coordinación y son enviados a Control Escolar donde se autorizan siempre y cuando no entren en conflicto con horarios de otras carreras, ya sea porque no existen suficientes salones disponibles o porque los días o periodos ya han sido asignados, en caso contrario se reasignan.

La FI, a diferencia de las demás Facultades pertenecientes a la UAEM, oferta en cada semestre todas las asignaturas dentro del plan de estudios de cada carrera, los alumnos que se inscriben deben elegir qué materias cursarán, esto permite que los alumnos puedan adelantar asignaturas de otros semestres siempre y cuando no estén serializadas con asignaturas que aún no han cursado. Debido a esto la FI crea grupos de alumnos en base a las asignaturas, por lo cual un alumno puede pertenecer a un grupo α en la asignatura A y su vez pertenecer al grupo β en la asignatura B, donde A y B se intersectan lo cual dificulta la generación de horarios. Otra característica que presenta la FI, es que existen asignaturas que son de tronco común para las cinco carreras, los grupos de estas asignaturas pueden estar formados por alumnos de distintas carreras y distintos semestres.

En este artículo se presenta un modelo de programación entera para la generación de horarios escolares de la FI que toma en cuenta las preferencias de los profesores, y plantea los horarios en base a la tupla grupo-alumno.

Este modelo es un prototipo para el desarrollo de un sistema automatizado para la generación de horarios escolares de la FI.

2. Modelo para el caso de estudio

Daskalaki en [1] propone un modelo de programación entera 0-1 para el problema de carga horaria de la Universidad de Patras, en ésta se ofrecen cursos de dos tipos: obligatorios y opcionales. Los cursos obligatorios están diseñados para los alumnos de los primeros años, por el contrario los cursos opcionales son para los alumnos que están en los últimos años de la carrera, además en el modelo se toma en cuenta que algunos de los cursos ofrecidos requieren de laboratorios en lugar de un salón clases. En [1] se toman en cuenta las siguientes restricciones: - Cada profesor debe ser asignado a lo más a un curso, un grupo de estudiantes y a un salón a la vez. - Para cada grupo de estudiantes a lo más un curso, un profesor y un salón debe ser asignado en cada periodo. - Cada salón debe ser asignado a lo más un curso, un profesor y a un grupo de estudiantes a la vez. - Cada curso en el plan de estudios de cada estudiante debe ser programado en la cantidad exacta de periodos de tiempo. - Cada curso debe ser programado tantos periodos como el plan de estudios de cada grupo de estudiantes requiera. - Cada profesor debe

ser asignado a tantos periodos a la semana como su carga de trabajo requiera. - Un curso c que requiere una sesión de $h_v \in H_c$ periodos consecutivos, debería ser asignado exactamente h_v periodos en el día especificado. - Si h_v periodos de un día en específico se asignan al curso c , estos deben ser consecutivos. - Las h_v sesiones deben ser tantas como se requieran semanalmente y deben tener por lo menos una sesión al día.

El modelo Daskalaki et al es uno de los más completos ya que, toma en consideración la mayoría de restricciones al problema: cursos, alumnos, profesores, salones, días y periodos de tiempo y su objetivo es minimizar los conflictos entre las clases, Para Daskalaki et. al. un horario debe cumplir cuatro condiciones: - El horario no debe tener colisiones tanto para cursos, alumnos, profesores y salones. - El horario debe ser completo para ello todos los cursos deben ser programados la cantidad requerida por el plan de estudios - El horario debe ser flexible tomando en consideración si el profesor desea impartir su curso en varias sesiones consecutivas en un día o repartidas durante la semana. - El horario debe permitir la programación de una sesión en dos o más clases, por ejemplo si un curso requiere de un laboratorio cuya capacidad sea menor a la cantidad de alumnos inscritos al curso, el horario debe ser capaz de programar dos o más sesiones a la semana.

Se puede desarrollar un modelo específico para la Facultad de Ingeniería de la Universidad Autónoma del Estado de México, tomando como base el modelo de Daskalaki.

Restricciones de la Facultad de Ingeniería

Los horarios generados para la Facultad de Ingeniería deben cumplir con las siguientes restricciones: - Ningún profesor puede impartir más de una clase al mismo tiempo

- Ningún salón puede ser ocupado por más de una clase al mismo tiempo - Ningún salón puede ser ocupado por una clase cuyo grupo de alumnos supere su capacidad - Cada asignatura debe ser programa tantas horas como se especifique en el plan de estudios - Una asignatura puede ser programa a lo máximo 2 horas diarias - Si una asignatura se programa en más de un periodo en un día estos deben ser consecutivos

3. Modelo para la FI

En esta sección se presenta el modelo de programación entera para la Facultad de Ingeniería de la Universidad Autónoma del Estado de México.

Notación

Conjuntos del modelo: - $D = (1, \dots, 6)$, días en que la facultad está disponible para impartir clases. - $H = \{1, \dots, 30\}$, periodos de tiempo en que la facultad puede impartir clases, estos periodos están compuestos por 30 minutos. - $P = \{1, \dots, |P|\}$, profesores que imparten clases en la facultad. - $C = \{1, \dots, |C|\}$, grupo-asignatura quienes reciben clases. - $S = \{1, \dots, |S|\}$, salones en los cuales se imparten clases. Subconjuntos: - P_c , profesores que imparten clases al grupo-asignatura c . - P_{dh} ,

profesores disponibles en el día d y la hora h . - C_p , grupo-asignatura a los que imparte clase el profesor p . - C_{ps} , clases que imparte el profesor p y puede ser albergada en el salón s . - S_c , salones en que la clase c no supera su capacidad. - D_p , días en que el profesor p está disponible. - H_{pd} , horas del día d en que el profesor p está disponible. - HF_{pd} , hora inicial en el día d en que el profesor p está disponible. - h_c , horas semanales requeridas por la asignatura c . - h_d , horas diarias máximas

Variables

Para el modelo se tiene la siguiente variable de decisión x_{dhpes} . La cual toma el valor de 1 si se programa un clase en el día d en el periodo h , y es impartida por el profesor p al grupoasignatura c en salón s , en caso de que no se programe alguna clase, la variable toma el valor de 0.

Función Objetivo

La función objetivo está orientada a tomar en consideración las preferencias de que día y hora el profesor prefiere impartir sus clases, para ello se hace uso de la constante $T_{dhpc} \in (0, 1, 2, 3, 4, 5)$ donde 0 indica mayor preferencia y 5 la menor.

$$\min \sum_{d \in D} \sum_{h \in H} \sum_{p \in P_{dh}} \sum_{c \in C_p} \sum_{s \in S_c} T_{dhpc} x_{dhpcs}$$

Restricciones Básicas

Ningún profesor puede impartir más de un clase al mismo tiempo.

$$\forall d \in D, \forall h \in H, \forall p \in P_{dh} \quad \sum_{c \in C_p} \sum_{s \in S_c} x_{dhpcs} \leq 1$$

Ningún salón puede ser ocupado por más de una clase al mismo tiempo.

$$\forall s \in S, \forall d \in D, \forall h \in H \quad \sum_{p \in P_{dh}} \sum_{c \in C_{ps}} x_{dhpcs} \leq 1$$

Cada asignatura debe ser programa tantas horas semanales como se especifique en el plan de estudios.

$$\forall c \in C, \forall p \in P_c \quad \sum_{d \in D_p} \sum_{h \in H_{pd}} \sum_{s \in S_c} x_{dhpcs} = h_c$$

Cada asignatura debe programarse a lo mucho 2 horas dias.

$$\forall d \in D, \forall p \in P_d, \forall c \in C_p \quad \sum_{s \in S_c} \sum_{h \in H_{pd}} x_{dhpcs} \leq h_d$$

4. Restricciones clases consecutivas

En esta sección se muestra cuatro técnicas para obtener clases consecutivas, una basada en una extensión del modelo propuesto por Daskalaki en [1], otra en una extensión del modelo de Barki y Aksop propuesto en [6], y dos restantes ideadas en este artículo con la finalidad de obtener la mejor solución posible.

Restricción Daskalaki

Daskalaki en [1] propone una serie de restricciones para obtener clases consecutivas, adaptadas a la Facultad de Ingenieria quedando como:

$$\begin{aligned} &\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall f \in HF_{pd}, \\ &\forall t \in \{1, \dots, h_v - 1\} \quad x_{dfpcs} - x_{d(f+t)pcs} \leq 0 \\ &\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall h \in H_{pdt} \\ &\forall t \in \{2, \dots, h_v\} \\ &\quad -x_{dhpcs} + x_{d(h+1)pcs} - x_{d(h+\varepsilon)pcs} \leq 0 \end{aligned}$$

Donde HF_{pd} es el periodo inicial del conjunto de los periodos en que el profesor p está disponible para impartir clases en el día d , h_v es el número de periodos consecutivos requeridos para la clase c .

Ahora supóngase que existan 4 periodos $X = \{x_1, x_2, x_3, x_4\}$, donde se desea programar una clase con $h_v = 2$ periodos consecutivos. Las restricciones generadas en base a Daskalaki son:

$$\begin{aligned} x_1 - x_2 &\leq 0 \\ -x_1 + x_2 - x_3 &\leq 0 \\ -x_2 + x_3 - x_4 &\leq 0 \end{aligned}$$

Uno de los posibles valores para X que cumpla con las restricciones anteriores es $X = \{1, 1, 0, 1\}$, es decir se pueden programar clases consecutivas permitiendo periodos vacios. En dado caso que únicamente se requiere programar una única clase, las restricciones de Daskalaki únicamente permiten que sea en el periodo final, por lo cual $X = \{1, 0, 0, 0\}$, $X = \{0, 1, 0, 0\}$ y $X = \{0, 0, 1, 0\}$ no son aceptadas.

Restricción Barki y Aksop

Barki y Aksop en [6] agregan la siguiente restricción a las propuestas por Daskalaki en [1]. $\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall l \in HL_{pd}, \forall t \in \{1, \dots, h_p - 1\} \quad x_{dhpcs} - x_{d(1-t)pes} \leq 0$ Donde HL_{pd} es el periodo final del conjunto de los periodos en que el profesor p está disponible para impartir clases en el día d .

Al igual que Daskalaki, Barki y Aksop permiten huecos y únicamente permite la programación de clases consecutivas de h_v o más horas consecutivas.

Formulación 1

La siguiente restricción esta ideada para programar las clases desde 0 hasta el número total de periodos disponibles de los profesores. Para el desarrollo de esta restricción se tomó como base la propuesta por Avella en [6]. Supóngase que existen únicamente 3 periodos consecutivos $X = \{x_1, x_2, x_3\}$, una clase se puede programar de la siguiente manera: - No se programe en ningún periodo - Se programe en solo uno de los periodos - Se programe en dos periodos que sean consecutivos - Se programe en los 3 periodos disponibles Lo anterior se puede cumplir con la siguiente restricción:

$$x_1 - x_2 + x_3 \leq 1$$

Para cualquier cantidad de periodos, la restricción puede generalizarse como:

$$\forall j \in J, \forall k \in \{j+2, \dots, |J|\} \quad x_j - x_{j+1} + x_k \leq 1$$

Donde J es el conjunto de periodos. La idea radica en tomar tres periodos y determinar que el segundo periodo al tomar el valor de 0 no provoque huecos, ver Tabla I.

TABLA I. FORMULACIÓN 1 PARA 3 PERÍODOS DE TIEMPO

Periodos			Restricción
x_j	x_{j+1}	x_k	$x_j - x_{j+1} + x_k \leq 1$
0	0	0	V
0	0	1	V
0	1	0	V
0	1	1	V
1	0	0	V
1	0	1	F
1	1	0	V
1	1	1	V

Para el modelo, las restricciones que determinan que las clases se programen en periodos consecutivos en base a esta formulación queda como:

$$\forall d \in D, \forall p \in P_d, \forall c \in C_p \quad \sum_{s \in S_c} y_{dcps} \leq 1$$

$$y \in \{0, 1\}$$

$$\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall h \in H$$

$$x_{dhpcs} \leq y_{dcps}$$

$$\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall h \in H_{pd},$$

$$\forall t \in \{h+2, \dots, |H_{pd}|\}$$

$$x_{dhpcs} - x_{d(h+1)pcs} + x_{dtpcs} \leq 1$$

Donde (13) determina el salón en que se programará la clase, la restricción (14) asegura que no se programen clases en más de un salón y la restricción (15) determina los periodos consecutivos. D. Formulación 2 Se puede obtener una segunda formulación utilizando el siguiente enfoque, supongamos que se tienen dos periodos cualquiera x_j y x_{j+1} , donde x_j es el periodo en el cual deben iniciar las clases en periodos consecutivos, por lo cual una clase se puede programar de la siguiente manera: - No se programe en ningún periodo. - Se programe en solo uno de los periodos. - Se programe en los dos periodos. Estos casos se pueden obtener por medio de la siguiente restricción, ver Tabla II:

$$x_i - x_{i+1} \geq 0$$

Para el caso de 3 periodos x_j, x_{j+1}, x_{j+2} , las siguientes restricciones son necesarias para obtener clases consecutivas con el periodo inicial en x_j :

$$x_j - x_{j+1} \geq 0$$

$$x_{j+1} - x_{j+2} \geq 0$$

TABLA II FORMULACIÓN 2 PARA 2 PERIODOS DE TIEMPO

Periodos		Restricción
x_j	x_{j+1}	$x_j - x_{j+1} \geq 0$
0	0	V
0	1	F
1	0	V
1	1	V

Si el periodo inicial es en x_{j+1} , se requieren la restricción (18) y la siguiente:

$$x_i \leq 0$$

En cambio si el periodo inicial es en x_{j+2} se requieren las restricciones (17) y (19). Se pueden agrupar las tres restricciones (17), (18) y (19), y dependiendo del periodo inicial desactivar la restricción correspondiente. Para x_j la restricción (19) se desactiva, para x_{j+1} la restricción (17) y para x_{j+2} la restricción (18), ver la Tabla III.

TABLA III FORMULACIÓN 2 PARA 3 PERIODOS DE TIEMPO

Periodos			Restricciones		
x_j	x_{j+1}	x_{j+2}	$x_j - x_{j+1} \geq 0$	$x_{j+1} - x_{j+2} \geq 0$	$x_j \leq 0$
0	0	0	V	V	V
0	0	1	V	F	V
0	1	0	F	V	V
0	1	1	F	V	V
1	0	0	V	V	F
1	0	1	V	F	F
1	1	0	V	V	F
1	1	1	V	V	F

Para desactivar restricciones se hace uso de la técnica K out of N Constraints Must Hold [7] [8] [9], que establece que dadas N posibles restricciones:

$$f_1(x_1, x_2, \dots, x_n) \leq d_1$$

$$f_2(x_1, x_2, \dots, x_n) \leq d_2$$

$$\vdots$$

$$f_N(x_1, x_2, \dots, x_n) \leq d_N$$

Se pueden tener N restricciones activas haciendo uso de la siguiente formulación:

$$f_1(x_1, x_2, \dots, x_n) \leq d_1 + My_1$$

$$f_2(x_1, x_2, \dots, x_n) \leq d_2 + My_2$$

$$\vdots$$

$$f_N(x_1, x_2, \dots, x_n) \leq d_N + My_N$$

$$\sum_{i=1}^N y_i = N - K$$

Donde y_i es una variable auxiliar binaria que toma el valor de 1 si la restricción esta desactivada o 0 si esta activada. M es un entero positivo lo suficientemente grande de tal manera que la restricción se satisfaga para cualquier valor de las variables, en este caso $M = 1$. De esta manera se obtiene la siguiente formulación para 3 periodos:

$$\begin{aligned}x_j &\leq y_j \\-x_j + x_{j+1} &\leq y_{j+1} \\-x_{j+1} + x_{j+2} &\leq y_{j+2} \\y_j + y_{j+1} + y_{j+2} &= 1\end{aligned}$$

Si se sustituye y_j por x_j en (25), se puede eliminar la restricción (22) y obtener el mismo resultado. Para el modelo de la Facultad de Ingeniería, la formulación que determina que las clases se programen de manera consecutivas queda como:

$$\begin{aligned}\forall d \in D, \forall p \in P_d, \forall c \in C_p, \forall s \in S_c, \forall h \in H_{pd} \\-x_{dhpc s} + x_{d(h+1)pc s} &\leq y_{d(h+1)pc s} \\ \forall d \in D, \forall p \in P_d, \forall c \in C_p \\ \sum_{s \in S_c} \sum_{f \in HF_{pd}} x_{dhpc s} + \sum_{s \in S_c} \sum_{h \in H_{pd}} y_{d(h+1)pc s} &\leq 1\end{aligned}$$

Con la restricción (27) además de determinar el periodo inicial, también se determina el salón en que se programen las clases.

Comparación entre formulaciones

Para eyley y Nemhauser en [10] no basta únicamente con el número de restricciones y el número de variables para determinar la calidad de una formulación, sino que influye desde el tamaño de los números en las operaciones básicas hasta el algoritmo que se usa para resolverlo. Sin embargo Wolsey y Nemhauser establecen que para un programa entero, la complejidad en tiempo al resolverse por medio de fuerza bruta es de:

$$O(2^n mn)$$

Donde n es el número de variables y m el número de restricciones.

Las restricciones generadas, para una única clase, por las diferentes formulaciones es de: - Daskalaki

$$\begin{cases} \frac{(h_v-1)(2|H_{pd}|-h_v)}{2} & 2 \leq h_v < |H_{pd}| \\ \frac{|H_{pd}|(|H_{pd}|-1)}{2} & h_v = |H_{pd}| \end{cases}$$

- Bakir

$$\begin{cases} \frac{(h_v-1)(2|H_{pd}|-h_v+2)}{2} & 2 \leq h_v < |H_{pd}| \\ \frac{(|H_{pd}|-1)(|H_{pd}|-2)}{2} & h_v = |H_{pd}| \end{cases}$$

- Formulación 1

$$\frac{(|H_{pd}|-2)(|H_{pd}|-1)}{2} + 2$$

- Formulación 2

$$|H_{pd}|$$

Las variables generadas por las diferentes formulaciones es de: - Daskalaki

$$|H_{pd}|$$

- Bakir

$$|H_{pd}|$$

- Formulación 1

$$|H_{pd}| + 1$$

- Formulación 2

$$2|H_{pd}| - 1$$

Con base en la complejidad de Wolsey y Nemhauser, la formulación 1 es la más fácil de resolver a comparación de la formulación 2 debido a la cantidad de variables que maneja.

5. Solución del Modelo

Para probar el modelo se uso el solucionador CBC (Coin-or branch and cut) en su versión 2.8 en un equipo con procesador i7 y 8 GB de memoria ram.

Control Escolar de la Facultad de Ingeniería proporcionó datos de sus profesores y sus cursos para el periodo escolar 2015 B, estos se dividieron en base a las carreras que ofrece la facultad, tomando las asignaturas de tronco común como una instancia adicional, ver la Tabla IV.

		PROFESORES CLASES SALONES		
Tabla 4 Instancias	CIVIL	78	112	22
	COMPUTACIÓN	81	130	22
	ELECTRÓNICA	42	61	22
	MECÁNICA	54	80	22
	ENERGÉTICOS	25	41	22
	COMÚN	137	230	22

Se tomaron en cuenta las siguientes consideraciones

1. La disponibilidad de los Profesores se asignaron de manera aleatoria.
2. La cantidad de horas semanales de las clases se asignó de manera aleatoria.
3. Las preferencias de los Profesores se asignaron de manera aleatoria.
4. Se utilizó la instancia Energéticos.
5. Se dio un tiempo máximo de ejecución de 2 horas.
6. Para las formulaciones Daskalaki y Bakir se uso $h_v = 2$.

6. Resultados

Los resultados de la ejecución del modelo utilizando Coin CBC, con la instancia Energéticos se observan en la Tabla V.

En la Tabla V se observa que la formulación de Bakir y la Formulación 1 propuesta en este documento no pueden dar ninguna solución entera en un periodo menor a 2 horas. Tanto Daskalaki y la Formulación 2, resuelven el modelo en menos de 2 horas sin embargo Daskalaki requiere de mayor cantidad de iteraciones y con ello de mayor tiempo. Basándonos en el valor de objetivo, la Formulación 2 toma en cuenta mejor las preferencias de los profesores.

TABLA V RESULTADOS DE LA EJECUCION DE INSTANCIA ENERGETICOS

	Daskalaki	Bakir	Formulación #1	Formulación #2
Valor objetivo lineal	252.33	209.20	128.00	209.50
Valor objetivo entero	259.00	-	-	213.00
Nodos enumerados	2486	-	-	50
Iteraciones	131289	-	-	2764
Tiempo (seg)	205.74	-	-	94.91

Para comparar los horarios generados por las diferentes formulaciones se tomó el profesor con identificador 177, que imparte las clases Ciclos Avanzados con identificador 1053 de 7 horas semanales, Aplicaciones No Eléctricas de la Geotermia con identificador 474 y 9 horas semanales, y Sistemas Geotérmicos con identificador 994 y 11 horas semanales.

Aunque con la formulación de Daskalaki genera horarios en poco tiempo, presenta huecos, esto se puede ver con la clase 474, el día lunes donde existe una separación de 13 periodos entre el segundo y tercer periodo de la clase. Otro problema que presenta Daskalaki es que logra programar clases consecutivas pero en diferentes salones tal es el caso de la clase 1053 en el día viernes. En el horario generado por la formulación 2, las clases se programan correctamente pero tiende a programar clases de un solo período de tiempo en distintos días de la semana.

TABLA VI Horario generado utilizando Daskalaki

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7 : 00			994			1053
7 : 30	474.		994			1053
8 : 00	474					
8,30		474				
9 : 00		474			1053	
9 : 30	994.			994	1053	
10 : 00	994	474		994		
10 : 30	994.	474				
11 : 00						
11 : 30						
12 : 00						
12 : 30						
13 : 00						
13 : 30						
14 : 00					994	
14 : 30					994	1053
15 : 00	474			474		1053
15 : 30				474		1053

TABLA VII Horario generado utilizando Formulación 2

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7 : 00			994			
7 : 30	474		994			
8 : 00	474					
8 : 30					474	
9 : 00					1053	
9 : 30	994				1053	994
10 : 00	994	474				
10 : 30	994	474				
11 : 00						
11 : 30						
12 : 00						
12 : 30		1053				
13 : 00		994		994		
13 : 30		994				474
14 : 00	1053		474		994	
14 : 30					994	1053
15 : 00				474		1053
15 : 30				474		1053

Se realizó una segunda prueba con la instancia Electrónica, la segunda instancia con la menor cantidad de datos, comparando únicamente a Daskalaki y la Formulación 2, se obtienen los resultados mostrados en la Tabla VIII.

En este caso la Formulación Daskalaki, no fue capaz de resolver la instancia de Electrónica en menos de 2 horas.

TABLA VIII RESULTADOS DE LA EJECUCIÓN
DE LA INSTANCIA ELECTRÓNICA

	Daskalaki	Formulacion 2
Valor objetivo real	418.41	337.88
Valor objetivo entero	-	346.00
Nodos enumerados	-	0
Iteraciones	-	1328
Tiempo (seg)	-	327.45

7. Conclusiones

Como se observa en los resultados la formulación 2 obtiene mejores resultados al programar las clases en periodos consecutivos en un mismo salón, esta formulación además presenta el mejor valor objetivo, es decir el modelo toma más en consideración las preferencias de los profesores. Las restricciones de Daskalaki aunque da soluciones en un tiempo similar a la formulación 2, son malas ya que presenta huecos además de cambios de salón. La formulación 2 sin embargo en ocasiones puede programar clases de un solo periodo por día, pero esto es preferible a tener clases consecutivas en diferentes salones como los horarios generados por Daskalaki, además de generar horarios en menor cantidad de tiempo. Con los resultados actuales se puede afirmar que la formulación 2 es la adecuada para aplicarse a la Facultad de Ingeniería de la Universidad Autónoma del Estado de México ya que genera soluciones adecuadas a las restricciones de la facultad.

Bibliografía

- [1] S. Daskalaki, T. Bitbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153:117–135, 2004.
- [2] A. Schaerf. A Survey of Automated Timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.
- [3] D. de Werra. The combinatorics of timetabling. *European Journal of Operational Research*, 96:504–513, 1997.
- [4] E. K. Burke, J. Marecek, A. J. Parkes, and H. Rudova. A branch-and-cut procedure for the Udine Course Timetabling problem. *Annals of Operations Research*, 194:71–87, 2011.
- [5] P. Avella and I. Vasil'Ev. A Computational Study of a Cutting Plane Algorithm for University Course Timetabling. *Journal of Scheduling*, 8:497–514, 2005.
- [6] M. A. Bakir and C. Aksop. A 0-1 integer programming approach to a university timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 37(1), 2008.
- [7] F. Hillier and G. Lieberman. *Introduction to Operations Research*. McGraw-Hill International Editions, McGraw-Hill, 2009.
- [8] D.-S. Chen, R. G. Batson, and Y. Dang. *Applied Integer Programming*. Wiley-Blackwell, 2009.
- [9] H. P. Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.
- [10] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*. Wiley-Interscience, 1999.

Direcciones de correo de los autores

FRANCISCO, I.: malopezme@uaemex.mx

MARCIAL ROMERO, J. R.: jrmarcialr@uaemex.mx

HERNÁNDEZ-SERVÍN, J. A.: xoseahernandez@uaemex.mx