

---

# UN NUEVO ENFOQUE PARA LA PROGRAMACIÓN HORARIA EN UNIVERSIDADES

---

ISABEL MÉNDEZ-DÍAZ \*  
JUAN JOSÉ MIRANDA-BRONT \* \*\*  
PAULA ZABALA \* \*\*

## Resumen

En este trabajo consideramos un problema de programación horaria proveniente de una aplicación real de una universidad privada de Argentina. El objetivo es diseñar horarios que contemplen simultáneamente los recursos disponibles de la universidad (docentes, aulas, horarios, etc.) y las preferencias de los alumnos sobre las materias que desean cursar. Nuestra propuesta es un algoritmo heurístico basado en programación lineal entera que, de acuerdo a la experiencia obtenida con datos reales provenientes de la universidad, mostró una excelente performance en la calidad de las soluciones obtenidas medida conforme a la satisfacción estudiantil alcanzada en base a sus preferencias.

**Palabras Clave:** Programación lineal entera, Metaheurística, Programación de horarios

---

## 1. Introducción

---

En el contexto de entidades educativas, la programación horaria suele ser una problemática difícil de resolver debido a la multiplicidad de requerimientos que debe satisfacer una solución. De acuerdo al escenario que se desee abordar, los trabajos existentes en la literatura se enmarcan fundamentalmente en tres tipos de problemas. Por un lado, la **Programación de exámenes (ETP)**

---

\* Departamento de Computación, FCEyN, Universidad de Buenos Aires

\*\* Consejo Nacional de Investigaciones Científicas y Técnicas

[4, 8, 16], cuyo objetivo es programar horarios y asignar aulas a exámenes en un período de tiempo, satisfaciendo un conjunto de restricciones, por ejemplo que no haya superposición horaria entre exámenes que debe rendir un mismo alumno. Un segundo enfoque es la **Programación de horarios de materias post inscripción (PECTP)** [16] donde el fin es programar horarios y asignar aulas a un conjunto de materias dentro de un período de tiempo, satisfaciendo la selección previamente realizada por los alumnos. Hay diferentes restricciones a tener en cuenta, en particular que las clases correspondientes a las materias elegidas por un mismo alumno no tengan superposición horaria. Finalmente, la **Programación de currícula (CCTP)** [3, 12, 14, 16] busca diagramar horarios y asignar aulas respetando la currícula de cada carrera, por ejemplo las clases de las materias que deben cursarse en el mismo período no deben superponerse en horario.

Una de las principales diferencias entre estos dos últimos problemas es que en el PECTP la asignación se realiza considerando una pre-inscripción de los alumnos a materias. En cambio, en el CCTP solamente se considera información referida a correlatividades y precedencias curriculares entre las materias.

Existen variantes de los tres enfoques mencionados que fundamentalmente difieren en qué tipo de restricciones consideran o desde qué óptica se busca una solución factible y/o óptima. Bettinelli et al. [2], Kristiansen et al. [11] y Babaei et al. [1] realizan un muy detallado relevamiento de los problemas de diagramación horaria y Lübbecke [15] analiza las dificultades en la implementación práctica de los diferentes enfoques de la literatura.

En su mayoría, la metodología para este tipo de problemas está enmarcada en propuestas metaheurísticas, fundamentalmente debido a que el tamaño de las instancias y la complejidad del problema hacen imposible un abordaje con métodos exactos. Entre los muchos trabajos podemos mencionar a Lewis [13] que usa la técnica de *Simulated Annealing*, Jat et al. [9] proponen un algoritmo genético con búsqueda tabú y Nothegger et al. [17] un algoritmo colonia de hormigas. Cambazard et al. [5] analizan diferentes algoritmos, incluyendo *Constraint Programming*. También pueden encontrarse en la literatura varios trabajos que utilizan modelos de programación lineal entera para desarrollar heurísticas, entre los cuales podemos mencionar a van del Broek et al. [20], Sørensen et al. [18], Kristiansen et al. [10] y Daskalaki et al. [7].

Cabe señalar que la mayoría de las propuestas aplican una fase previa de preprocesamiento en la cual, a partir de la derivación de conflictos, pueden disminuir significativamente la cantidad de variables y/o restricciones del problema, el espacio de búsqueda, etc. Este proceso es clave y de esta manera logran muy buenos resultados computacionales tanto en instancias de la

literatura como en instancias reales.

La perspectiva adoptada por van den Broek et al. [19] para el problema de horarios de Eindhoven University of Technology considera que los horarios de las clases ya están establecidos y que los alumnos tienen una lista de preferencias de materias a partir de la cual se le debe asignar un número dado de materias. El objetivo es lograr una asignación que satisfaga restricciones de recursos (por ejemplo capacidad de aulas), se ajuste a la programación horaria previamente diagramada y maximice la satisfacción del conjunto de alumnos de acuerdo a sus preferencias.

En este trabajo, nos enfocamos en una aplicación proveniente de una problemática real de una universidad privada de Argentina donde el objetivo es diseñar horarios que contemplen simultáneamente los recursos disponibles y las preferencias de los alumnos sobre las materias que desean cursar. El problema comparte muchas características con PECTP y con el trabajo de van den Broek et al. [19], pero la diferencia esencial es que en PECTP las materias en las cuales los alumnos están inscriptos están definidas a priori y en [19] los horarios de las clases están preestablecidos. En nuestro caso, tanto la asignación de materias a alumnos en función de sus preferencias como la asignación horaria son contempladas y definidas en forma conjunta.

Hasta donde llega nuestro conocimiento, no existe en la literatura este enfoque simultáneo. La contribución de nuestro trabajo es abordar un problema práctico que integra y combina problemas de la literatura. Proponemos una metodología para resolverlo basada en programación lineal entera y reportamos resultados en instancias reales. Además analizamos el beneficio obtenido por la universidad con el nuevo proceso aplicado en la diagramación horaria de las materias.

El trabajo está organizado de la siguiente manera. En la sección 2 introducimos el contexto y descripción de las características de la problemática de la universidad. En la sección 3 presentamos un modelo de programación lineal entera que formaliza el problema. En la sección 4 describimos la metodología propuesta para resolverlo y en la sección 5 analizamos la performance del algoritmo en instancias reales. Finalmente, la sección 6 muestra las conclusiones del trabajo y líneas futuras.

## 2. Contexto y características del problema

La universidad ofrece 12 carreras de grado, con aproximadamente 120 materias dictadas por alrededor de 220 docentes para 1200 alumnos. El año académico está dividido en dos semestres cuyos horarios se programan en forma independiente. Cada día de la semana está dividido en bloques horarios (en general de una hora) y el horario semanal se repite durante todo el semestre.

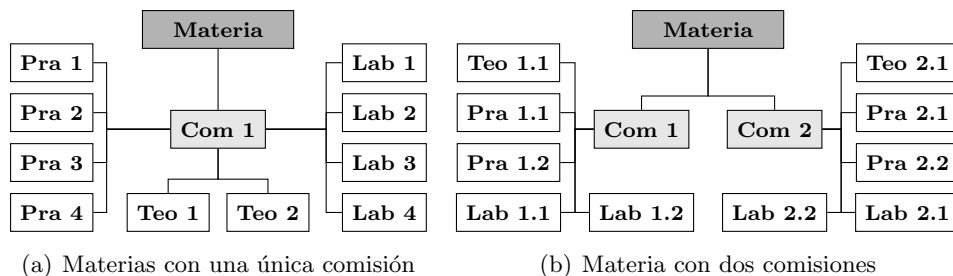
Las materias correspondientes a la currícula de una carrera de grado no tienen mucha relación de correlatividades. Esta característica otorga cierta libertad en la combinación de materias que puede cursar un alumno cada semestre.

Para cada carrera, las materias están clasificadas en mandatorias y electivas. En el primer caso, corresponden a materias que el alumno debe cursar obligatoriamente para obtener su título. En el segundo caso, son materias que están agrupadas por temáticas. De cada grupo temático los alumnos pueden elegir a lo sumo cierta cantidad que está establecida a priori en la currícula de la carrera correspondiente.

Las materias están conformadas por tres tipos de clases: teórica, práctica y laboratorio, combinadas en tres posibles esquemas: teórica, teórica-práctica o teórica-práctica-laboratorio. Cada una de estas clases tiene un patrón horario definido por una frecuencia semanal y una cantidad de bloques horarios continuos por vez. A cada unidad de este patrón lo llamamos *evento*.

Además, las materias están divididas en comisiones. Cada comisión tiene sus propias clases que siguen el mismo esquema. Los alumnos pueden estar inscriptos en una única comisión, no permitiéndose el cruce entre clases de diferentes comisiones y cada clase tiene un cupo determinado por la universidad.

Figura 1: Ejemplo de esquemas



La figura 1 ilustra posibles esquemas. En figura 2(a) la materia consiste de una única comisión y un alumno puede estar inscripto en cualquier combinación de teórica, práctica y laboratorio. La figura 2(b) muestra una materia con dos comisiones. Cada alumno puede estar inscripto en comisión 1 o comisión 2 (excluyente) y dentro de cada comisión, en cualquier combinación de clases que respete el esquema. Por ejemplo Teo. 1.1, Prac. 1.2 y Lab. 1.1 es una combinación factible, pero Teo. 1.1, Prac. 1.1, Lab. 2.2 no lo es.

Previo al comienzo de cada semestre, cada departamento de la universidad decide la oferta de materias y qué docentes estarán a cargo de cada una de sus clases. Junto a esta información, los docentes informan su disponibilidad horaria así como también el máximo de veces por semana que desean dedicar a la enseñanza.

Respecto a la infraestructura, la universidad dispone de 30 aulas con capacidad que varía entre 8 y 110 asientos. Las aulas están clasificadas según sus características en *tipos*, por ejemplo disponibilidad de equipamiento multimedia, instrumental de laboratorio. etc. Dependiendo de los requerimientos, las clases deben ser asignadas a ciertos tipos de aulas.

El proceso habitual de programación horaria seguido por la universidad está bajo la responsabilidad de una única persona quien, usando su conocimiento y datos estadísticos acerca de potenciales alumnos de cada materia, en forma manual procura armar una asignación horaria. Durante un tedioso proceso iterativo, contempla la disponibilidad de recursos y evita conflictos horarios con el objetivo de asegurar una cantidad mínima de materias a cada alumno. Este último punto es particularmente importante debido a que, siendo una facultad privada, los alumnos deben tener la posibilidad de terminar su carrera en tiempo y forma sin verse obligados a afrontar gastos extras.

A partir de una diagramación inicial, si hubiera algún conflicto o algún alumno no pudiera cursar el número requerido de materias, la persona responsable de la asignación, consulta a docentes acerca de la posibilidad de una mayor flexibilidad horaria y repite el proceso hasta lograr una programación horaria que la satisfaga. Una vez logrado el objetivo, los alumnos son informados sobre las materias que le fueron asignadas. Como resultado de este proceso, y fundamentalmente por causa de una asignación compulsiva que no tiene en cuenta los intereses de los alumnos, al comienzo del semestre numerosos alumnos piden al responsable de la programación cambios en la asignación. En algunos casos, las demandas de cambios son simples de satisfacer. En otros, alteran significativamente la programación y el responsable se encuentra ante la disyuntiva de negar el pedido o tener que rehacer el engorroso proceso buscando otras alternativas. En cualquiera de los dos casos se presenta una situación que sería deseable evitar.

Este proceso tiene varios puntos débiles. En primer lugar, es ineficiente e insume mucho tiempo ya que no es humanamente posible considerar todas las alternativas. Por otro lado, el trabajo depende exclusivamente de una persona quien está expuesta a una gran presión para llegar en tiempo y conformar a docentes y alumnos respetando los recursos disponibles. Finalmente, la asignación de materias a los alumnos es compulsiva y no les da posibilidad de elegir cómo planificar su carrera universitaria.

Debido a esta situación compleja, la universidad decide implementar un sistema informático que ayude en el proceso de programación horaria y contacta al grupo de Optimización Combinatoria del Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires del cual formamos parte. Este trabajo de consultoría y desarrollo se enmarcó dentro de las actividades de transferencia de tecnología realizadas por nuestro grupo.

Después de un relevamiento exhaustivo de los diferentes agentes involucrados en la diagramación horaria, nuestra sugerencia fue desarrollar un sistema de programación horaria donde, fundamentalmente, se cambia la óptica sobre la cual se toman decisiones incluyendo las preferencias de los alumnos en el proceso de asignación.

Para esto, el proceso sugerido es que se ofrezca a cada alumno una lista de materias que debe ordenar de acuerdo a sus preferencias. De esa lista se le asignará un número máximo de materias y se le garantiza un número mínimo de asignaciones para evitar retrasos en su carrera. Ambos números deben ser establecidos por la universidad. Por otro lado, la universidad clasifica a los alumnos de acuerdo a su desempeño académico con el objetivo de priorizar las preferencias de aquellos con mayor dedicación y además puede indicar si alguna materia de la lista debe ser asignada obligatoriamente (para contemplar casos en que no cursar cierta materia les cierre oportunidades en el próximo semestre).

El sistema entonces tendrá como objetivo realizar una asignación a materias para cada alumno y una programación horaria factible contemplando los recursos disponibles (aulas, docentes, etc) que maximice la preferencia global de los alumnos ponderada por el desempeño académico.

La propuesta, además de considerar en forma simultánea la asignación a materias y la diagramación horaria, debe contemplar que, a diferencia de muchos de los problemas que aparecen en la literatura, la universidad requiere que las restricciones operacionales y de recursos sean estrictas y no pueden ser violadas. En el caso que no sea factible cumplirlas, el sistema podría ser usado en escenarios menos restrictivos definidos por la universidad, como por ejemplo aumentando la disponibilidad horaria de los docentes.

Como mencionamos anteriormente, el problema así planteado conjuga características del PECTP y del trabajo de van den Broek et al. [19] quien toma en cuenta listas de preferencias pero asume que los horarios ya están fijados. Por otro lado, si la lista de preferencias de los alumnos está compuesta por exactamente la cantidad de materias garantizadas como mínimo, entonces nuestro problema se convierte en PECTP más allá de algunas diferencias en restricciones particulares.

Sin embargo, en nuestro caso la asignación de las materias es parte del proceso de decisión lo cual dificulta el problema. El conocer a priori las materias que cursa un alumno permite realizar una etapa previa de preprocesamiento y deducción de conflictos que pueden ser usados en la definición de la programación horaria. Por ejemplo, clases correspondientes a un mismo alumno no pueden tener intersección horaria.

En nuestro caso, este análisis no es posible debido a que generalmente la lista de preferencias de los alumnos es mayor que la cantidad de materias que debe asignarse. Por lo tanto, los conflictos horarios derivados de la asignación de materias dependen de cuáles efectivamente le sean asignadas y no es algo que pueda deducirse previamente. Además, dada la estructura de las materias, esta misma situación se presenta a nivel comisión.

---

### 3. Modelo de programación lineal entera

---

En esta sección formalizamos el problema a través de un modelo de programación lineal entera. Comenzamos por introducir algunas definiciones y notación agrupadas por incumbencia:

- Días y bloques horarios

$\mathcal{D}$	Conjunto de días de la semana.
$S^d$	Conjunto de bloques horarios del día $d \in \mathcal{D}$ .
$\mathcal{S} = \bigcup_{d \in \mathcal{D}} S^d$	Conjunto de bloques horarios.

- Aulas

$\mathcal{R}$	Conjunto de aulas.
$Types$	Conjunto de tipos de aulas.
$cap^r$	Capacidad del aula $r \in \mathcal{R}$ .
$nr^i$	Cantidad de aulas del tipo $i$ para $i \in Types$ .
$Classes^i$	Clases que pueden ser asignadas a aulas del tipo $i$ .
$t^r$	Tipo del aula $r \in \mathcal{R}$ .

■ Materias

$\mathcal{M}$	Conjunto de materias.
$\mathcal{C}$	Conjunto de clases.
$\mathcal{G}$	Conjunto de grupos de materias electivas.
$Q^m$	Comisiones de la materia $m \in \mathcal{M}$ .
$T^m$	Clases teóricas de la materia $m \in \mathcal{M}$ .
$Cl^m$	Clases de la materia $m \in \mathcal{M}$ .
$Lec^q \subseteq \mathcal{C}$	Clases teóricas de la comisión $q$ .
$Ex^q \subseteq \mathcal{C}$	Clases prácticas de la comisión $q$ .
$Lab^q \subseteq \mathcal{C}$	Clases de laboratorio de la comisión $q$ .
$Slots^c \subseteq \mathcal{S}$	Bloques horarios factibles para la clase $c \in \mathcal{C}$ .
$V^c$	Eventos de la clase $c \in \mathcal{C}$ .
$nd^c$	Periodicidad semanal de la clase $c \in \mathcal{C}$ .
$nv^c$	Cantidad de eventos consecutivos en un mismo día de la clase $c \in \mathcal{C}$ .
$Init^c$	Primer evento de cada bloque de eventos consecutivos de la clase $c \in \mathcal{C}$ .
$R^c \subseteq \mathcal{R}$	Conjunto de aulas factibles para la clase $c \in \mathcal{C}$ .
$q^c$	Cupo de la clase $c \in \mathcal{C}$ .
$nreq^m$	Cantidad de alumnos que deben cursar la materia $m \in \mathcal{M}$ .

■ Alumnos

$\mathcal{A}$	Conjunto de alumnos.
$rk^a$	Desempeño académico del alumno $a \in \mathcal{A}$ .
$\bar{m}^a$	Cantidad máxima de materias a ser asignadas al alumno $a \in \mathcal{A}$ .
$m^a$	Cantidad mínima de materias a ser asignadas al alumno $a \in \mathcal{A}$ .
$M^a \subseteq \mathcal{M}$	Conjunto de materias elegidas por el alumno $a \in \mathcal{A}$ .
$G^a$	Conjunto de grupos de electivas elegidas por el alumno $a \in \mathcal{A}$ .
$w^{am}$	Valor de preferencia para la materia $m \in M^a$ otorgado por el alumno $a \in \mathcal{A}$ .
$\bar{m}^{ag}$	Cantidad máxima de materias del grupo $g \in G^a$ que puede cursar el alumno $a \in \mathcal{A}$ .



- Docentes

$\mathcal{F}$	Conjunto de docentes.
$S^f \subseteq \mathcal{S}$	Disponibilidad de bloques horarios del docente $f \in \mathcal{F}$ .
$C^f \subseteq \mathcal{C}$	Clases a cargo del docente $f \in \mathcal{F}$ .
$nd\bar{a}ys^f$	Cantidad máxima de días por semana de enseñanza del docente $f \in \mathcal{F}$ .

Consideramos los siguientes 6 conjuntos de variables binarias:

- $x_{ac}$  toma valor 1 si el alumno  $a$  es asignado a la clase  $c$ . Esta variable está definida sólo para aquellas clases  $c$  que correspondan a materias que se encuentran en la lista de preferencias del alumno  $a$ .
- $xs_{acs}$  toma valor 1 si la clase  $c$  asignada al alumno  $a$  es programada en el bloque horario  $s$ . Esta variable está definida sólo para aquellas clases  $c$  que correspondan a materias que se encuentran en la lista de preferencias del alumno  $a$  y para bloques horarios  $s$  disponibles del docente a cargo de la clase  $c$ .
- $y_{cvs}$  toma valor 1 si el evento  $v$  de la clase  $c$  es programado en el bloque horario  $s$ . Esta variable está definida sólo para los bloques horarios  $s$  disponibles del docente a cargo de la clase  $c$ .
- $yr_{crs}$  toma valor 1 si la clase  $c$  es asignada al aula  $r$  en el bloque horario  $s$ . Esta variable está definida sólo para los bloques horarios  $s$  disponibles del docente a cargo de la clase  $c$  y para las aulas  $r$  que tengan el tipo requerido por la clase.
- $w_c$  toma valor 1 si la clase  $c$  es programada en el semestre.
- $z_{fd}$  toma valor 1 si el docente  $f$  tiene asignada una clase el día  $d$ . Esta variable está definida sólo para los días  $d$  en que haya bloques horarios disponibles para el docente  $f$ .

Las restricciones que describimos a continuación modelan muchas de las usuales que aparecen en la literatura así como también algunas particulares para nuestro problema.

### 3.1. Asignación alumno-clases

Según los esquemas de las materias, toda materia tiene clases teóricas por lo cual consideramos que un alumno está inscripto en una materia cuando está asignado a una de sus clases teóricas. Teniendo esto en cuenta, las siguientes restricciones modelan la asignación de materias para cada alumno:

$$\sum_{c \in T^m} x_{ac} \leq 1 \quad a \in \mathcal{A}, m \in M^a \quad (1)$$

$$\sum_{c \in Lec^q} x_{ac} = \sum_{c \in Ex^q} x_{ac} \quad a \in \mathcal{A}, m \in M^a, q \in Q^m, Ex^q \neq \emptyset \quad (2)$$

$$\sum_{c \in Lec^q} x_{ac} = \sum_{c \in Lab^q} x_{ac} \quad a \in \mathcal{A}, m \in M^a, q \in Q^m, Lab^q \neq \emptyset \quad (3)$$

$$\underline{m}^a \leq \sum_{\substack{c \in T^m \\ m \in M^a}} x_{ac} \leq \bar{m}^a \quad a \in \mathcal{A} \quad (4)$$

$$\sum_{\substack{c \in T^m \\ m \in g}} x_{ac} \leq \bar{m}^{ag} \quad a \in \mathcal{A}, g \in G^a \quad (5)$$

$$\sum_{a \in \mathcal{A}} x_{ac} \leq q^c w_c \quad c \in \mathcal{C} \quad (6)$$

Las restricciones (1) establecen que cada alumno puede estar asignado a lo sumo a una teórica de una única comisión de cada materia de su lista de preferencias. Si se requiere una asignación obligatoria, entonces la restricción vale por igualdad. Las restricciones (2) y (3) fuerzan la asignación a un esquema factible, es decir que si un alumno está asignado a una teórica de una comisión de una materia, entonces también deberá estar asignado a una clase práctica de la misma comisión y una de laboratorio, respectivamente. Las restricciones (4) aseguran un número mínimo y máximo de materias, y (5) es análogo para materias electivas que pertenecen a un mismo grupo. Finalmente, las restricciones (6) impiden que no se asigne más alumnos que el cupo permitido.

### 3.2. Factibilidad temporal

Las restricciones que describimos a continuación modelan la asignación de bloques horarios a clases impidiendo superposición horaria cuando corresponde.

$$\sum_{s \in Slots^c} y_{cvs} = w_c \quad c \in \mathcal{C}, v \in V^c \quad (7)$$

$$\sum_{s \in S^d} \sum_{v \in Init^c} y_{cvs} \leq 1 \quad d \in \mathcal{D}, c \in \mathcal{C} \quad (8)$$

$$\sum_{s=s_0+1}^{s_0+nv^c-1} \sum_{v=v_i+1}^{v_i+nv^c-1} y_{cvs} - (nv^c - 1)y_{cv_i s_0} = 0 \quad d \in \mathcal{D}, s_0 \in Slots^c,$$

$$s_0 \leq \bar{s}^d - nv^c,$$

$$c \in \mathcal{C}, v_i \in Init^c \quad (9)$$

$$\sum_{s \in S^d} \sum_{v \in Init^c} y_{cvs} + \sum_{s \in S^{d+1}} \sum_{v \in Init^c} y_{cvs} \leq 1 \quad d \in \mathcal{D}, d \neq \bar{d}, c \in \mathcal{C} \quad (10)$$

Las restricciones (7) programan los eventos de una clase en exactamente un bloque horario entre los factibles. Las restricciones (8) aseguran la frecuencia semanal y (9) que los eventos de un mismo día sean consecutivos. Por último, las restricciones (10) garantizan que las clases no se dicten en días consecutivos (para aquellas materias que así lo requieren).

### 3.3. Factibilidad de aulas

La asignación de aulas debe satisfacer restricciones estándares: no exceder la capacidad y no tener superposición horaria. Esto es establecido por las restricciones (12) y (13). La asignación de aula a una clase en un bloque horario es determinada por las restricciones (11) y (14) asegura que no haya cambio de aula dentro de un mismo día.

$$\sum_{r \in R^c} yr_{crs} = \sum_{v \in V^c} y_{cvs} \quad c \in \mathcal{C}, s \in Slots^c \quad (11)$$

$$\sum_{c \in \mathcal{C}, r \in R^c} yr_{crs} \leq 1 \quad r \in \mathcal{R}, s \in \mathcal{S} \quad (12)$$

$$\sum_{a \in \mathcal{A}} xs_{acs} \leq \sum_{r \in R^c} cap^r yr_{crs} \quad c \in \mathcal{C}, s \in Slots^c \quad (13)$$

$$(nv^c - 1)(y_{cv_i s_0} + yr_{crs_0} - 1) \leq \sum_{s=s_0+1}^{s_0+nv^c-1} yr_{crs} \quad d \in \mathcal{D}, s_0 \in Slots^c,$$

$$s_0 \leq \bar{s}^d - nv^c, c \in \mathcal{C},$$

$$v_i \in Init^c, r \in R^c \quad (14)$$

### 3.4. Factibilidad de cursada para alumnos

Cuando a un alumno se le asignan materias, las clases correspondientes deben estar programadas en bloques horarios que no estén superpuestos. Esto garantiza que el alumno pueda asistir a todas las clases.

$$y_{cvs} + x_{ac} - xs_{acs} \leq 1 \quad a \in \mathcal{A}, m \in M^a, c \in Cl^m \\ v \in V^c, s \in Slots^c \quad (15)$$

$$2xs_{acs} - \sum_{v \in V^c} y_{cvs} - x_{ac} \leq 0 \quad a \in \mathcal{A}, m \in M^a, c \in Cl^m \\ s \in Slots^c \quad (16)$$

$$\sum_{\substack{c \in Cl^m \\ m \in M^a}} xs_{acs} \leq 1 \quad a \in \mathcal{A}, s \in \mathcal{S} \quad (17)$$

$$\sum_{c \in Cl^m} \sum_{s \in S^d} xs_{acs} \leq 2 \quad a \in \mathcal{A}, m \in M^a, d \in \mathcal{D} \quad (18)$$

$$\sum_{m \in M^a} \sum_{c \in Cl^m} xs_{acs^d} + \sum_{m \in M^a} \sum_{c \in Cl^m} xs_{acs^{\bar{d}}} \\ - \sum_{\substack{s \in S^d \\ s \neq \bar{s}^d, \bar{s}^d}} \sum_{m \in M^a} \sum_{c \in Cl^m} xs_{acs} \leq 1 \quad a \in \mathcal{A}, d \in \mathcal{D} \quad (19)$$

Las restricciones (15) y (16) relacionan las variables  $x_{ac}$ ,  $y_{cvs}$  y  $xs_{acs}$ . Las restricciones (17) fuerzan que un alumno no tenga superposición horaria y (18) establecen que dentro de un día no tenga más de dos eventos por materia (restricción impuesta por cuestiones pedagógicas por la universidad).

Finalmente, para evitar muchas horas ociosas en la universidad, las restricciones (19) exigen que si un alumno tiene clases el primer y último bloque horario de un día, entonces al menos tenga algún otro bloque horario ocupado el mismo día.

### 3.5. Factibilidad docente

Basado en la información sobre disponibilidad horaria de cada docente y teniendo en cuenta que sus clases no deben solaparse, las restricciones (20) aseguran factibilidad horaria. Además, las restricciones (21) y (22) imponen cotas superiores a la cantidad de días dedicados a la docencia.

$$\sum_{c \in C^f} \sum_{v \in V^c} y_{cvs} \leq 1 \quad f \in \mathcal{F}, s \in S^f \quad (20)$$

$$\sum_{c \in C^f} \sum_{v \in \text{Init}^c} \sum_{s \in S^d} y_{cvs} \leq |S^d| z_{fd} \quad f \in \mathcal{F}, d \in \mathcal{D} \quad (21)$$

$$\sum_{d \in \mathcal{D}} z_{fd} \leq n \bar{d} a y s^f \quad f \in \mathcal{F} \quad (22)$$

### 3.6. Función objetivo

El objetivo a maximizar es la satisfacción global del alumnado. Para esto, para cada alumno  $a \in \mathcal{A}$ , consideramos una suma ponderada entre su preferencia ( $w_c^{ac} = w^{am}$ ) por la materia  $m \in M^a$  y su desempeño académico ( $r k^a$ ) definido por la universidad. Dado que la asignación de un alumno a una materia está definida por su asignación a una de sus clases teóricas, las variables  $x_{ac}$  y parámetros no negativos  $\alpha$  y  $\beta$ , nos permiten modelar una función para maximizar la *calidad* de las asignaciones.

$$\text{(GPECTP-ILP)} \quad \text{máx } z_{ILP} = \sum_{a \in \mathcal{A}} \sum_{\substack{c \in T^m \\ m \in M^a}} (\alpha r k^a + \beta w_c^{ac}) x_{ac} \quad (23)$$

$$\text{s.t.: (1) - (22)}$$

$$x_{ac}, x_{sacs}, y_{cvs}, y_{rcrs}, z_{fd}, w_c \in \{0, 1\}$$

La formulación tiene un gran número de variables y restricciones que hace difícil su resolución usando software de propósito general como CPLEX. Nuestra experiencia computacional nos mostró que CPLEX tiene dificultades tanto en encontrar soluciones factibles como en resolver las relajaciones. Por otro lado, las técnicas de preprocesamiento no logran una reducción significativa del tamaño del modelo. Como mencionamos anteriormente, debido a que la asignación a materias y a horarios se realiza en forma conjunta, hay muy pocas incompatibilidades entre variables que pueda ser deducida a priori, como ocurre en otros problemas de asignación horaria.

---

## 4. Metodología

---

Debido a las dificultades mencionadas, desarrollamos un proceso en dos fases basado en el modelo GPECTP-ILP de programación entera. La idea general del proceso es, en una primera etapa, considerar una relajación GPECTP-RED del modelo que incluye un subconjunto de alumnos y relaja algunas de las restricciones. En una segunda etapa, a partir de la asignación horaria obtenida en la primera etapa, se considera el modelo original GPECTP-ILP fijando las variables correspondientes a asignación de bloque horario a las clases y se busca una asignación de materias a alumnos que maximice la preferencia global.

Debido a que en la primera etapa consideramos una relajación del problema y sólo un subconjunto de alumnos, la segunda etapa puede brindar soluciones con un valor bajo en la función objetivo o, aún peor, resultar infactible. Para aumentar las posibilidades de obtener una buena asignación, en forma aleatoria generamos  $l$  subconjuntos de alumnos  $\bar{A}_1, \dots, \bar{A}_l$  usados en forma independiente para generar distintas programaciones horarias en la primera etapa. Además, cada vez que se resuelve GPECTP-RED para cada subconjunto, generamos más de una posible solución. Esto lo realizamos utilizando la rutina *populate* de CPLEX que permite generar óptimos alternativos o soluciones cercanas al óptimo dentro de un gap determinado. Los detalles del algoritmo utilizado por CPLEX pueden ser consultados en [6].

Para la construcción de los subconjuntos  $\bar{A}_1, \dots, \bar{A}_l$  seguimos la siguiente estrategia. Inicialmente todos los subconjuntos tienen a los alumnos:

1. cuyo su desempeño académico se encuentra dentro del  $K\%$  superior del alumnado
2. que no fueron incluidos en 1 pero que tienen dentro de su lista de preferencia materias de asignación obligatoria. Para ellos, incluimos en su lugar un alumno *ficticio* con sólo las materias obligatorias.

Finalmente, los alumnos no considerados en 1 ó 2 son asignados equitativamente en forma aleatoria a uno de los  $l$  subconjuntos. En caso que el alumno ficticio de 2 pertenezca al mismo subconjunto que el alumno original, se elimina del subconjunto.

En Algoritmo 1 mostramos un bosquejo del proceso. Como parámetros de entrada, además de los datos que definen la instancia, llamamos  $l$  al número de subconjuntos,  $N$  al número máximo de soluciones del proceso *populate*,  $TILIM$  al tiempo máximo de ejecución de la segunda etapa, y  $K$  al porcentaje usado para la construcción de los subconjuntos.

---

**Algoritmo 1** BOSQUEJO DEL ALGORITMO
 

---

**Entrada:** Instancia,  $l$ ,  $N$ ,  $TILIM$ ,  $K$

**Salida:** Conjunto  $\Delta$  de programaciones horarias factibles de GPECTP.

1. Formular el modelo GPECTP-ILP. Computar  $UB$  como el valor óptimo de la relajación lineal.
  2. Construir  $\bar{A}_1, \dots, \bar{A}_l$  de acuerdo a  $K$ .
  3. Sea  $BESTLB = -\infty$  y  $\Delta = \emptyset$  el *pool* de soluciones factibles.
  4. **Para**  $j = 1, \dots, l$  **hacer:**
    5. Formular el modelo ILP de la fase 1 para el subconjunto  $\bar{A}_j$ , resolver y generar  $N$  asignaciones horarias a clases usando la rutina *populate* de CPLEX. Sea  $\Gamma_j$  el *pool* de soluciones.
    6. **Para**  $i \in \Gamma_j$  **hacer:**
      7. Sea  $x^{(i)}$  la  $i$ -ésima solución de la primera fase. Construir el modelo ILP fijando las variables de asignación de horario a clases. Sea  $UB^{(i)}$  el valor objetivo de la relajación.
      8. Si  $UB^{(i)} \leq BESTLB$ , entonces omitir esta solución y retornar a 7 con la solución  $i + 1$  de  $\Gamma$ .
      9. Si no, resolver ILP por a lo sumo  $TILIM$  segundos. Sea  $z^{(i)}$  el valor de la función objetivo de la mejor solución encontrada durante la etapa 2. Si  $z^{(i)} > BESTLB$ , entonces fijar  $BESTLB = z^{(i)}$ , guardar la solución en  $\Delta$  y continuar con la solución  $i + 1$  de  $\Gamma_j$ .
  10. **fin Para**
  11. **fin Para**
  12. Reportar todas las soluciones de  $\Delta$ .
- 

A continuación describimos más detalles de cada etapa.

▪ **Etapas 1**

El propósito de esta etapa es generar programaciones horarias para alimentar la segunda etapa utilizando una relajación del problema, que llamaremos GPECTP-RED, que sea tratable computacionalmente. Con este fin, como ya mencionamos, consideramos un subconjunto de alum-

nos  $\bar{A} \subseteq \mathcal{A}$ . Además descartamos las variables y restricciones de asignación de aulas, reemplazándolas por la restricción descrita a continuación. Si  $nr^i$  es la cantidad de aulas disponibles de tipo  $i \in Types$  y  $Classes^i \subseteq \mathcal{C}$  el conjunto de clases a las que debe asignarse un aula de tipo  $i$ , entonces, para cada bloque horario, restringimos que la cantidad de clases programadas en el bloque no puede exceder  $nr^i$ . Claramente esto es una relajación del problema y no garantiza que pueda realizarse una asignación de aulas, sin embargo reduce sustancialmente el tamaño del modelo y genera soluciones parciales de buena calidad.

Por otro lado, dado que el conjunto de alumnos es reducido, no hay seguridad que todas las clases sean programadas y esto puede influir negativamente en las características de la solución que alimentará a la segunda etapa. Para salvar esta dificultad, consideramos una función objetivo (24) alternativa que combina en forma ponderada la función objetivo del modelo general restringida al conjunto de alumnos  $\bar{A}$  y el número de clases programadas.

$$\begin{aligned}
 \text{(GPECTP-RED)} \quad & \text{máx } \gamma \sum_{c \in T^m} w_c + \delta z_{ILP} & (24) \\
 \text{s.t.:} & (1) - (10), (15) - (22), \\
 & \text{restringido al conjunto } \bar{A} \\
 & \sum_{c \in Classes^i} y_{cvs} \leq nr^i \quad \forall i \in Types, s \in \mathcal{S} \\
 & x_{ac}, x_{sacs}, y_{cvs}, z_{fd}, w_c \in \{0, 1\}
 \end{aligned}$$

## ■ Etapa 2

La segunda etapa está enfocada en obtener una asignación de materias a alumnos y una asignación de aulas a materias a partir de una diagramación horaria obtenida en la etapa 1.

Formalmente, sea  $\pi^* = (x^*, xs^*, y^*, z^*, w^*)$  una solución de la primera etapa que representa una asignación factible de bloques horarios a clases. Entonces, la segunda etapa consiste en formular y resolver el modelo GPECTP-ILP fijando las variables  $y = y^*$ ,  $w = w^*$  y  $z = z^*$ .

Esta segunda etapa tiene dos condiciones posibles de parada. Por un lado, imponemos un tiempo límite de ejecución por cada diagramación de la etapa 1 (*TILIM*) establecido por el usuario. La segunda condición determina que, si la relajación del modelo tiene un valor objetivo inferior a la mejor diagramación obtenida hasta el momento, el proceso para.



Esto evita perder tiempo en la búsqueda de asignaciones que no pueden mejorar la solución actual.

Cabe mencionar que en la etapa 2 podría ocurrir que el modelo resulte no factible debido a que no se pudiera satisfacer la cantidad mínima de asignaciones para cada alumno. En este caso se descarta la asignación de bloques horarios a clases y se procede con la próxima solución obtenida en la etapa 1. Es claro que la elección de  $\bar{A}$  es un punto clave para la calidad de la solución que alimenta la etapa 2. A mayor cantidad de alumnos que conformen el subconjunto  $\bar{A}$ , mejor la calidad de la solución. Sin embargo, esto va acompañado de un mayor incremento en los requerimientos del tiempo computacional de la etapa 1.

---

## 5. Experiencia computacional

---

Con el objetivo de evaluar la efectividad de la propuesta, consideramos 3 instancias reales correspondientes al primer semestre del 2013 (1-2013), segundo semestre del 2014 (2-2014) y primer semestre del 2015 (1-2015). Los experimentos fueron realizados en una workstation Intel(R) Core(TM) i7 CPU (3.40GHz) y 16 Gb de RAM. El algoritmo fue programado en C++ usando CPLEX 12.1 Callable Library como LP y MIP solver. Los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$  y  $\delta$  los fijamos en 1. Cabe señalar que realizamos experimentos con otros valores pero no los incluimos en este reporte ya que el algoritmo no presenta diferencias en su performance. Estos parámetros cumplen la función de ser una herramienta que el usuario tiene para ponderar los diferentes términos de la función objetivo para considerar distintos escenarios.

En la Tabla 1 mostramos un resumen de las características de las instancias consideradas, tales como cantidad de alumnos, promedio de cantidad de materias en la lista de preferencias, cantidad total de materias, etc. En las últimas dos columnas, informamos la cantidad de variables y de restricciones del modelo GPECTP-ILP.

El desempeño académico de cada alumno, así como también la preferencia de materias, es un valor descendente entre 10 y 1 y se consideran 5 días divididos en 5 bloques horarios.

Para los experimentos utilizamos la configuración por defecto que trae CPLEX, imponiendo un tiempo límite de 1000s para cada subconjunto. Para evaluar la calidad de las soluciones encontradas, ejecutamos el algoritmo con  $l = 0, 5, 10, 20$  y  $K = 0\%, 5\%, 10\%$ . La calidad de la solución es medida conforme a la satisfacción global del alumnado con respecto a sus preferencias.

Inst	$ \mathcal{A} $	$\text{av}\bar{m}^a$	$\text{av} M^a $	$ \mathcal{M} $	$\sum_{m \in \mathcal{M}}  Q^m $	$ \mathcal{C} $	$\sum_{c \in \mathcal{C}}  V^c $	$ \mathcal{S} $	$ \mathcal{F} $	$\text{av} S^f $	$ \mathcal{R} $	vars	const
1-2013	981	4.15	7	118	131	341	473	30	220	7.55	31	310063	751086
2-2014	915	4.24	6.88	122	136	335	461	36	211	8.66	31	259395	769712
1-2015	1065	4.12	6.31	111	118	329	450	36	216	9.5	31	303229	804693

Tabla 1: Descripción de las instancias 1-2013, 2-2014 y 1-2015.

Los resultados los presentamos en las tablas 2, 3 y 4. Para cada combinación, informamos la cantidad de soluciones factibles encontradas, ( $|\Delta|$ ), el valor de la función objetivo y la cantidad de asignaciones de la mejor solución encontrada con sus correspondientes *gaps* y tiempo total de cómputo (en segundos). Para el cálculo del *gap* utilizamos el valor de la relajación de GPECTP-ILP, adecuando la función objetivo en cada caso. Si se excede la disponibilidad de memoria para cierta combinación de parámetros, en la tabla aparece *ml*.

Los resultados obtenidos son los esperados. Cuanto más grande sean  $K$  y  $l$ , mejor calidad de resultado. Por ejemplo, los resultados correspondientes al primer semestre del 2013, muestran que para  $l = 5, 10$  se generan subconjuntos  $\bar{A}$  de tamaño grande y GPECTP-RED resulta difícil de resolver, excediendo la capacidad de memoria. Cuando  $l = 0$  (un único conjunto, sólo alumnos con materias obligatorias y los que tengan desempeño académico dentro del  $K\%$  superior del alumnado), la etapa 1 encuentra programaciones horarias para alimentar a la etapa 2. Sin embargo, la calidad de la solución es inferior a la obtenida con  $l = 20$ . Esto era previsible ya que los subconjuntos para  $l = 20$  incluyen más alumnos. Respecto a los valores de  $K$ , también puede notarse que se incrementa la calidad cuando incrementamos el valor  $K$ . Claramente, la mejora en la calidad es a expensas de un mayor tiempo computacional. Sin embargo, dado el contexto de la aplicación, son tiempos de cómputo muy aceptables.

Conclusiones similares pueden hacerse para las instancias del segundo semestre de 2014 y primer semestre de 2015, si bien para estos casos el algoritmo pudo encontrar soluciones para más combinaciones de parámetros. Analizando los datos, notamos que para estas instancias la disponibilidad horaria de los docentes era mayor, facilitando la factibilidad de la programación horaria.

$l/K$	0%						5%					
	$ \Delta $	Obj	Gap	Asig	Gap	Tm	$ \Delta $	Obj	Gap	Asig	Gap	Tm
0	2	50514	6.61	3898	4.01	1680	3	51421	4.95	3929	3.25	1500
5	ml	-	-	-	-	-	ml	-	-	-	-	-
10	ml	-	-	-	-	-	1	52449	3.03	3999	1.53	63902
20	8	52232	3.44	3979	2.01	8580	9	52457	3.02	3997	1.58	67500

Tabla 2: Función objetivo y tiempo computacional para la instancia real 1-2013 para diferentes valores de  $K$  y  $l$ .

$l/K$	0%						5%						10%					
	$ \Delta $	Obj	Gap	Asig	Gap	Tm	$ \Delta $	Obj	Gap	Asig	Gap	Tm	$ \Delta $	Obj	Gap	Asig	Gap	Tm
0	3	46199	9.44	3635	6.25	1245	4	48588	4.75	3750	3.02	1457	5	48657	4.62	3766	2.61	1977
5	ml	-	-	-	-	-	ml	-	-	-	-	-	ml	-	-	-	-	-
10	3	48926	4.09	3773	2.43	19324	ml	-	-	-	-	-	ml	-	-	-	-	-
20	13	49059	3.83	3769	2.53	11694	8	49281	3.4	3792	1.93	35575	8	49356	3.25	3790	1.99	81934

Tabla 3: Función objetivo y tiempo computacional para la instancia real 2-2014 para diferentes valores de  $K$  y  $l$ .

$l/K$	0%						5%						10%					
	$ \Delta $	Obj	Gap	Asig	Gap	Tm	$ \Delta $	Obj	Gap	Asig	Gap	Tm	$ \Delta $	Obj	Gap	Asig	Gap	Tm
0	2	45321	10.84	3935	8.68	9159	3	46391	8.74	4039	6.27	3099	4	46021	9.47	3994	7.31	9661
5	ml	-	-	-	-	-	ml	-	-	-	-	-	ml	-	-	-	-	-
10	3	47040	7.46	4065	5.66	56214	2	46946	7.65	4052	5.96	66772	ml	-	-	-	-	-
20	5	47366	6.82	4104	4.76	77646	11	47482	6.59	4119	4.41	77575	8	47833	5.9	4140	3.92	102042

Tabla 4: Función objetivo y tiempo computacional para la instancia real 1-2015 para diferentes valores de  $K$  y  $l$ .

Con el objetivo de hacer un análisis más detallado del comportamiento del algoritmo, experimentamos con el primer semestre del 2013 con  $l = 0$  y considerando disponibilidad de todos los docentes en todos los bloques horarios.

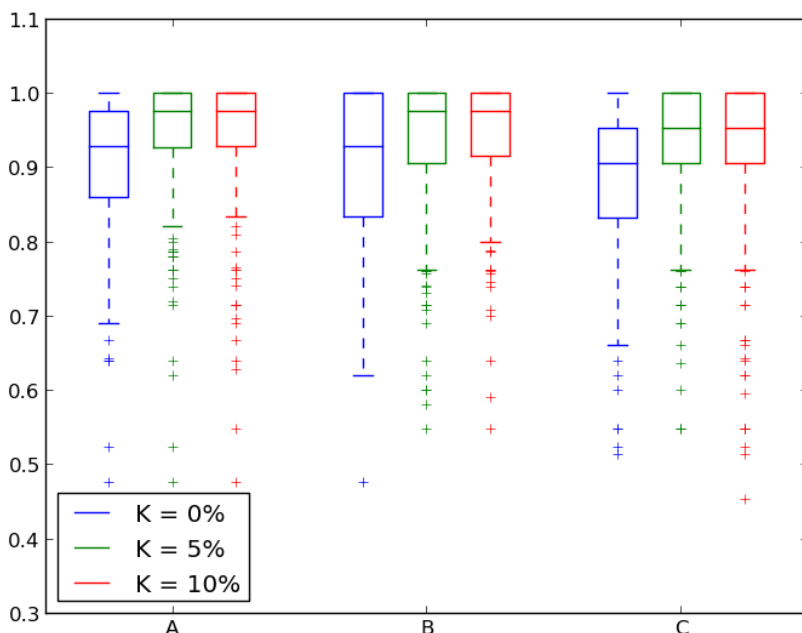
El próximo experimento está enfocado en analizar la satisfacción del alumno. Para esto, para cada alumno  $a \in \mathcal{A}$ , calculamos su valor de satisfacción como

$$sv_a = \sum_{m \in M^a} w^{am} x_{am}. \tag{25}$$

y además, calculamos una cota superior de la satisfacción por alumno considerando el valor de la relajación lineal de GPECTP-ILP con función objetivo (25). El cociente entre estos dos valores lo tomaremos como medida de satisfacción del alumno. En la figura 2 presentamos un diagrama de cajas donde particionamos a los alumnos en tres conjuntos (A,B,C) de igual cardinal ordenados por desempeño académico. Una primera observación es que el nivel de

satisfacción se incrementa cuando  $K$  crece, con mayor valor de la media y menos dispersión. Esto es consistente con los resultados mostrados previamente. Por otro lado, también puede observarse que el nivel de satisfacción es mayor para los alumnos con mejor desempeño académico.

Figura 2: Satisfacción del alumnado según desempeño.



Finalmente, realizamos un último experimento para evaluar el impacto de haber tenido en cuenta las preferencias de alumnos y su desempeño académico. Para esto, generamos un escenario donde la función objetivo considera que todos los alumnos tienen igual desempeño y todas las materias tienen igual preferencia. A posteriori, calculamos cuál es la satisfacción global de estas soluciones considerando las reales preferencias y los desempeños académicos y nos quedamos con la mejor.

En la tabla 5 presentamos los resultados para las 3 instancias y el tiempo de cómputo.

Los resultados muestran que haber considerado las preferencias tiene un impacto positivo en la calidad de las soluciones. En todos los casos, la satisfacción se mejora alrededor de un 10 %, no obstante a un costo computacional mayor pero dentro de niveles razonables para la aplicación.

Inst	$l$	$K$	sin prefs.		con prefs.	
			Obj	Tiempo	Obj	Tiempo
1-2013	20	5	48435	14700	52457	67500
2-2014	20	10	45597	18660	49356	81934
1-2015	20	10	44460	49020	47833	102042

Tabla 5: Análisis de satisfacción.

Por último queremos destacar que la metodología propuesta ha tenido un efecto positivo en la gestión de la universidad. Por un lado, la diagramación horaria que se llevaba a cabo en forma manual e insumía cerca de 2 meses, se redujo a unos pocos días y el responsable de realizar el proceso dispone de una herramienta que le permite analizar diferentes escenarios en poco tiempo. En general, a los 30 minutos de ejecución se obtiene la primera solución y, en promedio, después de 3 horas se cuenta con 10 soluciones. El tiempo total del proceso es de alrededor de 10 horas.

Por otro lado, los alumnos tienen una mayor libertad en elegir las materias y organizar sus estudios teniendo ingerencia en la asignación a partir de sus listas de preferencias. Como mencionamos al comienzo, la asignación se hacía en forma compulsiva y los alumnos reclamaban cambios al conocer las materias que debían cursar. A partir de la implementación del nuevo proceso, las demandas de cambios se redujeron notablemente. Esto influye no sólo en la satisfacción del alumnado sino también en que la universidad puede usar sus recursos en forma más eficiente.

Todo el proceso de relevamiento y definición del problema se llevó a cabo con la cooperación activa del responsable de la diagramación horaria, así como también con el departamento de sistemas de la universidad. El compromiso de las autoridades y personal de la universidad en llevar adelante un cambio tecnológico y de paradigma en la diagramación horaria fue fundamental para el éxito del proyecto.

---

## 6. Conclusiones y líneas de trabajo futuro

---

En este trabajo consideramos un problema de diagramación horaria proveniente de una aplicación real de una universidad privada de Argentina. Hasta donde llega nuestro conocimiento, el proceso propuesto está basado en un enfoque no contemplado en la literatura del área.

Desarrollamos un procedimiento heurístico en dos fases, basados en mode-

los de programación lineal entera que contemplan las restricciones operacionales así como también las preferencias de los alumnos.

La heurística fue analizada en instancias reales, obteniendo soluciones de muy buena calidad, con *gaps* del orden del 3-4% y resultó una herramienta muy útil para la gestión de la universidad.

Como trabajo futuro, además de incorporar algunas otras restricciones como disponibilidad horaria por aula y por alumno, podría ser de interés mejorar el comportamiento por defecto de CPLEX para reducir los tiempos de cómputo. Entre las posibilidades destacamos caracterizar desigualdades válidas e implementar rutinas de separación y heurísticas primales.

**Agradecimientos:** Este trabajo fue parcialmente financiado por la Universidad de Buenos Aires, Programación Científica 2014-2017, UBACyT 20020100100666.

## Referencias

- [1] H. Babaei, J. Karimpour, y A. Hadidi. A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59, 2015.
- [2] A. Bettinelli, V. Cacchiani, R. Roberti, y P. Toth. An overview of curriculum-based course timetabling. *TOP*, 23(2):313–349, 2015.
- [3] E.K. Burke, J. Mareček, A.J. Parkes, y H. Rudová. A branch-and-cut procedure for the udine course timetabling problem. *Annals of Operations Research*, 194(1):71–87, 2012.
- [4] E.K. Burke, N. Pham, R. Qu, y J. Yellen. Linear combinations of heuristics for examination timetabling. *Annals of Operations Research*, 194(1):89–109, 2012.
- [5] H. Cambazard, E. Hebrard, B. O’Sullivan, y A. Papadopoulos. Local search and constraint programming for the post enrolment-based course timetabling problem. *Annals of Operations Research*, 194(1):111–135, 2012.
- [6] E. Danna, M. Fenelon, Z. Gu, y R. Wunderling. Generating multiple solutions for mixed integer programming problems. In *International Conference on Integer Programming and Combinatorial Optimization*, páginas 280–294. Springer, 2007.

- [7] S. Daskalaki y T. Birbas. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1):106–120, 2005.
- [8] C. Gogos, P. Alefragis, y E. Housos. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*, 194(1):203–221, 2012.
- [9] S.N. Jat y S. Yang. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. *Journal of Scheduling*, 14(6):617–637, 2011.
- [10] S. Kristiansen, M. Sørensen, y T.R. Stidsen. Integer programming for the generalized high school timetabling problem. *Journal of Scheduling*, 18(4):377–392, 2015.
- [11] S. Kristiansen y T. R. Stidsen. A comprehensive study of educational timetabling - a survey. Technical report, Department of Management Engineering, Technical University of Denmark, 2013.
- [12] G. Lach y M.E. Lübbecke. Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*, 194(1):255–272, 2012.
- [13] R. Lewis. A time-dependent metaheuristic algorithm for post enrolment-based course timetabling. *Annals of Operations Research*, 194(1):273–289, 2012.
- [14] Z. Lü y J. Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235–244, 2010.
- [15] M.E. Lübbecke et al. Comments on: An overview of curriculum-based course timetabling. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 23(2):359–361, 2015.
- [16] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A.J. Parkes, L. Gaspero, R. Qu, y E.K. Burke. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, 22(1):120–130, 2010.
- [17] C. Nothegger, A. Mayer, A. Chwatal, y G.R. Raidl. Solving the post enrolment course timetabling problem by ant colony optimization. *Annals of Operations Research*, 194(1):325–339, 2012.

- [18] M. Sørensen y F. Dahms. A two-stage decomposition of high school timetabling applied to cases in Denmark. *Computers and Operations Research*, 43:36–49, 2014.
- [19] J. Van den Broek, C. Hurkens, y G. Woeginger. Timetabling problems at the TU Eindhoven. *European Journal of Operational Research*, 196(3):877–885, 2009.
- [20] J.J.J. Van den Broek y C.A.J. Hurkens. An IP-based heuristic for the post enrolment course timetabling problem of the ITC2007. *Annals of Operations Research*, 194(1):439–454, 2012.