ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

MODELO DE PROGRAMACIÓN LINEAL ENTERA PARA LA ASIGNACIÓN DE HORARIOS DE CLASE COMPACTOS EN LA FACULTAD DE CIENCIAS DE LA ESCUELA POLITÉCNICA NACIONAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO MATEMÁTICO

PROYECTO DE INVESTIGACIÓN

MARLON ALEJANDRO QUISAGUANO ACOSTA
imat.marlon.quisaguano@gmail.com

DIRECTOR: LUIS MIGUEL TORRES CARVAJAL
luis.torres@epn.edu.ec

Quito, 12 de abril de 2022

DECLARACIÓN

Yo MARLON ALEJANDRO QUISAGUANO ACOSTA, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

MARLON ALEJANDRO QUISAGUANO ACOSTA

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por MARLON ALEJANDRO QUISAGUANO ACOSTA, bajo mi supervisión.

LUIS MIGUEL TORRES CARVAJAL
Director del Proyecto

Índice general

Índice de tablas

Índice de figuras

Resumen

En el presente trabajo de investigación se exhibe un modelo de programación lineal entera para la asignación de horarios de clase en la Facultad de Ciencias de la Escuela Politécnica Nacional, abordando diferentes particularidades de esta unidad académica, como la existencia de sesiones de clase heterogéneas, la inclusión de disponibilidades y/o preferencias de horario de profesores, restricciones de disponibilidad de aulas, y la incorporación de un criterio para requerir un cierto grado de compacidad en los horarios de clases.

El modelo presentado en este trabajo fue tomado de [27] y modificado ligeramente para permitir la inclusión de sesiones de clase con horarios preestablecidos. El modelo ha sido implementado computacionalmente empleando la API Python del solver Gurobi [15] y ha sido integrado a un sistema de base de datos con interfaz web para facilitar su uso en de la Facultad de Ciencias. Se reportan los resultados de experimentos computacionales realizados sobre instancias obtenidas de la planificación académica de dos semestres en esta facultad.

Abstract

In this research work, an integer linear programming model is formulated for the computation of university course timetables at the Faculty of Science of Escuela Politécnica Nacional, addressing several specific constraints of this academic unit, such as allowing lectures with heterogeneous durations, including preferences and/or availability of lecturers and classrooms, and requiring a certain compactness criterion for class schedules.

The model presented in this work has been taken from [27] and slightly modified to allow the inclusion of letures with pre-established schedules. The model has been implemented using the Python API of the Gurobi solver [15] and has been integrated into a computational web-based tool designed to be used by the administrative staff of the Faculty of Science. The results of computational experiments carried out on instances obtained from the academic planning of two semesters in this faculty are reported.

Capítulo 1

Introducción

Una tarea importante en toda institución educativa es la adecuada administración de sus recursos tanto humanos, como físicos y materiales, de tal forma que se pueda establecer un ambiente donde se garantice una educación idónea y de calidad. En particular, la planificación de horarios y la asignación de aulas para el desarrollo de las actividades docentes es una actividad crítica que sea realiza antes del inicio de cada periodo académico.

Según [8], los planes de estudios prescriben las finalidades, contenidos y acciones necesarias a llevar a cabo por parte de los profesores y estudiantes para desarrollar un determinado currículum. De aquí la importancia de generar horarios que permitan un desarrollo adecuado y eficaz de las distintas actividades curriculares que permitirán a su vez potenciar la calidad de la educación en la institución.

En la Facultad de Ciencias de la Escuela Politécnica Nacional la planificación de horarios se realiza de forma manual, y debido la complejidad de esta tarea, requiere la dedicación de un monto considerable de trabajo por parte del Subdecano y del personal administrativo, está propensa a errores, y produce soluciones que en algunos casos no alcanzan a dar respuesta satisfactoria a todas las variadas expectativas de profesores y estudiantes.

En este contexto, un entorno de clases adecuado, sin conflictos, listo para la enseñanza y aprendizaje entre profesor y estudiantes depende también de una buena planificación académica que cumpla con las expectativas de quienes la conforman, pues como se menciona en [13], "el nivel de satisfacción que una persona experimenta como resultado de su trabajo puede tener un efecto significativo no solo en el individuo, sino también en los que interactúan con él.”

Con estos antecedentes, surge la necesidad de formular e implementar un modelo de optimización para la planificación de horarios de la Facultad de Ciencias, que considere aspectos importantes como que la duración de las sesiones es heterogénea, que existen determinadas sesiones ya preasignadas o con un horario fijo, que la asignación de los horarios se lleve a cabo sin ninguna información sobre la matriculación o preinscripción de estudiantes sino sobre la base de los planes de estudio de la facultad, que busque reducir el número de periodos libres entre clases y, en consecuencia, conseguir un horario más compacto y balanceado, y que sobre todo respete tanto la disponibilidad y/o preferencia de horario de los profesores a lo largo del horizonte de planificación.

Los modelos de planificación de horarios bajo las condiciones particulares existentes en la Facultad de Ciencias han sido abordados en algunos trabajos previos, en [16] proponen un modelo de programación lineal entera y una heurística primal de solución para la planificación de horarios tomando en cuenta la disponibilidad y preferencia de horarios de los profesores. Posteriormente, en [27] se propone un modelo mejorado, incorporando algunas ideas del modelo de planificación de cursos basado en el plan de estudios (CB-CTT, por sus siglas en inglés) reportado en la literatura. Este modelo es resuelto aplicando una heurística de dos fases apoyada en la programación lineal entera. Finalmente, en [18] se estudió una versión simplificada del problema de asignación de horarios y su relación con el problema de coloramiento de grafos, con el objetivo de obtener familias de planos cortantes y desigualdades válidas que pudieran utilizarse para mejorar los métodos de solución existentes. En este proyecto, se propone un nuevo modelo de programación entera para la asignación de horarios de clase de la Facultad de Ciencias de la Escuela Politécnica Nacional que, construyendo sobre el último modelo reportado, incorpore la posibilidad de tener cursos con horarios preasignados, que en la práctica corresponden a las asignaturas de Formación Básica.

De este modo, el presente trabajo esta organizado de la siguiente forma: En el Capítulo 2 se presentan distintos tipos de problemas de calendarización, con especial atención en el problema aquí estudiado, así como también diferentes enfoques y algoritmos de solución. La formulación del modelo de programación lineal entera se presenta y describe en el Capítulo 3. En el Capítulo 4 se presentan las herramientas computacionales utilizadas, se describe cómo fue realizada la implementación computacional del modelo, y se presentan y discuten los resultados obtenidos. Finalmente, en el Capítulo 5 se establecen algunas conclusiones y recomendaciones.

Capítulo 2

Marco Teórico

2.1. Problemas de calendarización

Una clase de problemas ampliamente estudiados dentro de la optimización combinatoria e investigación de operaciones está constituida por problemas de calendarización (*scheduling problems*) debido a su importancia práctica. Generalmente, estos problemas consisten en asignar un conjunto finito de recursos a tareas a lo largo del tiempo y espacio, sujeto a una serie de restricciones y tratando de optimizar una o varias funciones objetivo [3].

Los problemas de calendarización encuentran un gran número de aplicaciones en el mundo real, tales como en las líneas de producción de fábricas, para distribuir las actividades de profesores y alumnos en centros educativos, en la asignación de turnos de pacientes de hospitales, entre otros; es decir, problemas que usualmente tienen su origen en entornos sociales y productivos, en los cuales existe una alta demanda de servicios o productos y los recursos de los que se dispone para satisfacerlos no son abundantes (como personas, espacio o tiempo) [22].

Un tipo particular de problemas de calendarización son los problemas de generación de horarios (*timetabling problems*), que se pueden considerar como el problema de asignar recursos dados a intervalos de tiempos y lugares, con el objetivo de satisfacer un conjunto de restricciones en la mayor medida posible [32]. Las aplicaciones de este tipo de problemas, y de otros tipos de problemas de calendarización, pueden aparecer en distintos contextos, tales como la planificación de actividades dentro de instituciones de salud, transporte, educación e inclusive en instituciones dedicadas a la gestión de empleados.

Por ejemplo, uno de los problemas a los que por lo general se enfrentan empresas dedicadas a servicios y que se preocupan más por los costos en un entorno global, es la asignación del personal de que disponen (*personnel rostering*), que es el proceso de construcción de horarios de trabajo, de modo que la organización pueda satisfacer la demanda de sus bienes o servicios mientras se respetan restricciones técnicas y laborales [12]. Entre las áreas de aplicación de este tipo de problemas, encontramos: centros de llamadas, sistemas de protección y emergencia, servicios financieros, entre otros.

Independientemente del tipo, los problemas de calendarización pueden resultar ser muy desafiantes desde el punto de vista computacional, debido a que en la mayoría de los casos se ha probado que son NP-Difíciles, además que, la cantidad de cálculos necesarios para resolverlos aumentan exponencialmente con el tamaño del problema estudiado, [23] y [2]. Por lo tanto, no es de extrañar que existan múltiples trabajos de investigación sobre este tipo de problemas y que se haya desarrollado una enorme cantidad de algoritmos de solución exactos, de heurísticas y metaheurísticas. El estudio de problemas de calendarización constituye en la actualidad un campo activo dentro de la Investigación de Operaciones, la Optimización Discreta y la Inteligencia Artificial [22].

2.2. Problemas de generación de horarios de clases y exámenes

Como se mencionó en la sección anterior, los problemas de generación de horarios (*timetabling problem*) tienen aplicaciones en muchas áreas, según la función a la que estén destinados o el tipo de institución involucrada en la planificación, además de las limitaciones técnicas y/o organizativas que se consideren. Para este trabajo son de particular interés los problemas de horarios asociados a instituciones educativas como universidades, colegios y escuelas; que son conocidos en la literatura como horarios educativos (*educational timetabling*).

2.2.1. Horarios educativos

Existen distintas formas de definir este tipo de problemas. En general, según Schaerf es posible plantearlo como un problema de búsqueda o como un problema de optimización [24].

La primera opción consiste en encontrar cualquier horario que satisfaga un conjunto dado de restricciones, las mismas que se conocen como *restricciones fuertes*. Es decir, el problema del *horario educativo* puede definirse como la tarea de asignar una serie de conferencias entre profesores y estudiantes (cursos, exámenes, etc) a un número limitado de periodos de tiempo (distribuidos generalmente en el horizonte de tiempo de una semana) y aulas de clase, de tal manera que se satisfagan todas las *restricciones fuertes*.

En el otro caso, además de satisfacer todas las *restricciones fuertes*, la solución obtenida debe optimizar una determinada función objetivo que valora el grado de satisfacción de un conjunto adicional de restricciones que pueden incumplirse o relajarse pero de cuyo cumplimiento depende la calidad de la solución encontrada. A este tipo de restricciones las denominaremos *restricciones débiles*.

Independientemente del caso (búsqueda u optimización), los alcances y los objetivos pretendidos con la optimización de horarios educativos dependen de las necesidades específicas de cada institución.

2.2.2. Tipos de problemas de generación de horarios educativos

En general el problema de construcción de horarios educativos es considerado como un problema polifacético, debido a que se presenta en diversos contextos dentro de la planificación académica. En la literatura podemos encontrar una subdivisión de este tipo de problemas en tres clases: Horarios para Colegios (*School Timetabling*), Horarios para Cursos en Universidades (*University Course Timetabling*) y Horarios de Exámenes (*Examination Timetabling*).

- Horarios para Colegios: Comprende la programación semanal de todas las sesiones de clase contempladas dentro de la planificación académica de una determinada escuela, colegio o unidad educativa.
- Horarios para Cursos en Universidades: Consiste en asignar franjas horarias y

salas adecuadas a cada una de las conferencias semanales, de un conjunto de cursos universitarios, contemplados dentro de la planificación académica de un periodo académico determinado.

- Horarios de Exámenes: Consiste en la programación de horas y aulas para la toma de exámenes, respetando varias restricciones tales como que ningún estudiante esté obligado a rendir más de un examen en un mismo día.

Claramente existen similitudes y diferencias entre estas tres clases, que a más de ser interpretadas como diferentes restricciones (ya sean *fuertes* o *débiles*, según corresponda) se formulan mediante modelos que tienen la misma estructura básica y por lo tanto, cuya resolución puede ser muy parecida.

2.3. Problemas de Horarios para Cursos en Universidades

Independientemente del tipo de problema de planificación de horarios, la solución manual suele requerir varios días de trabajo, sobre todo en instancias grandes que típicamente representan problemas reales. Por este motivo, la programación automatizada de horarios ha despertado el interés de varios investigadores en el área de la optimización combinatoria y la investigación de operaciones.

2.3.1. Modelo básico de búsqueda

Como se mencionó, existen varias formas de abordar el problema de horarios en universidades. Empezamos presentando una versión simplificada del problema, que puede considerarse como un problema de búsqueda básica. Sin embargo, contiene restricciones que lo convierten en un problema difícil de resolver. Esta versión es tomada de [28].

Sea consideran q cursos $C_1, C_2, ..., C_q$, tal que para cada $i \in \{1, ..., q\}$, el curso C_i consta de c_i sesiones. También se consideran r *currículas*, denotadas como $S_1, S_2, ..., S_r$, cada una de las cuales comprende un conjunto de cursos que tienen un grupo de estudiantes en común. Esto significa que para todo $j \in \{1, ..., r\}$, todas las sesiones para todos los cursos de S_j deben ser programadas en distintos periodos de tiempo.

El número de periodos es p, y l_k es el máximo número de sesiones que pueden ser programadas en el periodo k (es decir, el número de aulas de clase disponibles en el periodo k).

El problema consiste en asignar sesiones de clases a periodos de tiempo, de tal forma que: (1) El número de sesiones dictadas de cada curso sea el correcto, (2) para cada periodo de tiempo, no se dicten más sesiones que el número de salas disponibles, (3) no se asignen al mismo período dos sesiones de clase del mismo curso, o de dos cursos pertenecientes a una misma currícula. Empleando variables binarias de asignación, este problema puede ser formulado de la manera siguiente:

Encontrar $\quad \mathrm{y}_{ik}, \quad \forall\, i = 1 \ldots q, \forall\, k = 1 \ldots p.$

s.r.

$$\sum_{k=1}^{p} y_{ik} = c_i, \qquad \forall\, i = 1...q, \tag{1}$$

$$\sum_{i=1}^{q} y_{ik} \leq l_k, \qquad \forall\, k = 1...p, \tag{2}$$

$$\sum_{i \in S_j} y_{ik} \leq 1, \qquad \forall\, j = 1...r, \forall\, k = 1...p, \tag{3}$$

$$y_{ik} \in \{0, 1\}, \qquad \forall\, i = 1...q, \forall\, k = 1...p, \tag{4}$$

donde $y_{ik} = 1$ si una sesión del curso C_i es programada en el periodo k, y $y_{ik} = 0$ caso contrario.

2.3.2. Modelo de optimización

En el momento que se añade una función objetivo a minimizar (o maximizar), el problema se convierte en un problema de optimización. La función objetivo podrá modelizar diferentes aspectos del problema, como la inconveniencia de asignar ciertas sesiones de clases en un determinado periodo, preferencias de horario de los profesores o una medida de la compacidad de los horarios de clase. El el mismo trabajo [28], se hace referencia a un ejemplo de función objetivo para el problema simple de búsqueda:

$$\text{mín} \sum_{i=1}^{q} \sum_{k=1}^{p} d_{ik} y_{ik}$$

donde d_{ik} es la insatisfacción de que se programe una sesión del curso C_i en el periodo k.

Generalmente las formulaciones de los problemas de horarios universitarios que se prueban en la vida real tienen un mayor número de restricciones *fuertes* y *débiles*, debido a que se tratan muchas más características particulares que están asociadas al entorno educativo en el que se aborda el problema.

2.3.3. Problemas basados en preinscripciones y en planes de estudio

Schaerf presenta una breve discusión de algunas variantes del problema de horarios en universidades, en la cuales se abordan temas como la disponibilidad de profesores y salas en determinados periodos de tiempo, la duración de conferencias variable o las preasignaciones de conferencias [24]. También MirHassani & Habibi consideran un gran número de características que pueden ser tomadas en cuenta al momento de realizar la formulación, ya sea como *restricciones fuertes* o *restricciones débiles* [22]. El mismo autor señala que, en el caso de horarios universitarios, al incluir nuevas restricciones al problema no se detecta un cambio notable en la complejidad del mismo, si las restricciones son incluidas como *fuertes* o como *débiles*.

Según la información de la que se dispone previo al proceso de programación de horarios; se distinguen dos tipos de problemas en el contexto de horarios universitarios: Horarios de Cursos Basados en Pre-inscripciones (*Post Enrolment based Course TimeTabaling*) y Horarios de Cursos Basados en el Plan de Estudios (*Curriculum Based Course Timetabling*).

- Horarios de Cursos Basados en Preinscripciones *(Post Enrolment based Course TimeTabling, PE-CTT)*: La programación de horarios se lleva a cabo luego de realizada la matriculación estudiantil, o un proceso de pre-inscripciones en el que los estudiantes indican los cursos que desean tomar.

- Horarios de Cursos Basados en el Plan de Estudios *(Curriculum Based Course TimeTabling, CB-CTT)*: La programación de horarios se lleva a cabo sin ninguna información sobre la matriculación o preinscripción de estudiantes, sino sobre la base de planes de estudio (mallas curriculares) de las carreras. Generalmente, en estos planes se especifican conjuntos de cursos que se espera que los estudiantes tomen en un mismo periodo académico.

Aunque *PE-CTT* parece un enfoque más realista ya que se basa en la información de los cursos en los cuales los estudiantes están realmente inscritos, el enfoque adoptado en *CB-CTT* encaja en la programación de horarios de muchas instituciones de educación superior, entre ellas la Escuela Politécnica Nacional. Esto se debe a la complejidad administrativa adicional de llevar un proceso de pre-inscripciones.

2.4. Modelos de Horarios Cursos basados en el Plan de Estudios

En esta sección describimos el problema *CB-CTT* con más detalle, debido a que el mismo servirá de base para formular el problema de horarios en la Facultad de Ciencias de la Escuela Politécnica Nacional en el próximo capítulo. Consideraremos también los algoritmos de solución más relevantes que se han planteado para este problema.

2.4.1. Notación y Formulación del Problema

El problema *CB-CTT* consiste en la programación semanal de las conferencias de un conjunto de cursos universitarios dentro de un número determinado de aulas y periodos de tiempo, satisfaciendo varias limitaciones debido a conflictos y otras características [24]. A continuación se describe una variante bastante estudiada de este problema [3].

Se consideran los siguientes conjuntos:

Periodos de tiempo: El horizonte de tiempo (típicamente una semana) se divide en días y cada día se divide en un número fijo de *franjas horarias*; así, un *periodo de tiempo* comprende al par (día, franja horaria);

Cursos: Cada curso consiste de un número determinado de *sesiones* de clase dictadas por un profesor (o *conferencista*) y dirigidas a un número estimado de *estudiantes*. Un curso puede pertenecer a varios currículos y sus sesiones deben extenderse a lo largo de un *número mínimo de días laborables*. Para cada curso, es posible especificar un conjunto de *periodos de tiempo no disponibles* los cuales corresponden a los periodos en los que el profesor no está disponible;

Currículo: Un currículo es un conjunto de cursos que se espera que compartan algunos estudiantes en común. La definición de los currículos se realiza en una fase previa a la planificación de los horarios, a partir de la información disponible en los planes de estudios de las carreras;

Salas: Cada sala (o aula) se caracteriza por su *capacidad*, que representa el número de asientos en la sala.

La tarea de CB-CTT consiste en asignar cada sesión de cada curso a un aula y periodo de tiempo, teniendo en cuenta las siguientes *restricciones fuertes*.

H1 *Asignación de sesiones*: todas las sesiones de cada curso deben ser programadas y asignadas a diferentes periodos de tiempo;

H2 *Ocupación de las salas*: cada sala puede albergar como máximo una sesión por periodo de tiempo;

H3 *Conflictos*: las sesiones de cursos pertenecientes a un mismo currículo o impartidos por el mismo profesor no se puede programar en el mismo periodo de tiempo;

H4 *Disponibilidades*: los periodos de tiempo en los que un determinado profesor no se encuentre disponible, no pueden ser utilizados para programar sesiones asociadas a cursos impartidos por ese profesor.

El objetivo es minimizar la suma ponderada (de acuerdo con ciertos pesos dados) de múltiples objetivos, los cuales representan penalizaciones por la violación de las siguientes *restricciones débiles*:

S1: *Número mínimo de días laborables*: para cada curso, se aplica una penalización para cada día por debajo del número mínimo de días laborables previstos para el dictado de clases del curso;

S2: *Compacidad del plan de estudios*: es preferible que las sesiones pertenecientes a un mismo currículo se dicten de manera consecutiva, sin ningún periodo de tiempo vacío en el medio (horas huecas); por lo tanto, se da una penalización por cada *sesión aislada*, es decir, una sesión no adyacente a ninguna otra conferencia del mismo currículo en el mismo día;

S3 *Capacidad de la sala*: se aplica una penalización por cada estudiante que no pueda tener un asiento en el aula asignada a la conferencia del curso;

S4 *Estabilidad del aula*: es preferible que un curso se imparta siempre en la misma aula o sala, por lo que se aplica una penalización por cada aula adicional utilizada para un mismo curso.

La variante del CB-CTT descrita aquí fue adoptada en la Segunda Competencia Internacional de Horarios (ITC-2007). Esta formulación se diseñó con el objetivo de ser simple y lo suficientemente general como para atraer la atención de muchos investigadores [11]. Numerosos estudios prácticos y artículos académicos publicados en la última década se han enfocado en el estudio de esta variante. Bonutti et. al. describen otras variantes adicionales. Los autores inician por dar una definición general del problema, y continuan definiendo un conjunto de componentes (características) de costos básicos y de costos opcionales. Luego, las variantes surgen al clasificar cada una de estas componentes como *restricciones fuertes* o como *restricciones débiles*, para el problema de optimización. Finalmente, se especifica un costo de penalización adecuado y diferente para cada componente clasificada como *restricción débil*.

2.4.2. Algoritmos de solución

Los algoritmos pueden considerarse como una secuencia finita de pasos o instrucciones para realizar una tarea. Los algoritmos de solución para un problema pueden ser distintos y variados, tanto simples como complejos, dependiendo del enfoque o técnica que se utilice. Por ejemplo, un algoritmo puede emplear técnicas que se basan en la simulación de la forma humana (manual) de resolver un problema [25]; o puede incorporar ideas más avanzadas que surgen del estudio teórico del mismo, desde enfoques de investigación como la optimización matemática o la inteligencia artificial.

Los algoritmos de solución pueden ser clasificados dependiendo de sus características. Por ejemplo, según el tipo de solución que entregan pueden ser catalogados como:

- *Algoritmos exactos*: Son aquellos que arrojan siempre una solución óptima del problema, o al menos información exacta sobre la brecha de optimalidad de la solución encontrada. Para instancias grandes de problemas computacionalmente difíciles, estos algoritmos pueden requerir un alto costo computacional.

- *Algoritmos aproximados*: Son algoritmos eficientes desde el punto de vista computacional, cuyas soluciones tienen una brecha de optimalidad acotada por un análisis teórico. Generalmente son considerados en lugar de los *algoritmos exactos* cuando la ejecución de estos últimos es muy costosa.

- *Algoritmos heurísticos*: Son algoritmos computacionalmente eficientes para los cuales no hay ninguna garantía teórica de la brecha de optimalidad de la solución encontrada, aunque generalmente son diseñados con la idea de encontrar soluciones razonablemente buenas en la práctica. Las heurísticas suelen emplearse con frecuencia como métodos de solución para problemas computacionalmente difíciles cuando los algoritmos exactos resultan inaplicables. En muchos casos, las heurísticas replican las ideas que se emplean en la solución manual de un problema.

A continuación se describen varias técnicas de solución empleadas para el problema de *Horarios de Cursos para Universidades*, en general, con énfasis en los algoritmos propuestos para el problema de *Horarios de Cursos Basados en Plan de Estudios* (CB-CTT).

Heurísticas directas

La gran mayoría de métodos heurísticos se basan en replicar y adaptar estrategias de solución que son observadas en la naturaleza por lo que existe una gran diversidad. Entre los métodos más populares se encuentran las heurísticas directas, de descomposición, de reducción, constructivas y de inducción. Una heurística directa comúnmente utilizada para la construcción de horarios consiste en programar sucesivamente cada una de las sesiones de clase, asignándolas a períodos y aulas donde no existan conflictos. Es decir, se amplía un horario parcialmente completo conferencia por conferencia hasta que se hayan programado todas las conferencias [24].

Las heurísticas directas han sido empleadas para la construcción de horarios desde los primeros trabajos que abordaron este problema [25]. Carrasco presenta un algoritmo basado en una heurística directa mediante un *OperadorAula* que elige un curso no asignado de manera aleatoria y determina las distintas posibilidades que el mismo tiene para ser programado en función de las aulas permitidas para el curso. Para realizar la asignación, se calculan las brechas de tiempo disponibles al mover cursos ya asignados a otros horarios [7].

Reducción a la coloración de grafos

El problema de coloración de grafos consiste en asignar un color a cada uno de los nodos de un grafo, de tal forma que nodos adyacentes tengan siempre colores distintos, y que el número total de colores empleados sea el menor posible [29].

En la literatura asociada a los problemas de horarios educativos, la reducción del problema a uno de *coloración de grafos* es considerada ampliamente como un enfoque de solución. Esto se debe, en primer lugar, a que el problema de *coloración de grafos* es un problema bien conocido y estudiado; y a que los problemas de horarios educativos lo tienen al problema del coloramiento de grafos como un subproblema central.

En [3] se describe el problema *CB-CTT* como un problema de *coloración de grafos*. Se considera un grafo no dirigido que tiene un nodo para cada conferencia, una arista para cada par de conferencias que no deben ser programadas simultáneamente (es decir, conferencias que pertenecen a un mismo curso, o a cursos en una misma currícula, o que son impartidas por el mismo profesor) y también se considera un color para cada periodo de tiempo. Entonces, el problema se reduce a asignar un color a cada nodo de manera que a los nodos adyacentes se les asignen colores diferentes. Notar que de esta manera se incorporan las restricciones *H1* y *H3*. Además para modelar la restricción *H2*, se impone una restricción sobre el número máximo de veces que se puede usar cada color, correspondiente al número de aulas de clase disponibles de cada tipo. Finalmente se agrega una restricción para cada nodo para prohibir el uso de determinados colores, correspondiente a la restricción *H4* (disponibilidad de profesores).

En [24] se hace una descripción de este enfoque para los tres tipos de problemas de *horarios educativos* aquí revisados, por otro lado en [22] se hace referencia al problema de *coloración de grafos* como una herramienta para resolver subproblemas dentro de distintos métodos de solución descritos.

Algoritmos de búsqueda local

El término *búsqueda local* designa un conjunto de algoritmos empleados para resolver problemas de optimización. Básicamente estos algoritmos empiezan por elegir una *solución inicial* dentro del *espacio de búsqueda*. Luego de manera iterativa, a partir de esta solución se establece un *subespacio de búsqueda* o *vecindad* que se explora en búsqueda de una solución que mejore el valor de la *función objetivo*. Si esto sucede, la solución actual se reemplaza por la nueva solución encontrada y el proceso se repite. Finalmente, cuando se consigue satisfacer un determinado criterio de parada el algoritmo se detiene.

Dentro de los problemas de *asignación de horarios universitarios* la noción de *búsqueda local* también ha sido utilizada, sin embargo, se debe redefinir el concepto *solución factible* para garantizar la conectividad del espacio de búsqueda y de las vecindades. Así, el concepto de *solución factible* u *horario factible* se relaja permitiendo que el algoritmo pase por soluciones que violan restricciones fuertes y así escapar de óptimos locales. Sin embargo, esto puede conducir a que el algoritmo caiga fácilmente en un bucle.

Búsqueda Tabú. Utiliza la noción de *búsqueda local* incorporando una memoria en forma de un lista de soluciones llamada *lista tabú* con el fin de prohibirlas y así evitar bucles en el proceso de búsqueda. Esto se complementa mediante un mecanismo denominado *criterio de aspiración* [24] que evita que una solución esté en la lista si su evaluación en la denominada *función de aspiración* es mejor que la actual.

Con relación a los principales parámetros de control, White & Xie estudiaron la influencia del tamaño de la *lista tabú* en la generación de *horarios educativos* en [30] y [31], concluyendo en ambos casos que listas de mayor tamaño generaban resultados de mayor calidad. Di Gaspero en [10] y Adbullah & Turabieh en [1], investigaron el comportamiento del algoritmo bajo la incorporación de estructuras multi-vecindad obteniendo mejoras en sus resultados.

En la aplicación, Hertz en [17], considera la aplicación de algoritmos de *búsqueda tabú* para abordar problemas de *horarios universitarios* incorporando también conferencias de duración heterogénea. Más aún, Zhipeng & Jin-Kao en [21], proponen un algoritmo para el problema *CB-CTT* en el que se integran y adaptan varias características, como una estructura de cadenas de *Kempe*, un operador de perturbación de solución y un mecanismo de búsqueda adaptativo mostrando una alta efectividad. Y más recientemente Bonutti et. al. [4] describen los resultados computacionales obtenidos de aplicar un algoritmo de *búsqueda dinámico* sobre el problema *CB-CTT* y sobre las instancias proporcionadas para la competición *ITC-2* en 2007-2008.

Recocido Simulado. Basado también en la noción de *búsqueda local*, viene motivado por el proceso de purificación de metales llamado recocido, en el cual, la temperatura del metal es elevada hasta el punto en el que pueda aceptar varios cambios en sus propiedades o estructura. Entonces, el proceso continúa dejándolo enfriar lentamente provocando que los cambios aceptados por el material sean cada vez menos hasta llegar a su estado fundamental en el cual no acepta más alteraciones en su estructura.

El algoritmo comienza eligiendo una solución inicial a partir de la cual un proceso iterativo genera soluciones vecinas. El algoritmo acepta nuevas soluciones siempre que sean mejores a la actual y si son peores se aceptarán con probabilidad $e^{-\frac{d}{t}}$, para permitir que el algoritmo escape de óptimos locales. El parámetro d es la diferencia entre el valor objetivo de la nueva solución y la actual, y t es denominado *parámetro de temperatura* que se inicializa en un valor relativamente alto y va descendiendo con cada iteración, de manera lenta y de acuerdo a una estrategia de enfriamiento establecida. Finalmente, el algoritmo termina cuando el *parámetro de temperatura* alcanza un determinado límite inferior.

Con relación a los parámetros de control, se ha determinado que la velocidad de la estrategia de enfriamiento influye directamente con la calidad de la solución en el contexto de *horarios educativos* [26]. Grigorios et. al. [14] y Kustoch [19], adoptaron también este enfoque para abordar el problema *CB-CTT*.

Programación Lineal Entera (PLE)

Iniciaremos indicando que los modelos formulados en las secciones 2.3.1 y 2.3.2, son ejemplos de modelos de *programación lineal entera*. La *programación lineal entera* es una técnica ampliamente usada en la investigación de operaciones que consiste en formular un problema de optimización combinatoria como el problema de maximizar o minimizar una función lineal de variables enteras, sujeto a restricciones que se expresan como ecuaciones o desigualdades lineales.

Burke et. al. presentaron una formulación monolítica que aborda la variante del problema *CB-CTT* descrita en la Sección 2.4.1 como un modelo de optimización, cuya función objetivo busca minimizar las penalizaciones por incumplir las restricciones débiles e incorporando 16 familias de restricciones para modelar las restricciones fuertes [5]. Esta formulación utiliza 5 familias de variables en total, 4 de las cuales se utilizan para modelar las restricciones débiles que, sin mencionar el número de familias de restricciones, hacen que su complejidad computacional sea alta.

En estudios posteriores, Bruke et. al [6] presentan nuevas ideas que ayudan a reducir el número de variables y restricciones. De aquí se puede rescatar que, al ignorar las penalizaciones por incumplir la capacidad de la sala y su estabilidad (restricciones *S2* y *S4* de la sección 2.4.1) se reduce el tiempo de cálculo del modelo. Este resultado es muy importante para el trabajo que desarrollaremos aquí y que detallaremos en las siguientes secciones. Por otro, lado Daskalaki & Birbas presentan un procedimiento de relajación en dos etapas que resuelve eficientemente una formulación de *PLE* propuesta por los autores [9]. En comparación con los enfoques de una sola etapa, el tiempo de cálculo se reduce significativamente sin pérdida de calidad en los horarios resultantes. De igual manera, Lach & Lübbecke [20] describen un método de *PLE* de dos etapas en el cual durante la primera etapa se asignan las conferencias a los períodos de tiempo y en la segunda etapa continúa a partir de dicha asignación para asignar las conferencias programadas en cada período de tiempo a las aulas de clase.

Algoritmos Heurísticos basados en PLE

En los últimos años, el desarrollo de nuevas tecnologías en los sistemas de información, la disponibilidad de programas confiables y la capacidad de resolver problemas relativamente grandes en un tiempo relativamente corto son las principales razones para que los enfoques basados en formulaciones de *PLE* hayan recibido renovada atención en la solución de problemas de horarios educativos de gran escala [22].

Algunas heurísticas emplean información obtenida a partir de la solución de programas lineales y programas lineales enteros. Por ejemplo, una heurística puede consistir en redondear la solución fraccionaria óptima correspondiente a la relajación lineal de un *PLE* hasta obtener una solución factible. Otras heurísticas pueden resolver en una primera fase una relajación de un *PLE* y emplear la información obtenida para fijar ciertas variables del *PLE* original en una fase posterior. En [3] se describen varios algoritmos que emplean esta estrategia, como el propuesto por Lach y Lübbecke [20], que consta de dos fases y fue clasificado dentro de los primeros cinco lugares en la competencia *ITC-2007*.

Dentro del desarrollo de modelos matemáticos para la planificación de un horario de clase para la Facultad de Ciencias de la Escuela Politécnica Nacional, asociados a algoritmos heurísticos basados en PLE, Heredia propone un modelo de programación lineal entera y una heurística primal que además toma en cuenta la disponibilidad y preferencia de horarios de profesores [16]. Posteriormente, Torres & Torres proponen un modelo mejorado de planificación basado en el plan de estudios que corresponde a un variante del problema *CB-CTT* revisado. Para su solución, emplean una heurística de dos fases que consiste en resolver primero un modelo simplificado que no incluye restricciones débiles de compacidad, y utilizar la solución obtenida como solución inicial en un método de branch-and-cut para el modelo completo [27].

A lo largo de esta sección se han descrito algunas estrategias de solución para problemas de horarios de clase basadas en la reducción del número de variables y restricciones con el objetivo de reducir tiempo de cálculo del modelo. En este contexto, surge la idea de reducir aún más el número de variables al eliminar aquellas asociadas a cada profesor y periodo en el que no están disponibles o aquellas asociadas a sesiones que no pueden iniciar en determinado periodo por su duración. En el siguiente capítulo, se describe un modelo de *PLE* propuesto en [27] que recoge las características particulares del problema de horarios en la Facultad de Ciencias de la Escuela Politécnica Nacional, y que ha sido implementado en el presente trabajo, usando el API Python del solver Gurobi.

Capítulo 3

Formulación del Modelo

La Facultad de Ciencias, referente en la investigación, docencia y aplicación de las ciencias físicas, matemáticas y económicas a nivel nacional, actualmente cuenta aproximadamente con 721 estudiantes y 77 docentes en labor.

La facultad oferta cuatro carreras académicas de tercer nivel: Física, Matemática, Ingeniería Matemática y Economía. Las respectivas mallas curriculares contienen materias o asignaturas de Formación Básica y Profesional. Al planificar los horarios de clase, los cursos asociados a las asignaturas de Formación Básica son los primeros en ser asignados, proceso que en la Escuela Politécnica Nacional ocurre de manera centralizada. Como parte de la planificación académica semestral, en la Facultad de Ciencias se debe asignar un horario para el resto de cursos que se contemplan en el plan de estudios.

En este capítulo abordaremos un modelo para la generación automatizada de horarios de clases en la Facultad de Ciencias, desarrollado en concordancia con los modelos de **Horarios de Cursos Basados en el Plan de Estudios** (CB-CTT, por sus siglas en inglés). La asignación se realiza sobre el análisis de conflictos de los conjuntos de sesiones de cursos que se espera que los estudiantes tomen según la estructura de la malla curricular, es decir, sin información previa acerca de preinscripciones. A continuación se presentan algunas definiciones previas requeridas para la formulación del modelo. Este modelo fue propuesto en [27].

La asignación del horario de clases se realiza teniendo en cuenta un *horizonte de planificación* que, generalmente, es de 5 o 6 días en la semana. Cada día del horizonte de planificación consta de un conjunto de *períodos* de duración de una hora.

En un *curso* se dicta una asignatura específica que se imparte por un profesor, en un determinado número de sesiones de clase, a un número estimado de estudiantes. El nombre de un curso está formado por el nombre de la asignatura, las iniciales de la carrera que pertenece y el nombre del grupo de estudiantes a quienes se dicta. Por ejemplo, "Álgebra Lineal M,F - A1" y "Álgebra Lineal M,F- A2" son dos cursos en los cuales se dicta la asignatura Álgebra Lineal a los grupos A1 y A2 que están formados por estudiantes de Matemática y Física.

Cada profesor puede especificar su *disponibilidad* y/o *preferencias* de horario dentro del horizonte de planificación. Una *sesión* o conferencia de clase es una reunión en la que se dicta determinada asignatura para un curso en específico. Por ejemplo, el curso de "Álgebra Lineal M,F - A1"puede ser impartido en 3 sesiones semanales, dos sesiones con una duración de 2 períodos y una sesión con una duración de 3 períodos. El número de períodos total de cada asignatura o materia se encuentra en el establecido en cada pénsum de cada carrera. Un *pensúm* asigna a cada materia un número de *créditos* o períodos de una hora, un nivel, pre- y correquisitos y una unidad curricular (Formación Básica o Profesional). Por ejemplo, Álgebra I puede ser tomada únicamente luego de haber aprobado la asignatura de Álgebra Lineal, además Álgebra lineal debe ser tomada simultáneamente con con la componente práctica de Álgebra Lineal de Formación Básica.

Un conjunto de cursos que se espera sean atendidos por un mismo grupo de estudiantes representa una *currícula*. Las currículas se determinan de manera previa al cálculo del horario sobre la base del análisis de la malla curricular, de la oferta académica y de la situación actual de los estudiantes. Por ejemplo, los cursos correspondientes a materias de un mismo nivel forman usualmente una currícula.

La planificación académica se realiza de forma semestral, y consta de las siguientes actividades:

1. Se definen las asignaturas o materias a ser ofertadas en el semestre en planificación;
2. Se estima el número de estudiantes esperado para cada asignatura;
3. Se establece el número de cursos a ofertarse por cada asignatura;
4. Se define para cada curso el número de sesiones, la duración de cada sesión y el tipo de aula (pequeña, grande, laboratorio) requerido;

5. Se asignan profesores a los diferentes cursos, proceso que se conoce como "asignación académica";
6. Se programan los horarios de clase;
7. Los estudiantes se inscriben en los cursos.

Dentro de estas actividades, la asignación del horario de clases aborda el sexto paso y consiste en asignar un conjunto de sesiones de cursos a periodos y aulas teniendo en cuenta la existencia de sesiones prefijadas, correspondientes a las materias de Formación Básica.

El horario a calcular debe satisfacer las siguientes restricciones fuertes:

- H1 : Cada sesión o conferencia de duración d debe ser programada en un intervalo de d periodos consecutivos en el mismo día;
- H2 : Cada aula puede albergar máximo una sesión por período, si y sólo si el aula está disponible en ese periodo;
- H3 : Las sesiones impartidas por un mismo profesor o pertenecientes a una misma currícula deben ser programadas en intervalos de periodos disjuntos;
- H4 : Una sesión de un determinado curso no puede ser programada en un periodo en el cual el profesor asignado no esté disponible;
- H5 : Cada sesión de un determinado curso debe ser programada en un aula compatible, es decir, un aula con la capacidad suficiente y del tipo adecuado;
- H6 : (Excepto para las asignaturas de Formación Básica) Por cada curso, máximo se puede impartir una sesión al día, es decir, el número de días empleados para cada curso debe ser igual al número de sesiones del curso.

Por otra parte, es deseable que el horario calculado satisfaga las dos siguientes restricciones débiles:

- S1 : Para cada currícula, la máxima longitud de un día de trabajo no debe exceder un límite prescrito; se impone una penalización por cada periodo que exceda este límite.

S2 : Los cursos de los profesores deberían ser programados solo en periodos donde estos hayan indicado su preferencia para dictar clases; de otro modo, se imponen penalizaciones por enseñar en periodos con menor valor de preferencia.

La violación de las restricciones débiles será representada en el modelo de programación lineal entera mediante una función objetivo de penalización, cuyo valor se desea minimizar.

3.1. Conjuntos y Parámetros

A continuación, se definen y describen los conjuntos y parámetros necesarios para la formulación del modelo propuesto en [27], y que en este trabajo ha sido levemente modificado para incluir horarios prefijados para las materias de Formación Básica.

C : conjunto de cursos;

C* : conjunto de cursos pertenecientes a Formación Básica, $C^* \subset C$;

$\mathbf{S}_c$: conjunto de sesiones (sesiones) del curso $c \in C$;

S : conjunto de sesiones, $S := \bigcup_{c \in C} S_c$;

G : conjunto de currículas;

T : conjunto de profesores;

$\mathbf{C}_g$: conjunto de cursos pertenecientes a la currícula $g \in G$;

$\mathbf{C}_t$: conjunto de cursos dictados por el profesor $t \in T$;

D : conjunto de días en el horizonte de planificación;

H : conjunto de períodos de una hora;

$\mathbf{H}_d$: conjunto de períodos en el día $d \in D$;

$\mathbf{H}_s$: conjunto de períodos en los que es posible iniciar la sesión $s \in S$;

R : conjunto de aulas;

$\mathbf{R}_s$: conjunto de aulas compatibles con la sesión $s \in S$;

$\mathbf{R}_c$: conjunto de aulas compatibles con el curso $c \in C$, $R_c := \bigcup_{s \in S_c} R_s$.

Cada sesión del curso $s \in S$ tiene una duración δ_s, que indica el número de períodos consecutivos requeridos para que esta se lleve a cabo. Observe que esto es posible solamente en ciertos períodos de tiempo, debido a la disponibilidad del profesor a cargo, como también a su duración. La duración de un curso c está definida como la duración total de sus sesiones, i.e. $\delta_c := \sum_{s \in S_c} \delta_s$, y también se define la duración máxima de todas las sesiones como $\delta_{max} := \max_{s \in S} \delta_S$.

Cada aula $r \in R$ se caracteriza por las siguientes propiedades: (i) Su capacidad, dada por el número de asientos disponibles en r para ser ocupados por los estudiantes; (ii) su disponibilidad, que consiste en el subconjunto de períodos de $h \in H$ en los cuales r puede ser utilizada; y (iii) su tipo, que indica para qué sesiones $s \in S$ la infraestructura disponible en r es adecuada. Por ejemplo, el tipo de un aula puede ser "salón de clase", "aula magna", "laboratorio de computación", etc.

Se dice que un aula r es compatible con la sesión s, si la capacidad de r es mayor o igual al número de estudiantes inscritos en el curso $c \in C$ para el cual $s \in S_c$ y si r es del tipo de aula requerida por s. R_s contiene todas las aulas compatibles con la sesión $s \in S$.

La disponibilidad de las aulas se expresa mediante un parámetro binario ψ_{hr}, el cual es igual a 1 si y sólo si r está disponible en el período de tiempo $h \in H$, y a 0 caso contrario.

La hora de inicio de un período $h \in H$ se denota por un parámetro entero a_h. Dada una currícula $g \in G$, un día $d \in D$ se dice que es una *jornada laboral* para g si hay al menos una sesión de un curso $c \in C_g$ programada en ese día. La longitud de una *jornada laboral* se define como el tiempo transcurrido (expresado en horas) desde el inicio de la primera sesión hasta el final de la última sesión de los cursos de C_g programados para ese día, i.e., si la primera sesión es programada para empezar en el período h_1 y si la última sesión es programada para finalizar en el período h_2, entonces la duración de la *jornada laboral* viene dada por $a_{h_2} + 1 - a_{h_1}$. Para cada currícula, la longitud de una *jornada laboral* no debe exceder una longitud máxima prescrita L^{max}; de otro modo, se incurre en una penalización de W_L para cada periodo en exceso.

Cada profesor $t \in T$ está disponible para dictar clases solo en ciertos períodos de tiempo que conforman un conjunto $H_t \subset H$. Además, el profesor puede especificar preferencias para impartir clases dentro de los períodos de tiempo en H_t, i.e., para cada $h \in H_t$ se define un valor de preferencia $w_{th} \in \Omega$, con $\Omega \subset \mathbb{Z}^+ \cup \{0\}$. Un valor bajo de w_{th} indica una alta preferencia para dictar clases en el periodo h, mientras que los valores más altos indican menores niveles de preferencia por parte del profesor.

Para cada sesión $s \in S$ de un curso $c \in C_t$ dictado por el profesor $t \in T$, y para cada período $h \in H_s$, definimos:

$$w_{sh} := \sum_{j=0}^{\delta_s - 1} w_{t,h+j},$$

que es igual al costo total o penalización incurrida por violar las preferencias de horario del profesor t si s es programada para empezar en h.

Definimos adicionalmente los siguientes conjuntos auxiliares: $S_r := \{s \in S : r \in R_s\}$ es el conjunto de sesiones que pueden ser dictadas en un aula $r \in R$; $S_{ch} := \{s \in S_c : h \in H_s\}$ el conjunto de sesiones del curso c que pueden empezar en el período $h \in H$; y $K_{sh} := \{k \in H_s : 0 \leq h - k \leq \delta_s - 1\}$ que es el conjunto de períodos en los cuales $s \in S$ puede haber empezado dado que s está teniendo lugar en el período h.

3.2. Variables de decisión

Emplearemos los siguientes cuatro conjuntos de variables de decisión en el modelo:

$\mathbf{y}_{shr}$ son *variables binarias* que indican el período y aula en los que una sesión es programada, con $y_{shr} = 1$ si y sólo si la sesión $s \in S$ es programada para comenzar en el período $h \in H_s$ y en el aula $r \in R_s$;

$\mathbf{t}^1_{gd}, t^2_{gd}$ son *variables continuas no negativas*, que representan para cada currícula $g \in G$ y para cada *jornada laboral* $d \in D$ de dicha currícula, la hora de inicio de la primera sesión y la hora de finalización de la última sesión de los cursos de C_g en el día d, respectivamente. Si no se dictan clases de la currícula g en el día d, entonces estas variables toman valores que no afectan al valor de la solución;

L_g son *variable continuas no negativas* que miden la cantidad de periodos en exceso de la *jornada laboral* más larga para la currícula g con respecto al límite prescrito L^{max}, i.e.:

$$\mathrm{L}_g := \text{máx}\left\{\text{máx}_{d\in D}\left\{t^2_{gd} - t^1_{gd} - L^{max}\right\}, 0\right\}.$$

3.3. Función Objetivo

La función objetivo corresponde a la suma ponderada de dos componentes. En la primera componente se suman los costos correspondientes a las penalizaciones por violación de preferencias de los profesores en el horario. Este valor puede ser calculado mediante la expresión:

$$\sum_{s\in S}\sum_{h\in H_s}\sum_{r\in R_s} w_{sh} y_{shr}.$$

En la segunda componente se suman los costos de penalización por violar la duración máxima prescrita para una *jornada laboral* sobre todo $g \in G$:

$$W_L \sum_{g\in G} L_g.$$

Debido a que se pretende obtener un horario que respete la preferencia de los profesores y sea lo más compacto posible, la función objetivo debe ser minimizada. A continuación se describen y formulan cada una de las familias de restricciones involucradas en el modelo.

3.4. Restricciones

3.4.1. Compatibilidad de sesiones, aulas y periodos

$$\sum_{r\in R_s}\sum_{h\in H_s} y_{shr} = 1, \qquad \forall s \in S.$$

Cada sesión $s \in S$ debe programarse para iniciar exactamente en un aula compatible y en un periodo permitido, que corresponden a los conjuntos R_s y H_s, respectivamente.

3.4.2. Límite diario de sesiones de un mismo curso

$$\sum_{s \in S_c} \sum_{r \in R_s} \sum_{h \in H_d \cap H_s} y_{shr} \leq 1, \qquad \forall c \in C - C^*, \forall d \in D.$$

No debe programarse más de una sesión $s \in S_c$ de un mismo curso $c \in C - C^*$ en un mismo día $d \in D$, la misma que debe tener lugar en un aula compatible $r \in R_s$ en un periodo permitido $h \in H_s \cap H_d$.

Notar que estas restricciones no se aplican para los cursos pertenecientes al conjunto C^* que corresponde a las materias de Formación Básica, pues para estas asignaturas sí está permitido programar más de una sesión por día.

3.4.3. Disponibilidad de aulas

$$\sum_{s \in S_r} \sum_{k \in K_{sh}} y_{skr} \leq \psi_{hr}, \qquad \forall h \in H, \forall r \in R.$$

Estas restricciones toman en cuenta la disponibilidad ψ_{hr} de las aulas en cada periodo. Dados un aula $r \in R$ y periodo $h \in H$, una sesión $s \in S_r$ puede ser programada en r para empezar en h solamente si el aula r está disponible en ese periodo. La suma sobre el conjunto K_{sh} se emplea para tomar en cuenta todos los periodos en los cuales la sesión s pudo haber iniciado para que se encuentre en ejecución en el período h.

3.4.4. Conflictos entre profesores

$$\sum_{c \in C_t} \sum_{s \in S_c} \sum_{k \in Ksh} \sum_{r \in R_s} y_{skr} \leq 1, \qquad \forall h \in H, \forall t \in T.$$

Esta restricción evita que se programe más de una sesión de alguno de los cursos dictados por el profesor $t \in T$, en un mismo periodo $h \in H$. El conjunto de sesiones asociadas al profesor $t \in T$ corresponde a las sesiones de los cursos del conjunto C_t, el conjunto de periodos en los cuales pueden tener lugar es H_s y el conjunto de aulas compatibles con cada sesión es R_s.

El conjunto K_{sh} se emplea para tomar en cuenta todos los periodos en los que pudo haber iniciado la sesión s de tal forma que esté siendo dictada durante el periodo h.

3.4.5. Conflictos entre currículas

$$\sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in Ksh} \sum_{r \in R_s} y_{skr} \leq 1, \qquad \forall h \in H, \forall g \in G.$$

De manera similar a la anterior familia de restricciones, estas desigualdades evitan que se programe más de una sesión perteneciente a los cursos de una currícula $g \in G$ en un mismo periodo $h \in H$.

3.4.6. Desigualdades de compacidad

$$t^1_{gd} \leq a_h + M(1 - \sum_{c \in C_g} \sum_{s \in S_{ch}} \sum_{r \in R_s} y_{shr}), \qquad \forall g \in G, \forall d \in D, h \in H_d;$$

$$t^2_{gd} \geq (a_h + 1) \sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr}, \qquad \forall g \in G, \forall d \in D, h \in H_d;$$

$$L_g \geq t^2_{gd} - t^1_{gd} - L^{max}, \qquad \forall g \in G, \forall d \in D.$$

Finalmente, estas restricciones se emplean para fijar los valores de las variables t^1_{gd},t^2_{gd} y L_g a partir de los valores de las variables de asignación y_{shr}. La primera familia de restricciones de enforzamiento indica que $t^1_{gd} \leq a_h$ debe cumplirse cada vez que hay una sesión de algún curso de la currícula g que está programada para empezar en el período h del día d. De manera similar, la segunda familia de restricciones requiere que $t^2_{gd} \geq a_h + 1$ se cumpla cada vez que hay una sesión de algún curso de la currícula g que está programada para dictarse en el período h del día d.

3.4.7. Formulación Completa

A continuación presentamos la formulación de programación lineal entera descrita en las secciones anteriores.

$$\min \sum_{s \in S} \sum_{h \in H_s} \sum_{r \in R_s} w_{sh} y_{shr} + W_L \sum_{g \in G} L_g \tag{1}$$

s.t.

$$\begin{aligned}
\sum_{r \in Rs} \sum_{h \in H_s} y_{shr} = 1, &\qquad \forall s \in S && (2)\\
\sum_{s \in S_c} \sum_{r \in Rs} \sum_{h \in H_d \cap H_s} y_{shr} \leq 1, &\qquad \forall c \in C - C*, \forall d \in D && (3)\\
\sum_{s \in S_r} \sum_{k \in K_{sh}} y_{skr} \leq \psi_{hr}, &\qquad \forall h \in H, \forall r \in R && (4)\\
\sum_{c \in C_t} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr} \leq 1, &\qquad \forall h \in H, \forall t \in T && (5)\\
\sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr} \leq 1, &\qquad \forall h \in H, \forall g \in G && (6)\\
t^1_{gd} \leq a_h + M(1 - \sum_{c \in C_g} \sum_{s \in S_{ch}} \sum_{r \in R_s} y_{shr}), &\qquad \forall g \in G, \forall d \in D, h \in H_d && (7)\\
t^2_{gd} \geq (a_h + 1) \sum_{c \in C_g} \sum_{s \in S_c} \sum_{k \in K_{sh}} \sum_{r \in R_s} y_{skr}, &\qquad \forall g \in G, \forall d \in D, h \in H_d && (8)\\
L_g \geq t^2_{gd} - t^1_{gd} - L^{max}, &\qquad \forall g \in G, \forall d \in D && (9)\\
y_{shr} \in \{0, 1\}, &\qquad \forall s \in S, h \in H_s, r \in R_s &&\\
t^1_{gd}, t^2_{gd} \geq 0, &\qquad \forall g \in G, d \in D &&\\
L_g \geq 0, &\qquad \forall g \in G. &&
\end{aligned}$$

Capítulo 4

Resultados Computacionales

En este capítulo se presentan los resultados computacionales arrojados durante la implementación del modelo, a partir de instancias basadas en la planificación académica de los semestres 2020-B y 2021-A de la Facultad de Ciencias de la Escuela Politécnica Nacional.

Para la implementación del modelo se hizo uso de los cuadernos Jupyter, que permitieron guardar cada versión y etapa del desarrollo del modelo, a partir del lenguaje de programación Python 3.0, y la interfaz Gurobi, para la construcción y ejecución del modelo. La versión del solver Gurobi utilizado fue9.0.3. Y el tipo de laptop utilizada fue una Intel(R) Core i7 2.70 GHz con 12 GB de RAM y Windows 10.

4.1. Gurobi Python

El solver Gurobi fué utilizado en este proyecto como motor de optimización computacional, que además de servir para resolver problemas de programación lineal *LP*, programación cuadrática *QP* y programación entera mixta *MIP*, entre otros; cuenta con una interfaz de Python compatible con los cuadernos Jupyter utilizados, que proporciona una serie de beneficios. Principalmente, cuenta con un conjunto básico de módulos de Python llamado "*gurobipy*" y una API para todas las funciones de Gurobi. Además, también incluye algunas construcciones de nivel superior que le permiten construir modelos usando una sintaxis más matemática, similar a la forma en que podría trabajar con un lenguaje de modelado tradicional. Para mas información visite:`https://www.gurobi.com/documentation/quickstart.html`

Para la implementación del modelo se utilizó el ambiente predeterminado de la interfaz de Gurobi Python.

La implementación se realizó de forma gradual, inicialmente se crea un objeto vacío de clase *Model*, de la siguiente manera:

```
from gurobipy import *
m=Model('scheduling')
```

La clase Model provee la funcionalidad para definir las **variables de decisión** del modelo (objetos de clase *Var* o *MVar*), acotadas por un límite superior o inferior y caracterizadas por su tipo (continuo, binario, etc.). Estas variables se crearon llamando a métodos *Model.addVar* y *Model.addVars*. Por ejemplo la creación de las variables binarias $y_{s,h,r}$ para todo $s \in S$, para todo $h \in H_s[s]$ y todo $r \in R_s[s]$, se realiza de la siguiente manera:

```
y=tupledict()
for s in S:
    for h in Hs[s]:
        for r in Rs[s]:
            y[s,h,r]=m.addVar(vtype=GRB.BINARY, name="y_{}_{}_
               {}".format(s, h, r))
m.update()
```

Las familias de **restricciones** del modelo (objetos de clase *Constr*, *MConstr*, entre otros), fueron agregadas utilizando los métodos *Model.addConstr* y *Model.addConstrs*. A continuación, presentamos la creación de la restricción (2) del modelo revisado en la sección 3.4.7:

```
m.addConstrs((sum([y.sum(s,Hs[s],Rs[s])])==1) for s in S)
m.update()
```

Las variables y_{shr} correspondientes a las sesiones de cursos del Departamento Formación Básica, son fijadas a valores predeterminados mediante restricciones de igualdad para reflejar la preasignación de estas sesiones a ciertas aulas y periodos:

```
#CP ALGEBRA LINEAL E - GR6 CAICEDO ROJAS BONILLA
m.addConstr(y[939,36,'SESA10']==1)#1
```

```
m.addConstr(y[987,8,'SESA10']==1)#2
```

Luego, la **función objetivo** se especifica mediante el método *Model.setObjective*, que por defecto minimiza la función establecida. Para nuestro caso y correspondiente a la función objetivo en (1) del modelo revisado en la sección 3.4.7, tenemos:

```
m.setObjective(quicksum(y[s,h,r]*Wsh[s,h] for s in S for h in Hs[s]                  for r in Rs[s])
                 +Wl*quicksum(L[g] for g in G))
m.update()
```

La mayoría de la información del modelo es almacenada en forma de atributos de las clases Model y GurobiVar a los que se puede acceder de la siguiente manera:

Atributo	**Descripción**	**Ejemplo**
VarName	Nombre de la Variable	Y.VarName
x	Valor de la variable	Y.x

Tabla 4.1: Atributos de las variables (Clase: Var)

Atributo	**Descripción**	**Ejemplo**
NumVars	Número de variables.	m.NumVars
NumConstrs	Número de restricciones.	m.NumConstrs
ModelName	Nombre del modelo.	m.ModelName
ObjVal	Valor objetivo de la solución actual.	m.ObjVal
Runtime	Tiempo de ejecución de la optimización.	m.Runtime

Tabla 4.2: Principales atributos del modelo (Clase: Model)

Otros métodos para consultar y modificar atributos son getAtrr() y setAttr(), respectivamente.

Finalmente, una vez creado el modelo se utiliza el método *Model.optimize* para calcular su solución, que en nuestro caso dispara la ejecución de los algoritmos para la solución de programas lineales enteros mixtos.

```
m.optimize()
```

El proceso de optimización puede ser controlado a través de ciertos parámetros de configuración.

Nombre del parámetro	**Descripción**
MIPGap	Brecha de optimalidad relativa de *MIP*
TimeLimit	Límite de Tiempo
MIPGapAbs	Brecha de optimalidad absoluta de *MIP*
IterationLimit	Límite de iteración simplex

Tabla 4.3: Principales parámetros del modelo utilizados

La mayoría de estos parámetros pueden ser visualizados durante la ejecución del modelo en lo que se conoce como **Registro MIP**. Para acceder a estos parámetros se utiliza el comando *Model.Params*.

Al término del proceso de optimización, se pueden consultar los valores de los atributos de cada variable, así como también valor de la función objetivo, de la siguiente manera:

Nombre y valor de cada variable del modelo:

```
for v in m.getVars():
    print('% s% g' %(v.varName, v.x))
```

Valor de la función objetivo:

```
print('Obj:% g % m.objVal')
```

4.2. Instancias

Para los experimentos computacionales se utilizaron en total 10 instancias. Se emplearon 3 instancias relativamente pequeñas para verificar la correcta formulación e implementación del modelo, 5 instancias simplificadas y 2 instancias reales basadas en la planificación académica de los semestres 2020-B (noviembre/2020 - abril/2021) y 2021-A (mayo/2021 - octubre/2021), en la Facultad de Ciencias.

Dentro del horario de disponibilidad de los profesores, se identificaron dos niveles de preferencia "prefiero" y "no prefiero", a los cuales se les asignaron dos valores de peso w_{th}=0 y $w_{th} = 10$, respectivamente. Por lo tanto, estos valores conforman los coeficientes de la primera componente de la Función Objetivo correspondiente a la penalización por violación de preferencia de los profesores.

Adicionalmente, para cada instancia se probaron tres valores diferentes del parámetro de compacidad L_{max}, que prescribe la longitud límite de una jornada laboral permitida sin incurrir en penalidad. Y se estableció el coeficiente de penalización de la segunda componente $W_L = 10$, es decir, se impone una penalización de 10 unidades por cada periodo que exceda la longitud límite de una jornada laboral prescrita.

Los tres tipos de instancias descritas a continuación, fueron construidas inicialmente desde las pequeñas para implementar el modelo hasta las más grandes para conseguir simular la asignación de horarios dentro de la facultad.

Instancias Pequeñas: Para estas instancias se consideró información extraída de la Carrera de Ingeniería Matemática y de algunos de sus niveles, dentro de la planificación académica 2020-B. Estas son las instancias 1,2 y 3.

Instancias Simplificadas: Son instancias basadas en la planificación académica de los periodos 2020-B y 2021-A, pero realizando sobre ellas algunas simplificaciones de modo que se pueda analizar el comportamiento del modelo ante alteraciones en sus características, como por ejemplo, como el número de cursos, currículas, preferencia de profesores o la presencia de variables prefijadas. Estas son las instancias 4,5,6,7 y 8.

Instancias Reales: Son instancias correspondientes a la planificación académica 2020-B y 2021-A, estas instancias son la 9 y 10, respectivamente.

A continuación se describe con mayor detalle cada una de estas instancias:

Instancias										
Instancia	Carreras	Cursos	Sesiones	Currículas	Profesores	Días	Periodos	Aulas	Variables	Restricciones
1	1	18	36	5	6	5	30	4	3655	901
2	1	18	36	5	6	5	30	4	1505	901
3	1	31	63	7	8	5	60	4	7863	2198
4	4	115	284	56	69	6	84	18	179273	22730
5	4	155	359	102	83	6	84	36	277971	37529
6	4	143	335	74	77	6	84	36	271037	29831
7	4	115	284	56	69	6	84	18	146101	22730
8	4	110	277	59	62	6	84	18	167064	23131
9	4	155	359	74	83	6	84	18	245153	30354
10	4	143	335	63	77	6	84	18	242734	26987

Tabla 4.4: Características generales de las instancias utilizadas.

Instancias Pequeñas

1. **Intancia 1:**
    - Todas las conferencias tienen la misma duración (2 horas).
    - Se considera una currícula por nivel.
    - No se incluyen restricciones de disponibilidad de profesores y disponibilidad de aulas.
    - No se incluyen restricciones de capacidad o tipología de aulas, se considera un solo tipo de aula (salón de clase). Todas las aulas son compatibles con todas las conferencias.
    - No existen variables prefijadas.
2. **Instancia 2:**
    - Existen conferencias con diferente duración (2 o 3 horas).
    - Se considera una currícula por nivel.
    - Se incluyen restricciones de disponibilidad de profesores, disponibilidad de aulas.
    - Existen restricciones por la capacidad y tipología de aulas, se consideran dos tipos de aulas (salón de clase y laboratorio de computación).
    - No existen variables prefijadas.
3. **Instancia 3:**

    Es idéntica a la **Instancia 2** en sus características pero se incrementa el número de cursos, profesores, aulas y periodos disponibles.

Instancias Simplificadas

Se considera una currícula por carrera, nivel y por grupo de estudiantes. Por ejemplo, si en el primer nivel (semestre) de la carrera de Física existen dos cursos de la materia 'Mecánica Cuántica II' para los grupos 'A' y 'A1', con los nombres 'Mecánica Cuántica II F - A' y 'Mecánica Cuántica II F - A1 ', respectivamente. Entonces, se generarían dos currículas para cada curso por separado.

4. **Instancia 4:** Basada en la planificación académica 2020-B pero no se toma en cuenta la disponibilidad/preferencia de periodos de profesores y tampoco las materias de formación básica.
5. **Instancias 5-6:** Basadas en la planificación académica 2020-B y 2021-A, respectivamente; no se toma en cuenta la disponibilidad/preferencia de tiempo de profesores y sí las materias del Departamento de Formación Básica aunque sin horarios prefijados.
6. **Instancias 7-8:** Basadas en la planificación académica 2020-B y 2021-A, respectivamente; pero no se toman en cuenta las asignaturas del Departamento de Formación Básica y sí la disponibilidad/preferencia de tiempo de profesores.

Instancias Reales

Se considera el número de currículas consultadas al personal de encargado de generar el horario de clases en la Facultad de Ciencias.

9. **Instancias 9-10:** Se trata de la planificación académica 2020-B y 2021-A, considerando todas sus características incluido que se prefijan las sesiones de los cursos del Departamento de Formación Básica.

   Nota: Al decir que no se toma en cuenta la disponibilidad/preferencia de tiempo de profesores, suponemos que cualquier profesor está disponible en cualquier periodo entre el Lunes y Viernes.

Es importante notar que en la formulación del modelo no se crean variables innecesarias. Por ejemplo: si la disponibilidad de un profesor, aula o duración de una sesión no permiten que esta se lleve a cabo en un periodo determinado, entonces no es necesario crear una variable o conjunto de variables asociadas a esta sesión en ese periodo. Este resultado lo podemos apreciar en la cantidad de variables y restricciones de cada instancia indicada en la **tabla 4.4**.

4.3. Resultados Computacionales

En la **tabla 4.5** se encuentran resumidos los resultados obtenidos en las pruebas computacionales. La tabla está seccionada en 3 partes, correspondientes a los valores de compacidad indicados en la columna "L_{max}", para las 10 instancias mencionadas en la sección anterior. Dentro de cada sección tenemos la siguiente información por columnas: "Nodos $B\&B$", indica el número de nodos explorados por el algoritmo Branch & Bound; "Función Objetivo", muestra el valor de la función objetivo de la mejor solución encontrada; "Gap", presenta la brecha de optimalidad alcanzada; y "Tiempo de Ejecución", indica el tiempo utilizado en la solución del modelo.

El criterio de parada para cada sección se fijó mediante el parámetro del modelo $TimeLimit$ en 1000s, 2000s y 7200s, respectivamente. En ninguna de las pruebas se fijó un límite para los parámetros de brechas de optimalidad $MIPGap$ o $MIPGapAbs$.

Resultados						
L_{max}	Instancia	Nodos $B\&B$	Función Objetivo	Cota Inferior	Gap	Tiempo de Ejecución
15	1	0	40	40	0 %	0.42s
	2	0	100	100	0 %	4.13s
	3	0	70	70	0 %	1.78s
	4	0	0	0	0 %	13.24s
	5	0	0	0	0 %	37.05s
	6	0	0	0	0 %	21.72s
	7	0	0	0	0 %	25.82s
	8	0	0	0	0 %	15.02s
	9	0	0	0	0 %	33.58s
	10	0	0	0	0 %	39.32s
8	1	0	40	40	0 %	0.61s
	2	58	100	100	0 %	7.19s
	3	0	110	110	0 %	29.51s
	4	2543	50	0	100 %	2000s
	5	1	660	0	100 %	2000s
	6	1	0	0	0 %	623.65s
	7	1	50	20	60 %	2000s
	8	1	0	0	0 %	87.70s
	9	8	290	290	0 %	1547.37s
	10	1	50	50	0 %	111.66s
6	1	0	40	40	0 %	0.86s
	2	8	100	100	0 %	2.36s
	3	7995	130	110	15.38 %	1000s
	4	34	320	0	100 %	2000s
	5	1	2110	0	100 %	2000s
	6	1	340	0	100 %	2000s
	7	1	440	20.02	95.45 %	2000s
	8	2531	50	0	100 %	2000s
	9	2867	780	570	26.92 %	7200s
	10	1	220	220	0 %	1353.67s

Tabla 4.5: Resultados computacionales.

En términos generales, mediante las pruebas computacionales realizadas hemos podido probar el funcionamiento y desempeño del modelo, además de brindarnos algunas ideas de las relaciones existentes entre las características de cada instancia el 'Tiempo de Ejecución', la brecha de optimalidad relativa 'Gap', mejor 'Cota Inferior' encontrada,'Función Objetivo' y 'Nodos $B\&B$' explorados.

En primer lugar podemos observar que cuando el parámetro de compacidad $L_{max} = 15$, el solver proporciona soluciones óptimas en menos de un minuto de ejecución para todas las instancias consideradas, esto debido a que la longitud de una jornada laboral permitida es de hasta 15 horas consecutivas sin incurrir en ninguna penalización, es decir, se permite la generación de horarios de clase que comiencen a las $07:00$ $a.m.$ y culminen a las $22:00$ $p.m.$.

Horas	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7-8	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---	---	---
8-9	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---	---	---
9-10	ECONOMETRIA I ICEF,E - A	---	---	ECONOMETRIA I ICEF,E - A	---	---
10-11	ECONOMETRIA I ICEF,E - A	---	---	ECONOMETRIA I ICEF,E - A	---	---
11-12	---	ECONOMETRIA I ICEF,E - A	---	---	---	---
12-13	---	ECONOMETRIA I ICEF,E - A	ECONOMETRIA I ICEF,E - A	---	---	---
13-14	---	---	---	MERCADOTECNIA ICEF - A	---	---
14-15	---	TEORIAS DEL DESARROLLO - A	---	MERCADOTECNIA ICEF - A	TEORIAS DEL DESARROLLO - A	---
15-16	---	TEORIAS DEL DESARROLLO - A	---	---	TEORIAS DEL DESARROLLO - A	---
16-17	---	---	TEORIAS DEL DESARROLLO - A	TEORIA MONETARIA ICEF,E - A	---	---
17-18	---	TEORIA MONETARIA ICEF,E - A	TEORIAS DEL DESARROLLO - A	TEORIA MONETARIA ICEF,E - A	---	---
18-19	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	TEORIA MONETARIA ICEF,E - A	MERCADOTECNIA ICEF - A	---	TEORIA MONETARIA ICEF,E - A	---
19-20	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	MERCADOTECNIA ICEF - A	---	TEORIA MONETARIA ICEF,E - A	---
20-21	---	---	---	---	---	---

Figura 4.1: Horario de las asignaturas correspondientes a la currícula del tercer nivel de la carrera de Economía contemplando una jornada laboral de hasta 15 horas (Instancia 8).

Cuando se fijó el parámetro de compacidad $L_{max} = 8$, se comenzaron a obtener soluciones factibles no óptimas en 3 de las 10 instancias consideradas. Como podemos observar en la **tabla 4.5**, para el caso de las **instancias pequeñas**, existe un incremento en el tiempo de ejecución aunque se mantiene una brecha de optimalidad relativa 'Gap' igual a 0 %. En el caso de las **instancias simplificadas**, únicamente las **instancias 6 y 8** alcanzaron soluciones óptimas dentro del límite de tiempo establecido, y por otro lado, dentro de las instancias que no alcanzaron su solución óptima, la solución de la **instancia 7** es la más cercana a su óptimo con un 'Gap' de 60 %, seguida de la soluciones de las **instancias 4 y 5**, en las que no se logró reducir sus brechas de optimalidad relativa de 100 % con respecto a sus cotas inferiores, las cuales mantuvieron el valor de cero. Finalmente, en los resultados de las **instancias reales**, podemos observar que el modelo logra calcular en ambos casos las soluciones óptimas, con valores de penalidad de 290 (penalidad de compacidad=290 , penalidad de preferencia=0) y 50 (penalidad de compacidad=50, penalidad de preferencia=0), respectivamente. Sin embargo, los tiempos de cálculo son muy distintos: para la **instancia 9** se requieren de casi 40 minutos de cálculo, mientras que la instancia 10 se resuelve en algo más de dos minutos, esto debido principalmente a que son instancias basadas en dos semestres diferentes.

Horas	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7-8	---	---	---	---	---	---
8-9	---	---	---	---	---	---
9-10	TEORIAS DEL DESARROLLO - A	---	---	---	---	---
10-11	TEORIAS DEL DESARROLLO - A	---	---	---	ECONOMETRIA I ICEF,E - A	---
11-12	ECONOMETRIA I ICEF,E - A	ECONOMETRIA I ICEF,E - A	ECONOMETRIA I ICEF,E - A	---	ECONOMETRIA I ICEF,E - A	---
12-13	---	ECONOMETRIA I ICEF,E - A	ECONOMETRIA I ICEF,E - A	---	TEORIAS DEL DESARROLLO - A	---
13-14	---	---	TEORIAS DEL DESARROLLO - A	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	TEORIAS DEL DESARROLLO - A	---
14-15	MERCADOTECNIA ICEF - A	---	TEORIAS DEL DESARROLLO - A	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	MERCADOTECNIA ICEF - A	---
15-16	MERCADOTECNIA ICEF - A	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	MERCADOTECNIA ICEF - A	---
16-17	---	TEORIA MONETARIA ICEF,E - A	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	TEORIA MONETARIA ICEF,E - A	TEORIA MONETARIA ICEF,E - A	---
17-18	---	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	TEORIA MONETARIA ICEF,E - A	---
18-19	---	---	---	---	---	---
19-20	---	---	---	---	---	---
20-21	---	---	---	---	---	---

Figura 4.2: Horario de las asignaturas correspondientes a la currícula del tercer nivel de la carrera de Economía contemplando una jornada laboral de hasta 8 horas (Instancia 8).

Por otro lado, al fijar el parámetro de compacidad en $L_{max} = 6$ no se encuentran soluciones óptimas dentro del límite de tiempo establecido en siete de las diez instancias. Solamente las instancias pequeñas 1 y 2, y la instancia real 10 pudieron ser resueltas hasta la optimalidad. De los resultados obtenidos y registrados en la **tabla 4.5**, para las **instancias pequeñas**, únicamente para la **instancia 3** no fue posible encontrar una solución óptima dentro del tiempo límite establecido de 1000s, pues la mejor solución encontrada hasta ese momento presenta una brecha de optimalidad del 15.38 % (Función Objetivo=130, Cota Inferior =110). Para el caso de las **instancias simplificadas**, el modelo no logró encontrar una solución óptima dentro del tiempo límite establecido. En general, para ninguna instancia se alcanza una brecha de optimalidad relativa 'Gap' por debajo del 95 %, siendo la mejor solución obtenida la presentada para la **instancia 8**, seguida de las **instancias 6, 4, 7 y 5**. De estos resultados se concluye que la complejidad del problema se ve afectada principalmente por las características ligadas a las restricciones fuertes del problema como el número de cursos, currículas y periodos disponibles por profesor. Finalmente, en base a los resultados obtenidos para las **instancias reales**, podemos argumentar que la calidad de las soluciones se ve afectada principalmente por el número de sesiones de los cursos del Departamento de Formación Básica que se prefijan.

Horas	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7-8	---	---	---	TEORIAS DEL DESARROLLO - A	---	---
8-9	---	---	---	TEORIAS DEL DESARROLLO - A	---	---
9-10	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	ECONOMETRIA I ICEF,E - A	---	---
10-11	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	ECONOMETRIA I ICEF,E - A	---	---
11-12	---	ECONOMETRIA I ICEF,E - A	---	---	ECONOMETRIA I ICEF,E - A	---
12-13	ECONOMETRIA I ICEF,E - A	ECONOMETRIA I ICEF,E - A	MERCADOTECNIA ICEF - A	---	ECONOMETRIA I ICEF,E - A	---
13-14	---	---	MERCADOTECNIA ICEF - A	---	TEORIAS DEL DESARROLLO - A	---
14-15	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	MERCADOTECNIA ICEF - A	TEORIAS DEL DESARROLLO - A	---	TEORIAS DEL DESARROLLO - A	---
15-16	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	MERCADOTECNIA ICEF - A	TEORIAS DEL DESARROLLO - A	---	---	---
16-17	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---
17-18	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---
18-19	---	---	---	---	---	---
19-20	---	---	---	---	---	---
20-21	---	---	---	---	---	---

Figura 4.3: Horario de las asignaturas correspondientes a la currícula del tercer nivel de la carrera de Economía contemplando una jornada laboral de hasta 6 horas (Instancia 8).

Como hemos podido apreciar durante esta sección de resultados computacionales, los peores resultados se presentaron cuando consideramos todas las combinaciones posibles de currículas provocando que el problema sea cada vez más complejo conforme se reducía el parámetro de compacidad, y que al momento de considerar las currículas utilizadas por la facultad el modelo obtuvo mejores resultados. De este modo, las instancias simplificadas resultaron ser las más difíciles de resolver, pues los valores 'Gap' fueron en general los peores. En la siguiente sección proponemos contrastar las soluciones manuales utilizadas por la Facultad de Ciencias frente a las soluciones de nuestro modelo para los períodos académicos correspondientes a las dos instancias reales.

4.4. Eficacia del Modelo

En esta sección compararemos la solución obtenida por nuestro modelo, con la solución empleada por la Facultad de Ciencias, la misma que fue obtenida manualmente y se encuentra documentada en el Sistema de Administración Estudiantil (SAEw). La solución de nuestro modelo fue calculada sin prefijar ninguna sesión.

Semestre 2020-B

Para este semestre, la solución manual usada resultó ser no factible para nuestro modelo debido a que no se respetaban varias restricciones fuertes. Por ejemplo, del total de 359 sesiones, 59 fueron asignadas a aulas con menor capacidad al número de estudiantes previstos; 10 se programaron fuera del horario de disponibilidad del profesor asociado; 9 se asignaron a un aula que inicialmente no constaba en la lista de aulas disponibles en la Facultad de Ciencias, aula "FC501"; y 5 sesiones presentaban conflicto por ser ubicadas en aulas en periodos no disponibles.

Semestre 2021-A

Al igual que para el semestre 2020-B, la solución manual empleada por la facultad no es factible para nuestro modelo; en este caso el número de sesiones que tenían conflictos fue mayor: de un total de 335 sesiones, 65 sesiones presentaban el problema de capacidad en sus aulas asignadas; 10 sesiones presentaban conflictos con la disponibilidad del profesor asignado; 3 sesiones fueron asignadas al aula "FC501";

y 15 sesiones presentaban conflicto por ser ubicadas en aulas en periodos no disponibles.

Para tener soluciones que se puedan comparar, se relajó gradualmente el problema hasta que la solución manual empleada en la Facultad de Ciencias fuera factible: se disminuyó el número de estudiantes para cada sesión que presentaba el problema de capacidad; se amplió el horario de disponibilidad para cada profesor que tenía clases programadas fuera de su horario; el aula $''FC501''$ fue añadida al listado de aulas disponibles; y para el problema de disponibilidad de aulas, se amplió la disponibilidad de cada aula hasta resolver el conflicto.

En lo que respecta a las currículas que se emplearon, debido a que en la planificación manual de la Facultad de Ciencias no se utiliza este concepto, se generaron a partir del horario final publicado en cada semestre,proceso similar al utilizado en las instancias 9 y 10. A continuación, se describe en detalle este proceso.

El procedimiento utilizado fue el siguiente:

- Para cada carrera, generamos una currícula por nivel conformada por todos sus cursos. Esto se realiza considerando las mallas curriculares de cada carrera.
- Si las currículas generadas provocan conflictos de cruce de cursos, en la solución manual cada currícula con problemas se divide en currículas más pequeñas.

Por ejemplo, supongamos tener la siguiente currícula:

$$\begin{array}{c}\text{Currícula}\\\text{inicial}\end{array}\left\{\begin{array}{c}\text{'ECONOMETRIA I E - A',}\\\text{'HISTORIA DEL PENSAMIENTO ECONOMICO E,F - A',}\\\text{'TEORIA DE JUEGOS E,IM - A',}\\\text{'HISTORIA DEL PENSAMIENTO ECONOMICO E,F - A1'.}\end{array}\right.$$

Y supongamos también que la solución manual tiene un conflicto entre los cursos 'HISTORIA DEL PENSAMIENTO ECONOMICO E,F' para los grupos A y A1, es decir, se encuentran programados a la misma hora. Entonces, esta currícula es reemplazada por las dos currículas siguientes para que se respete la restricción 3.4.5.

$$\text{Currícula 1} \begin{cases} \text{'ECONOMETRIA I E - A',} \\ \text{'HISTORIA DEL PENSAMIENTO ECONOMICO E,F - A',} \\ \text{'TEORIA DE JUEGOS E,IM - A'.} \end{cases}$$

$$\text{Currícula 2} \begin{cases} \text{'ECONOMETRIA I E - A',} \\ \text{'HISTORIA DEL PENSAMIENTO ECONOMICO E,F - A1',} \\ \text{'TEORIA DE JUEGOS E,IM - A'.} \end{cases}$$

Esta tarea se realiza progresivamente con todos los niveles de cada carrera, hasta que la solución provista por la Facultad de Ciencias sea admitida como solución factible y pueda ser evaluada por nuestro modelo.

Por otro lado, se ejecutó el modelo de optimización sobre las instancias generadas a partir de relajar las **instancias 9 y 10**, conforme lo descrito en los párrafos anteriores. El parámetro de compacidad se fijó en $L_{max} = 6$ y como criterio de parada el parámetro $TimeLimit = 7200$. De esta manera, obtuvimos los siguientes resultados:

Resultados					
Solución	Semestre	Compacidad	Preferencia	Función Objetivo	Gap
Manual	2020-B	2440	0	2440	-
	2021-A	2320	0	2320	-
Modelo	2020-B	500	0	500	(100 % Gap)
	2021-A	30	0	30	(0 % Gap)

Tabla 4.6: Resultados computacionales comparativos del modelo y la solución manual empleada por la Facultad de Ciencias.

En base a estos resultados, notamos que existe una gran mejora en la función objetivo con respecto a la solución calculada de manera manual por la Facultad de Ciencias. Tanto en la solución manual como en la del modelo, los valores de la función objetivo miden exclusivamente las penalizaciones por violaciones de la compacidad del horario.

Horas	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7-8	---	TEORIAS DEL DESARROLLO - A	---	---	---	---
8-9	---	TEORIAS DEL DESARROLLO - A	---	---	---	---
9-10	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	ECONOMETRIA I ICEF,E - A	TEORIAS DEL DESARROLLO - A	---	---
10-11	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	ECONOMETRIA I ICEF,E - A	TEORIAS DEL DESARROLLO - A	---	---
11-12	MERCADOTECNIA ICEF - A	---	---	---	TEORIAS DEL DESARROLLO - A	---
12-13	MERCADOTECNIA ICEF - A	ECONOMETRIA I ICEF,E - A	---	---	TEORIAS DEL DESARROLLO - A	---
13-14	---	---	---	---	---	---
14-15	ECONOMETRIA I ICEF,E - A	MERCADOTECNIA ICEF - A	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	ECONOMETRIA I ICEF,E - A	---
15-16	ECONOMETRIA I ICEF,E - A	MERCADOTECNIA ICEF - A	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	ECONOMETRIA I ICEF,E - A	---
16-17	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---
17-18	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---	TEORIA MONETARIA ICEF,E - A	---
18-19	---	---	---	---	---	---
19-20	---	---	---	---	---	---
20-21	---	---	---	---	---	---

Figura 4.4: Horario generado manualmente por la Facultad.

Horas	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7-8	---	---	---	---	TEORIAS DEL DESARROLLO - A	---
8-9	---	---	---	---	TEORIAS DEL DESARROLLO - A	---
9-10	---	---	---	---	MERCADOTECNIA ICEF - A	---
10-11	---	ECONOMETRIA I ICEF,E - A	---	---	MERCADOTECNIA ICEF - A	---
11-12	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---	ECONOMETRIA I ICEF,E - A	---
12-13	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---	ECONOMETRIA I ICEF,E - A	---
13-14	---	MERCADOTECNIA ICEF - A	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---
14-15	ECONOMETRIA I ICEF,E - A	MERCADOTECNIA ICEF - A	---	FORMULACION Y EVALUACION FINANCIERA DE PROYECT...	---	---
15-16	ECONOMETRIA I ICEF,E - A	---	---	TEORIAS DEL DESARROLLO - A	---	---
16-17	TEORIA MONETARIA ICEF,E - A	---	ECONOMETRIA I ICEF,E - A	TEORIAS DEL DESARROLLO - A	---	---
17-18	TEORIA MONETARIA ICEF,E - A	---	ECONOMETRIA I ICEF,E - A	TEORIA MONETARIA ICEF,E - A	---	---
18-19	TEORIAS DEL DESARROLLO - A	---	TEORIA MONETARIA ICEF,E - A	TEORIA MONETARIA ICEF,E - A	---	---
19-20	TEORIAS DEL DESARROLLO - A	---	TEORIA MONETARIA ICEF,E - A	---	---	---

Figura 4.5: Horario generado por el Modelo.

Como podemos observar en la **tabla 4.6** y en las **Figuras 4.4** y **4.5**, las soluciones obtenidas por nuestro modelo generan claramente horarios de clases más compactos que la solución manual. Por otra parte, como se observa en la **tabla 4.6**, la solución proporcionada por el modelo para el semestre 2020-B, la brecha de optimalidad indica que su solución puede eventualmente mejorar. Finalmente, la componente que penaliza la preferencia de horario de los profesores, es nula para ambas soluciones; esto se debe a que estamos utilizando una instancia relajada en la que se fijó el nivel de preferencia "prefiere"(es decir, penalización cero) casi en todos los períodos y casi para todos los profesores.

Capítulo 5

Conclusiones

A través de los capítulos, hemos descrito, presentado, e implementado un modelo de programación lineal entera que nos permitió automatizar el cálculo de horarios de acuerdo a las condiciones presentes en la Facultad de Ciencias. En las pruebas computacionales realizadas, nuestro modelo obtuvo soluciones significativamente mejores a las soluciones calculadas manualmente, con respecto al valor de una función objetivo que mide el respeto de las preferencias de horario de los profesores y un criterio de compacidad para el horario de cada uno de los niveles de la malla curricular.

Siguiendo con el análisis de los resultados computacionales, se demostró que la cantidad de recursos disponibles como aulas y profesores, resultó ser determinantes, pues de este hecho depende el número de combinaciones posibles en las que una sesión puede ser asignada (sesión,periodo,aula). Por una parte, esto se vio reflejado al momento de querer medir la calidad de solución generada por la Facultad de Ciencias, ya que existían muchos casos en los que decidieron asignar sesiones con exceso de estudiantes a un aula, asignar a profesores una sesiones en periodos en los que no estaban disponible inicialmente y también tuvieron que disponer de un aula que en un inicio no era tomada en cuenta. Esto puede deberse, en parte, a que el personal encargado de la planificación manual de los horarios, pese a su experiencia y su dedicación a esta tarea, no puede contemplar todas las posibles asignaciones, cuyo número experimenta una explosión combinatoria cuando el número de sesiones crece. Y por otro lado en lo que le corresponde al modelo, tenemos que, agregar más aulas y profesores disponibles a una instancia genera mejores resultados, esto se corroboró en los resultados computacionales de la **instancia 2** a la **instancia 3** y de la **instancia 4** a la **instancia 5**.

Otro dato que también se pudo corroborar a partir de los resultados computacionales de las 10 instancias utilizadas y posteriormente en la comparación entre la solución manual de la Facultad de Ciencias y nuestro modelo, fue que la calidad de la solución y tiempo de ejecución se degradan considerablemente al prefijar las sesiones correspondientes a las asignaturas del Departamento de Formación Básica. De hecho, en la solución generada por el modelo cuando se permite que todas las sesiones se puedan mover libremente (Cuadro 4.6) tan solo se registraron 50 horas fuera del límite de 6 horas diarias, frente a 244 horas en los horarios calculados manualmente para cada carrera para el semestre 2020-B y que fueron publicados por la Facultad en el sistema Saew.

Aún cuando las soluciones obtenidas por el modelo son satisfactorias si se las compara con las soluciones manuales, existe todavía bastante potencial para optimización. Las brechas de optimalidad obtenidas para algunas instancias son todavía altas y amerita el esfuerzo investigar técnicas más avanzadas de solución para intentar disminuirlas. De igual manera, nuestro modelo demostró ser muy susceptible a la reducción de aulas compatibles o disponibilidad de profesores, ya que una sola sesión que no pueda ser asignada puede generar la infactibilidad de toda una instancia. Por otra parte, en cuanto a la aplicabilidad del trabajo aquí desarrollado, se considera que, con una buena gestión y proyección recursos el despliegue del modelo a otras facultades de la Escuela Politécnica Nacional puede ser completamente viable.

Para investigaciones posteriores recomendaría considerar incorporar las restricciones de compatibilidad entre sesión y tipo de aula como fuerte y débil a la vez, es decir, considerando dos escenarios posibles: uno en el que una aula de clase no puede ser ocupada para sesiones de laboratorio (como una restricción fuerte), y otro escenario en el que un aula de laboratorio sí podría ser considerada apta para recibir sesiones de clases (restricción débil). De este modo, se estaría relajando el problema y se podría obtener soluciones que mejoren el uso de los recursos disponibles.

Un último aspecto que ameritaría un estudio más profundo en un trabajo futuro consiste en la aplicación de criterios alternativos para la evaluación de la compacidad del horario. En el presente modelo, hemos especificado la compacidad a partir de fijar un límite (débil) para la duración de cada jornada de clases. Alternativamente, puede penalizarse la presencia de horas huecas en el horario, o requerirse la existencia de franjas de franjas de horario libres para cada curso.

Bibliografía

[1] Abdullah, S. y Turabieh, H. On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences*, 191:146–168, 05 2012.

[2] Asratian, A. y Werra, D. A generalized class-teacher model for some timetabling problems. *European Journal of Operational Research*, 143:531–542, 02 2002.

[3] Bettinelli, A., Cacchiani, V., Roberti, R., y Toth, P. An overview of curriculum-based course timetabling. *TOP*, 23:313–349, 07 2015.

[4] Bonutti, A., De Cesco, F., Di Gaspero, L., y Schaerf, A. Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results. *Annals OR*, 194:59–70, 04 2012.

[5] Burke, E., Marecek, J., Parkes, A., y Rudová, H. Penalising patterns in timetables: novel integer programming formulations. *Operations Research*, págs. 409–414, 2008.

[6] Burke, E., Marecek, J., Parkes, A., y Rudová, H. Decomposition, reformulation, and diving in university course timetabling. *Computers Operations Research*, 37:582–597, 04 2010.

[7] Carrasco, G., Martínez, C., y Rodríguez, D. Un algoritmo heurístico para la asignación de aulas en el campus universitario. 09 2019.

[8] Casarini Ratto, M. Teoría y diseño curricular. 1999.

[9] Daskalaki, S. y Birbas, T. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160:106–120, 01 2005.

[10] Di Gaspero, L. Recolour, shake and kick: a recipe for the examination timetabling problem. En *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2002)*, págs. 404–407, Gent, Belgium, August 2002.

[11] Di Gaspero, L., McCollum, B., y Schaerf, A. The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Technical report, Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's . . . , 2007.

[12] Ernst, A., Jiang, H., Krishnamoorthy, M., y Sier, D. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153:3–27, 02 2004.

[13] Griffin, D. A survey of bahamian and jamaican teachers' level of motivation and job satisfaction. *JOURNAL OF INVITATIONAL THEORY AND PRACTICE*, 16:57–77, 01 2010.

[14] Grigorios, NB and Charalampos, NM and Georgios, PK and others. Applying evolutionary computation to the school timetabling problem: the greek case. *Computers & Operations Research*, 35:1265, 2008.

[15] Gurobi Optimization, L. Gurobi Optimizer Quick Start Guides - Python interface, 2021. `https://n9.cl/bmsj4`.

[16] Heredia, M. B. Modelo de Programación Lineal Entera para la generación de horarios de clase en la Universidad. Thesis, Escuela Politécnica Nacional, 2014. Repositorio Institucional. `https://n9.cl/2stgf`.

[17] Hertz, A. Finding a feasible course schedule using tabu search. *Discret. Appl. Math.*, 35:255–270, 1992.

[18] Jiménez, F. G. Métodos Poliedrales de Coloramiento de Grafos aplicados al problema de horarios en Universidades con Restricciones de Compacidad. Thesis, Escuela Politécnica Nacional, 2020. Repositorio Institucional. `https://n9.cl/ef7sx`.

[19] Kustoch, P. Timetabling competition s a-based heuristic. *International timetabling competition results*.

[20] Lach, G. y Lübbecke, M. Curriculum based course timetabling: New solutions to udine benchmark instances. *Annals of Operations Research*, 194:255–272, 04 2012.

[21] Lü, Z. y Hao, J.-K. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235–244, 2010.

[22] Mirhassani, S. A. y Habibi, F. Solution approaches to the course timetabling problem. *Artif. Intell. Rev.*, 39(2):133–149, feb 2013.

[23] Parker, R. y Rardin, R. *Discrete optimization*. Elsevier, USA, 1 edición, August 1988.

[24] Schaerf, A. A Survey of Automated Timetabling. *Artificial Intelligence Review*, 13:87–127, 01 1999.

[25] Schmidt, G. y Ströhlein, T. Timetable construction - an annotated bibliography. *Comput. J.*, 23:307–316, 1980.

[26] Thompson, J. M. y Dowsland, K. A. A robust simulated annealing based examination timetabling system. *Comput. Oper. Res.*, 25:637–648, 1998.

[27] Torres, L. M. y Torres, D. R. University timetabling with heterogeneous lectures and compactness constraints. Technical report, Escuela Politécnica Nacional, 2019.

[28] Werra, D. An introduction to timetabling. *European Journal of Operational Research*, 19:151–162, 1985.

[29] West, D. B. *Introduction to Graph Theory*. Prentice Hall, 2 edición, September 2000.

[30] White, G. y Xie, B. Examination timetables and tabu search with longer-term memory. volume 2079, págs. 85–103, 08 2000.

[31] White, G., Xie, B., y Zonjic, S. Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, 153:80–91, 02 2004.

[32] Wren, A. Scheduling, Timetabling and Rostering-A Special Relationship? En *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformática)*, volume 1153. Springer, Berlin, Heidelberg, 1996.