UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

MODELOS DE OPTIMIZACIÓN PARA LA PROGRAMACIÓN DE HORARIOS Y ASIGNACIÓN DE SALAS DE CLASE EN UNIVERSIDADES

TESIS PARA OPTAR AL GRADO DE DOCTOR EN SISTEMAS DE INGENIERÍA

JAIME ANDRÉS MIRANDA PINO

PROFESOR GUÍA:
RICHARD WEBER HAAS

MIEMBROS DE LA COMISIÓN:
DANIEL ESPINOZA GONZÁLEZ
VLADIMIR MARIANOV KLUGE
FERNANDO ORDÓÑEZ PIZARRO
JAVIER MARENCO

SANTIAGO DE CHILE
MAYO 2014

Resumen

Hoy en día, la planificación de la capacidad y la administración de las actividades son dos problemas centrales para cualquier directivo en una universidad. Si bien los objetivos de la planificación de la capacidad difieren de los objetivos que se persiguen con la programación de actividades, eśtos están ligeramente relacionados. En el corto plazo, una mala programación de las actividades genera múltiples problemas operativos, como por ejemplo: la existencia de conflictos horarios entre cursos que deben ser inscritos por un mismo grupo de estudiantes o la asignación de una sala de clase con capacidad inferior a la requerida, es natural pensar que estas descoordinaciones provocarán un descontento general, tanto en los estudiantes y profesores, como en los directivos de estas unidades académicas. En el largo plazo, si la planificación de la capacidad está muy por debajo de los requerimientos de las mallas curriculares, disminuirá considerablemente la calidad del servicio, no habrá espacio disponible para programar ciertos cursos, ni tampoco profesores idóneos para dictar sus sesiones. Mientras que si planificamos la capacidad por sobre la demanda de requerimientos, ocasionará un aumento considerable en los costos de inversión y costos de operación.

Este trabajo de tesis tiene como objetivo principal dar direcciones para la mejor utilización de los recursos en las universidades y la planificación de éstos en el tiempo. Para el corto plazo se presentan modelos de optimización basados en patrones que permiten resolver el problema de programación de horarios y asignación de salas de clase en universidades. Los modelos de optimización basados en patrones facilitan la utilización de paquetes comerciales que permiten resolverlos. Mientras que para el largo plazo, se presenta un enfoque de solución que determina la planificación de la capacidad para hacer frente a cambios en las matrículas. Este enfoque de solución propuesto se basa en la resolución de un modelo de optimización que utiliza como información de entrada el pronóstico de dos modelos predictivos. El primer modelo determina el número de estudiantes que ingresarán a primer año durante el período de planificación mediante un modelo de series de tiempo, mientras que el segundo modelo determina el número de estudiantes que se inscribirán en cada curso y período simulando el paso de los estudiantes dentro de una malla curricular.

Todos los enfoques de solución presentados en esta tesis fueron adaptados para ser aplicados en tres instituciones académicas. En general, al aplicar estos enfoques fue posible obtener mejoras significativas respecto de los enfoques de solución manuales, como por ejemplo: reducción de costos operativos, eliminación de conflictos horarios, un mejor uso de los recursos actuales y la planificación del crecimiento de éstos en el largo plazo.

A mis padres, abuela y hermana... Una familia incondicional...

Agradecimientos

Me gustaría expresar mi agradecimiento a todos los que me ayudaron a que este trabajo de tesis saliera adelante y de forma satisfactoria. En especial quiero agradecer por su tiempo y dedicación al profesor Richard Weber, quien durante todo este tiempo fue un guía y amigo, siendo pilar fundamental de los logros obtenidos en este trabajo. Quiero también expresar un agradecimiento especial a Daniel Espinoza, por su generosidad en la entrega de conocimientos, por su paciencia y por los incontables buenos consejos que me dio en los momentos difíciles de este trabajo tesis. También me gustaría expresar mis agradecimientos a los demás miembros de la comisión de tesis, profesores Marianov, Ordóñez y Marenco, por sus importantes comentarios que sirvieron de gran ayuda en el desarrollo de esta tesis.

Es importante destacar que todo este trabajo no se hubiera podido llevar a cabo sin el apoyo incondicional de mis seres queridos. Gracias familia por su apoyo y valiosos consejos en todos estos años. Agradezco también de forma muy especial a Karolina por su enorme cariño, calidez y apoyo incondicional. Sin todos ustedes nada hubiera sido posible.

Finalmente, quiero dar las gracias al director del Instituto Milenio Sistemas Complejos de Ingeniería, profesor Andrés Weintraub, por su generosidad y calidez que fueron fundamentales para llegar a buen término de esta etapa. Agradezco también a las personas que trabajan para el Doctorado en Sistemas de Ingeniería y, de manera muy especial, a Fernanda Melis y Julie Lagos por todo el apoyo brindado.

Tabla de contenido

Índice de tablas

Índice de figuras

Introducción

Capítulo 1

Introducción

1.1. Contexto y motivación

Por décadas, con el fin de mejorar la calidad de los servicios que ofrecen a sus estudiantes y profesores, las instituciones de educación superior han utilizado sistemas basados en modelos de optimización para soportar un amplio rango de decisiones operativas. Dentro de este contexto, tres problemas claves para cualquier directivo de una universidad son: 1) la programación de las actividades que realizan durante el período académdico; 2) el uso eficiente de la infraestructura que posee actualmente y; 3) la planificación de la capacidad, entendida como el dimensionamiento adecuado de los recursos necesarios para poder operar y su evolución en el tiempo. Claramente, estos problemas se entremezclan entre si, ya que la planificación de la capacidad es un input para la programación de actividades de largo plazo y viceversa. Por tanto, si somos capaces de resolver estos tres problemas de manera conjunta será posible determinar la solución óptima del problema global y, por ende, se determinará de manera más precisa los recursos que demanda una universidad para realizar sus actividades académicas.

En el corto plazo, los problemas 1) y 2) se traducen en determinar la programación horaria de las actividades (como cursos, seminarios, o charlas) que imparten las universidades y la asignación de la infraestructura más adecuada para llevarlas a cabo. Generalmente, estos problemas deben ser resueltos semestre a semestre, no siendo posible replicar completamente las programaciones de semestres anteriores. Esto se explica principalmente por el dinamismo intrínsico con que funcionan las universidades, observándose por ejemplo, que las disponibilidades horarias de los profesores cambian constantemente; que existen cambios en las mallas curriculares, tanto en los requisitos, como en la creación de nuevos cursos; que se crean nuevos planes de estudio que compiten por el espacio físico común; así como, que constantemente los requerimientos de los cursos por ciertos tipos de salas de clase también cambian.

Ahora bien, si consideramos que una mala programación horaria puede producir por ejemplo: conflictos entre cursos que deben ser inscritos por un mismo grupo de estudiantes, actividades en horarios no deseados o que se asignen actividades a salas de clase con capacidad insuficiente, es claro que estos errores provocarán un descontento general, tanto en el alumnado y profesores, como en los directivos de estas universidades. El descontento afecta negativamente la imagen de las universidades y aumenta la percepción de desorganización y caos interno. Si consideramos que estos errores pueden provocar un aumento en los costos de operación asociados a: tener que arrendar salas de clase externas a la universidad; tener que financiar mecanismos de compensación a los estudiantes afectados; costos de oportunidad al subutilizar las salas de clase, permitiendo un gran número de asientos vacíos, entre otros, realzan la importancia y necesidad de contar con enfoques de solución más sofisticados que resuelvan estos problemas de forma eficiente. En consecuencia, el resolver el problema de programar los horarios y asignar las salas de clase se transforma en un tema central en todas aquellas universidades que pretendan ser eficientes en la asignación de sus recursos escasos.

En el largo plazo, el problema 3) se traduce en dimensionar dos de los recursos más importantes, como por ejemplo: salas de clase y profesores en el tiempo. Específicamente, se debe determinar el número y tipo de salas de clase que deben ser arrendadas, remodeladas o construídas, así como el número y tipo de profesores que serán necesarios para hacer frente a variaciones (aumentos o disminuciones) del número de matrículas. Cabe destacar que al considerar un horizonte de planificación mayor, se deben considerar varios elementos que no se conocen con exactitud al momento de tomar las decisiones, como por ejemplo, la cantidad de estudiantes que ingresarán a primer año, así como, determinar cuántos estudiantes se inscribirán en los distintos cursos que se imparten en una malla curricular. Por tanto, es evidente que para poder pronosticar de buena manera el número de "horas-asiento-profesor", antes debemos comprender cómo es el paso de los estudiantes por las mallas curriculares de cada carrera.

De acuerdo con lo anterior, el presente trabajo de tesis busca dar direcciones, en el corto y largo plazo, para la mejor utilización de la infraestructura y staff de profesores en las universidades. En el corto plazo se busca aportar en el desarrollo de sistemas y modelos de optimización basados en patrones que permitan resolver el problema de programación de horarios y asignación de salas de clase en universidades. Mientras que en el largo plazo, se propone un enfoque de solución que determine la planificación de la capacidad para hacer frente a cambios en las matrículas. Adicionalmente, en esta tesis se proponen diferentes áreas de trabajo futuro, extensiones y potenciales aplicaciones en otras áreas disciplinarias.

1.2. Estructura de la tesis

El contenido de esta tesis se encuentra dividido en dos partes. Las parte I presenta el estudio del problema de programación de horarios y salas de clase en el corto plazo. Además, en esta parte se presentan los sistemas de soporte desarrollados para dos entidades académdicas. Luego, la parte II presenta un enfoque de solución basado en modelos de optimización para resolver el problema de planificación de la capacidad en una universidad. A continuación se presenta un breve resumen de cada una de las partes.

Parte I.A: Programación de horarios y asignación de salas de clase de una escuela de postgrado [112].

Este parte presenta enfoque de solución basado en un modelo de programación entera mixta que permite resolver el problema de la programación de horarios y asignación de salas de clase para una Escuela de Postgrado. Particularmente, en esta parte se incoporán tres nuevos elementos respecto de la literatura existente. En primer lugar, se incorpora que los cursos tienen distinta duración dentro del horizonte de planificación, por ejemplo el curso 1 podría durar 15 semanas, mientras que el curso 2 podría durar 30 semanas de clase. Segundo, se incorpora dentro del problema que las fechas de inicio de los cursos son variables en el tiempo, siendo repartidas durante todo el horizonte de planificación académico y, por tanto, en una semana particular pueden comenzar o terminar diferentes cursos. Finalmente, en tercer lugar se incorpora que las fechas de inicio de los cursos deben estar contenidas dentro de una ventana temporal de dos o mas semanas de duración. Es importante destacar que en este escenario se debe considerar la programación de horarios y asignación de salas de clase para el horizonte de planificación completo, no siendo posible replicar una semana tipo para todas las semanas.

El enfoque de solución propuesto utiliza variables de decisión que se basan en asignar a cada curso a un único patrón. En este caso el patrón está compuesto por tres componentes: 1) un bloque horario con dos días diferentes; 2) un período consecutivo de semanas en la que se dictarán las sesiones de los cursos y; 3) una sala de clase adecuada para las sesiones de un curso. Por ejemplo, si el curso c se asigna al patrón $(LJ, 6-25, F3)$ implica que las sesiones del curso son programadas los días Lunes y Jueves, entre la semana 6 y la 25 y utilizará la sala de clases $F3$. La función objetivo del modelo minimiza los costos totales de la programación de cursos, siendo posible mencionar por ejemplo los siguientes términos: 1) costos de arriendo de salas de clase externas; 2) costos de oportunidad por tener asientos vacíos (capacidad ociosa) y; 3) costos de oportunidad al existir conflictos horarios entre cursos.

Este enfoque fue aplicado al problema de una Escuela de Postgrado obteniendo disminuciones significativas en los costos totales de la operación. Por ejemplo, fue posible disminuir en un 30 % los costos de arriendo de salas de clase. Además, fue posible liberar algunas salas de clase completas siendo posible asignarlas para otro tipo de actividades académicas, así como, se disminuyó un 25 % en promedio de los conflictos horarios entre cursos que debían ser cursados por un mismo conjunto de estudiantes. Estas mejoras fueron obtenidas respecto de un enfoque manual basado en la prueba y el error.

El artículo elaborado en esta parte se encuentra disponible en la siguiente dirección web:
http://interfaces.journal.informs.org/content/40/3/196.refs

Parte I.B: Programación de horarios y asignación de salas de clase en una escuela de pregrado [114].

En esta parte se presenta un enfoque de solución que permite resolver de manera eficiente la programación de horarios y asignación de salas de una Escuela de Pregrado. El enfoque propuesto contempla un modelo de programación lineal entera mixta que incorporá varios de los objetivos que persiguen los directivos de este tipo de Escuela, así como, las principales restricciones y condiciones que se deben cumplir para el normal funcionamiento de la institución académica. El enfoque propuesto programa los cursos y asigna las salas de clase considerando una semana tipo la cual se replica para todas las semanas dentro del horizonte de planificación académica.

Las variables de decisión utilizadas en esta parte se basan en asignar cada curso a un único patrón. En este caso el patrón está compuesto por dos componentes: 1) una combinación de uno o más bloques horarios y 2) una o más salas de clase para realizar las actividades del curso. En este caso se incoporó un elemento nuevo respecto de la literatura existente relacionado a que las sesiones de un curso pueden requerir diferentes tipos de salas de clase. Por ejemplo, si el curso c se asigna al patrón *(M.1-V.1,F2-Lab1)* implica que la primera sesión será programada el día Martes en el bloque 1 en la sala F2, mientras que la segunda sesión será programada el Viernes en el bloque 1 en una sala tipo Laboratorio 1. La función objetivo del modelo minimiza una serie de condiciones no deseadas por los directores de este tipo de unidad académica, siendo posible mencionar por ejemplo: 1) el uso de salas de clase especiales como auditorios; 2) que cierto tipo de sesión (por ejemplo tutoriales) se programen fuera de un día deseado; 3) que se programen sesiones de un curso en salas de clase externas no pertenecientes a la Escuela; 4) evitar los conflictos horarios entre cursos de un mismo semestre y de semestres consecutivos; o 5) que se maximice las preferencias horarias de los profesores para dictar sus cursos.

El enfoque propuesto fue aplicado al problema de una Escuela de Pregrado obteniendo reducciones significativas de las condiciones no deseadas por los directores. Dos de los principales logros obtenidos en este caso fueron que por primera vez no se utilizaron salas de clase externas a la Escuela y se evitó completamente la utilización de auditorios en la programación. Estas mejoras fueron obtenidas respecto de un enfoque manual basado en la prueba y el error.

El artículo elaborado en esta parte se encuentra disponible en la siguiente dirección web:

http://www.sciencedirect.com/science/article/pii/S0167923611001746

Parte II: Planificación de la capacidad en una universidad

En esta parte se presenta un enfoque de solución basado en un modelo de programación lineal entera mixta que permite resolver el problema de la planificación de la capacidad para una universidad o facultad. Específicamente, este enfoque permite dimensionar las necesidades de infraestructura, en términos de número y tipo de salas de clase arrendadas, remodeladas y construídas, así como, la dotación de profesores que son necesarios para dictar los cursos de las distintas línea disciplinaria en el tiempo.

Las variables de decisión del enfoque propuesto permiten asignar a cada curso a uno o más tipos de profesores y a un único patrón horario y patrón de salas de clase. Un patrón horario está compuesto por una combinación de bloques horarios (por ejemplo L.1-J.1) y un patrón de salas de clase está compuesto por un conjunto de salas de clase cuya suma de sus capacidades es mayor o igual al número total de estudiantes inscritos en el curso. La función objetivo del modelo propuesto minimiza los costos totales asociados a la planificación de la capacidad en el largo plazo, siendo posible mencionar por ejemplo: 1) costos de construcción; 2) costos de arriendo; 3) costos de remodelación; 4) costos de mantención; o 5) costos de contratación de profesores. Respecto de las restricciones del modelo fueron consideradas variadas condiciones respecto de la carga docente de los profesores, mantener ciertas proporciones entre tipos de profesores, respetar los requerimientos de infraestructura y tipos de profesores de los cursos y evitar los conflictos horarios, entre otras cosas.

El enfoque propuesto contempla el uso de dos modelos de pronósticos para determinar los parámetros del modelo de optimización. El primer modelo permite estimar el número de estudiantes que se inscribirán en cada malla curricular al inicio del horizonte de planificación: Mientras que el segundo modelo permite estimar el paso de los estudiantes por cada malla curricular, específicamente determina el número de estudiantes que se inscribirán en cada curso de una malla curricular. Ambos modelos utilizan información de diferentes series de tiempo, como por ejemplo: número de matriculados, puntajes de corte para pruebas de adminisión, número de cupos, entre otras.

Nuestro enfoque fue aplicado a la Facultad de Economía y Negocios de la Universidad de Chile como herramienta para poder llevar a cabo el desarrollo sustentable en el tiempo de esta unidad académica. En este caso, fue posible estimar las variaciones de costos y construcción de nueva infraestructura ante diferentes esquemas de disponibilidades horarias de los profesores.

1.3. Elementos de los problemas abordados en esta tesis

A continuación, se detallan los principales elementos que fueron incorporados en esta tesis. En la sección 1.3.1 se presenta cómo fue modelada una malla curricular y los cursos que la componen. La sección 1.3.2 se presentan las características relevantes de la infraestructura y salas de clase. Finalmente, la sección 1.3.3 describe las caractrerísticas de los profesores.

1.3.1 Mallas curriculares y cursos

La demanda por "horas-asiento-profesor" está asociada a las actividades de los cursos de cada malla curricular. Una malla curricular se compone de una serie de cursos interconectados conducentes a un título profesional o grado académico. Esta malla determina qué cursos deben ser inscritos de manera simultánea por un conjunto de estudiantes en un semestre particular y, además permite saber cuáles son los requisitos de cada curso. Cabe destacar que en este trabajo asumiremos que la malla no sufre cambios sustanciales dentro del período de planificación.

La Figura 1.1 muestra una red que representa una malla curricular tipo. Los nodos I y F son dos nodos ficticios que representan el inicio y término de la malla curricular, respectivamente. Los arcos de esta red representan los requisitos de cada cursos siendo posible clasificarlos como requisito *core* o *no core*. Un requisito *core* de un curso c corresponde al curso antecesor dentro de una misma línea disciplinaria, por tanto sus contenidos son la base central del curso sucesor. Mientras que un requisito *no core* también son cursos antecesores de c, sin embargo no son de la mísma línea disciplinaria entregando contenidos complementarios. En la Figura 1.1 los requisitos *core* de un curso están marcados con línea continua y los *no core* con una línea punteada.

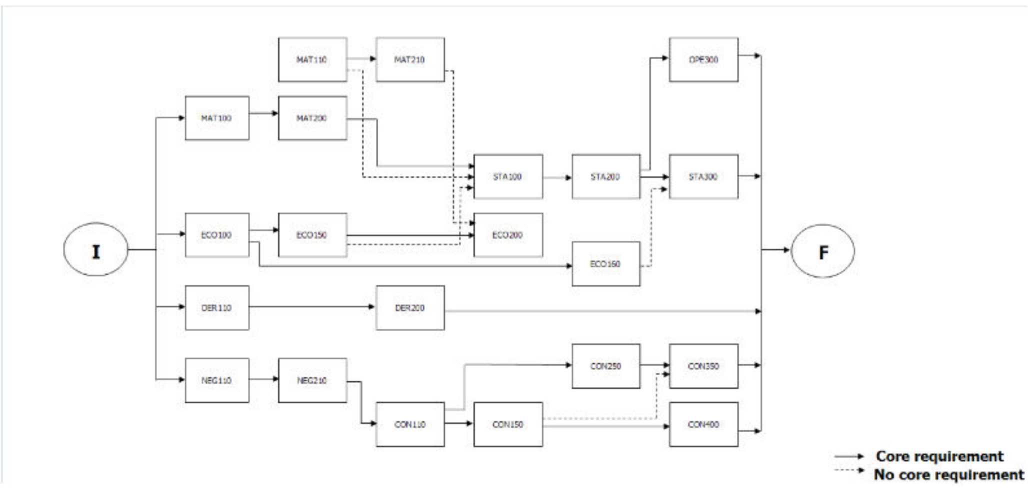


Figura 1.1: Estructura de una malla curricular basada en requisitos.

Cada curso puede tener dos o más secciones. En general cuando un curso tiene un gran número de estudiantes inscritos, se debe dividir el curso en una o más secciones de menor tamaño. Además, se puede considerar que cada curso contempla diferentes tipos de sesiones, tales como de cátedra, tutoriales o seminarios, las cuales no pueden ser programadas en un mismo horario al ser inscritos por un mismo conjunto de estudiantes. Sin embargo, en casos especiales se puede imponer que todas las sesiones de un cierto tipo para un determinado curso deban ser programadas en un mismo horario. Por ejemplo, se puede considerar que las sesiones tutoriales de todas las secciones de un curso deban ser programados en un mismo horario, con el objetivo de utilizar ese horario para realizar evaluaciones comunes. Sin embargo, también se puede considerar que para ciertos cursos se imponga un máximo de sesiones de un mismo tipo en un módulo horario particular. Esta condición permite tener mayor flexibilidad en los horarios que puede inscribir un estudiante al ofrecerle un mayor número de opciones. Además, los cursos pueden ser clasificados en varias categorías, como por ejemplo: 1) área disciplinaria (marketing, operaciones, recursos humanos, etc.); 2) tipo de salas de clase que utiliza para dictar sus sesiones (laboratorio computacional, salas planas, auditorios, etc.); 3) número de horas pedagógicas semanales; o 4) por su importancia dentro de una malla curricular (cursos *core* o *no core*). Estas características implícitamente determinan la demanda de requerimientos de recursos.

1.3.2 Características de las salas de clase

Asumiremos que la universidad tiene un conjunto de salas de clase las cuales pueden ser clasificadas en diferentes tipos de acuerdo con sus características, como por ejemplo: tamaño y forma. El tamaño de una sala de clase se determina por el número de asientos disponibles, mientras que la forma se refiere a la disposición y movilidad de los asientos o por su equipamiento tecnologíco. Por tanto, es posible asignarle a cada curso un tipo de sala de clase más apropiada de acuerdo con las características de las actividades que deben realizarse en cada sesión y el número de estudiantes inscritos en cada curso.

Cabe destacar que no existe solamente una única asignación de un curso a una sala de clase tanto en forma, como en tamaño, siendo posible definir ciertas preferencias. Por ejemplo, si un curso puede ser asignado a una sala de clase de tamaño pequeño, también puede ser asignado sin problemas a una sala de clase de tamaño mediano o grande. No obstante, en este último caso existiría un mayor número de asientos vacíos, lo que en general es una asignación poco deseable.

Respecto del ajuste de los requerimientos de infraestructura en el tiempo podemos considerar la opción de remodelación de las salas de clase existentes en el tiempo. Por ejemplo, una sala de clase de tamaño grande podría ser remodelada en dos salas de clase de tamaño pequeño o en un laboratorio computacional o viceversa. Además, si la demanda de requerimientos “horas-asiento"no son posibles de satisfacer con las salas de clase existentes, podemos considerar el arriendo o la construcción de nuevas salas de clase. En el caso de construir un nuevo edificio, se puede considerar que la construcción se realice en una o varias fases. Mientras que si se desea arrendar es importante considerar que los arriendos de salas de clase pueden también estar limitados semestre a semestre.

Respecto de los costos, se puede considerar que para el caso de construir un nuevo edificio, existe un costo fijo de construcción más un costo variable asociado al número y tipo de salas de clases nuevas que se desee construir. En general podemos asumir que el costo de construcción de una sala nueva es superior al costo de remodelación de una sala de clase ya existente y varias veces mayor al costo de arriendo de una sala de clase extra. Además, al ir aumentando la cantidad de salas de clase hay que considerar el costo asociado al mantenimiento como por ejemplo aseo o servicios básicos.

1.3.3 Características del staff de profesores

Consideraremos que la universidad cuenta con un conjunto de profesores que pueden ser categorizados en diferentes tipos. Por ejemplo, un profesor se puede caracterizar de acuerdo a su jornada laboral (Full-time o Part-time), por el área disciplinaria de los cursos que dicta (marketing, operaciones, finanzas, etc.) o por su jerarquía académica (asistente, asociado, titular). Generalmente, los profesores desarrollan diferentes actividades relacionadas con la investigación, docencia y extensión. Específicamente en este trabajo, solamente se considera la carga de trabajo asociada a la docencia la cual puede estar limitada para cada tipo de profesor. El número máximo de cursos que puede dictar un tipo de profesor se relaciona a que cierto tipo de profesor puede tener asignado una mayor carga docente, mientras que otros pueden tener una mayor carga asociada a la investigación u otras actividades académicas, por lo cual dictan un menor número de cursos. Además, dependiendo de la importancia de un curso dentro de la malla curricular, podemos asumir que solo pueden ser dictados por un cierto tipo de profesor.

Otro aspecto importante son las disponibilidades horarias que tienen los diferentes tipos de profesores para dictar los cursos. Estas disponibilidades determinan las franjas horarias en donde se pueden programarán ciertos cursos, observándose en la práctica que ciertas franjas horarias son preferidas respecto de otras.

1.4. Objetivos de la tesis

1.4.1 Objetivo general

El objetivo general de esta tesis es:

Aportar en el desarrollo de enfoques de solución basados en modelos de programación lineal entera que permitan resolver en el corto plazo el problema de programación de horarios y asignación de salas de clase y, en el largo plazo, la planificación de la capacidad en universidades.

1.4.2 Objetivos específicos

Los objetivos específicos de esta tesis son:

1. Desarrollar modelos de programación lineal entera basados en variables que asignen los cursos a patrones para resolver óptimamente el problema de programación de horarios y asignación de salas de clase en universidades.
2. Desarrollar modelos de programación lineal entera basados en variables que asignen los cursos a patrones para resolver óptimamente el problema de planificar la capacidad en una universidad, en términos del número y tipo de salas de clase y profesores.
3. Cuantificar los beneficios de aplicar los modelos de programación lineal entera propuestos desarrollados en aplicaciones reales.
4. Proponer nuevas líneas de trabajo futuro en ámbitos similares, pero considerando diferentes objetivos y estructura del problema.

1.5. Revisión de la literatura relevante

1.5.1 Programación de horarios y asignación de salas de clase

El problema de programación de horarios no es un problema nuevo existiendo una amplia literatura en investigación de operaciones al respecto [62, 138, 43, 100, 23, 103, 41]. Este tipo de problema es un caso especial de un conjunto más general llamados problemas de programación o *Scheduling* [98], los cuales buscan asignar eventos o tareas a diferentes recursos y bloques horarios. En la práctica, se han presentado diferentes variantes de este tipo de problema para ruteo de vehículos [95, 72], programación de ligas deportivas [70, 94], programación de máquinas [59, 104], programación de personal [33] y *Rostering* [74], entre otros.

En su forma más simple, la programación horaria de cursos asigna cada curso a un bloque horario y sala de clase considerando una serie de requerimientos y restricciones que permiten el normal funcionamiento de estas instituciones de educación superior [43]. Dada la naturaleza combinatorial del problema, es de difícil resolución, por lo que los enfoques manuales basados en la prueba y error son ineficientes en escenarios caracterizados por una enorme cantidad de cursos y salas de clase. Por ejemplo, en [47] muestran que cuando se impone que los cursos deben mantener la misma sala de clase para todas las sesiones, el problema se transforma en *NP-Hard*.

Uno de los enfoques de solución más comúnmente utilizados para modelar el problema se basa en modelos de programación lineal entera [150, 60, 112]. Dentro de este contexto, el trabajo presentado por [62] es uno de los primeros trabajos seminales de esta área, el cual muestra una serie de modelos de optimización lineal que han servido de inspiración para una serie de aplicaciones. Un conjunto de enfoques de solución que se basan en modelos de programación lineal entera se presentan en los siguientes trabajos [61, 60, 66, 115], los cuales muestran los beneficios de la aplicación de este tipo de modelo en diversas universidades.

Cabe destacar que además de los métodos exactos, se han propuesto diversos enfoques heurísticos para resolver el problemas de programación de horarios. Este tipo de enfoque busca aproximar la solución óptima del problema mediante diferentes procedimientos. Los enfoques de solución basados en heurísticas pueden ser divididos en 3 tipos [43]. El primer tipo son las *Heurísticas Secuenciales* las cuales asignan cada curso en un bloque horario válido de manera secuencial, tal que en el mismo bloque, no existan conflictos horarios con otros cursos que deban ser inscritos por un mismo grupo de estudiantes [47]. Este tipo de enfoque se ha formulado como un problema de coloreo de grafos, en que los cursos serán representados como vértices y los conflictos horarios serán representados mediante los arcos [49]. Esta idea fue extendida por [42], quienes presentan una metaherística que resuelve el problema de coloreo de grafos utilizando la técnica *Tabu Search*. El segundo tipo son las *Heurísticas de Cluster* que contemplan dos etapas secuenciales. En una primera etapa particionan el conjunto de cursos en grupos que satisfacen las restricciones duras del problema y, en una segunda etapa, se programan los cursos en los diferentes bloques horarios tratando de satisfacer las restricciones blandas [163]. Cabe destacar que una debilidad de este tipo de enfoque es que los grupos de cursos son fijados al comienzo del procedimiento lo que puede provocar que la programación horaria sea de baja calidad [43]. Finalmente, el tercer tipo son las *Metaheurísticas* las cuales han sido ampliamente utilizadas para resolver problemas de programación horaria de cursos. Entre los principales desarrollos podemos mencionar por ejemplo: *Simulated Annealing* [3, 111, 165], *Tabu Search* [162, 7, 8, 103], algoritmos genéticos [160, 122], redes neuronales artificiales [46], *honey-bee mating* [117] y, algunos enfoques híbridos que mezclan dos metaheurísticas particulares [52]. En general, las metaheurísticas comienzan con una o mas soluciones factibles iniciales del problema y, luego mediante diferentes estrategias de búsqueda, tratan de mejorar la calidad de la solución evitando caer en mínimos locales. Las metaheurísticas pueden encontrar soluciones de buena calidad, sin embargo, a menudo tienen un considerable costo computacional y no permiten saber qué tan lejos se encuentra de la solución óptima del problema original [100].

En las instancias analizadas en esta tesis fue posible resolver los problemas de optimización, utilizando métodos exactos. Cabe destacar que los modelos de optimización propuestos fueron implementados en sistemas computacionales que entregan la programación de horarios y la asignación de salas de clase óptmimas en diferentes entidades educativas. El uso de variables basados en patrones hace posible diseñar y aplicar filtros que permiten descartar asignaciones infactibles a priori permitiendo reducir la dimensión del problema fácilmente. A pesar de existir abundante literatura respecto de la programación de horarios en universidades, en la práctica un número reducido de estos trabajos han sido implementados como sistemas de soporte a la toma de decisiones (DSS). Una serie de factores pueden explicar la escasez de este tipo de sistemas en el mundo real, como por ejemplo: la resistencia al cambio en la organización y la adopción de nuevas tecnologías [57], falta de compromiso del usuario [82], las percepciones negativas de la "facilidad de uso de sistemas"[136], el grado de dificultad del problema soportado por el *DSS* [156, 84], la necesidad de contar con equipo multidisciplinario de trabajo para desarrollar las tareas relacionadas a la construcción del sistema [14, 82, 137], los niveles de la experiencia requeridos de los usuarios [129, 135], la formación profesional [156] y la falta de participación del usuario en el desarrollo del sistema [82, 137].

Modelos básicos para la programación de cursos

El problema de programación de horarios y asignación de salas de clase no es un problema nuevo, existiendo una amplia literatura en Investigación de Operaciones al respecto presentándose varias formulaciones [62, 138, 100, 133, 41]. Uno de los primeros modelos de optimización dentro de esta línea lo presenta [62]. Este modelo determina la programación de horarios de C cursos con k_c sesiones cada uno $(c = 1, ..., C)$ y, además considera dentro de la programación que varios subconjuntos de cursos no deben ser programados en los mismos bloques horarios. Se define como $h \in H$ al subconjunto de cursos que no deben ser programados en un mismo bloque horario. Esta condición se asocia a que un mismo grupo de estudiantes debe inscribir de forma simultánea todos los cursos que contiene el subconjunto h. En este modelo el número total de bloques horarios es P y, se define el parámetro l_p, como el número máximo de sesiones que pueden ser programadas en el bloque horario $p = 1, ..., P$.

En consecuencia, el modelo de optimización es formulado de la siguiente manera:

$$\text{máx}\, z = \sum_{c=1}^{C}\sum_{p=1}^{P} \mathrm{d}_{cp} y_{cp}, \tag{1.1}$$

$$s.a.$$

$$\sum_{p=1}^{P} y_{cp} = k_c \quad \forall\ c = 1, ..., C, \tag{1.2}$$

$$\sum_{c=1}^{C} y_{cp} \leq l_p \quad \forall\ p = 1, ..., P, \tag{1.3}$$

$$\sum_{c \in h} y_{cp} \leq 1 \quad \forall\ h \in H; p = 1, ..., P, \tag{1.4}$$

$$y_{cp} \in\ \{0, 1\} \quad \forall\ c = 1, ..., C; p = 1, ..., P, \tag{1.5}$$

en donde, y_{cp} es una variable binaria que toma valor igual a 1 si una sesión el curso c es programada en el bloque horario p y, un valor igual a 0 en caso contrario. La función objetivo del modelo (1.1) maximiza que los cursos sean programados en los bloques horarios deseados. El parámetro d_{cp} representa que tan deseado es que una sesión del curso c sea programada en el bloque horario p. La restricción (1.2) asegura que cada curso tenga programadas todas sus sesiones. La restricción (1.3) impone que en cada bloque horario no se programen más sesiones que el número total de salas de clase disponibles en el bloque horario p (l_p). La restricción (1.4) evita completamente los conflictos horarios entre cursos que deben ser inscritos por un mismo grupo de estudiantes dentro de un mismo período académico.

Ahora bien, si consideramos el caso en que no es posible evitar todos los conflictos horarios definidos por los subconjuntos de cursos $h \in H$, podemos definir un nuevo término en la función objetivo que minimice el número total de conflictos. Por ejemplo, [158] minimiza el número total de estudiantes que poseen un conflicto horario, por tanto se busca evitar los conflictos horarios de los cursos más numerosos.

Algunos autores han extendido esta línea de investigación considerado que las restricciones del modelo (1.1)-(1.5) pueden ser duras o blandas [43]. Las restricciones duras siempre se deben satisfacer, por ende, definen el espacio de soluciones factibles del problema. Algunos ejemplos de este tipo de restricciones son que todas las sesiones de un curso sean programadas o que en un bloque horario particular no se programen más sesiones que el número de salas de clase disponibles. Mientras que las restricciones blandas, son condiciones deseables, pero no absolutamente esenciales para el normal funcionamiento de la universidad. La violación de estas condiciones se incorpora como nuevos términos en la función objetivo.

A continuación, discutiremos brevemente algunas variantes que se han presentado en la literatura:

1. **Múltipes secciones de un mismo curso**. Este caso considera cuando se deben programar cursos que poseen varias secciones de manera simultánea. Un curso debe ser dividido en varias secciones cuando el número total de estudiantes inscritos no puede ser asignado a ninguna sala de clase por no contar con la capacidad suficiente o cuando existe alguna limitante de carácter pedagógico que limita el máximo de estudiantes por sección. Cuando un curso tiene varias secciones permite disminuir el número de conflictos horarios, al contar con más opciones para los estudiantes para inscribir una sección del curso. Por ejemplo, si el subconjunto h_1 contiene el curso c_1 y c_2 y el subconjunto h_2 contiene los cursos c_1 y c_3, cuando se programe el curso c_2 en el bloque horario p y el curso c_3 en el bloque horario q, el curso c_1 no debe ser programado ni en el bloque horario p ni q. No obstante, si el curso c_1 tiene dos secciones, una de las secciones puede ser programada en el bloque horario p y la otra en el bloque horario q [139]. La restricción (1.6) permite calcular el número de conflictos horarios, dentro de cada subconjunto de cursos h en cada bloque horario p, mediante la variable entera no negativa z_{hp}.

$$\sum_{c \in h} y_{cp} \leq 1 + z_{hp} \quad \forall\ h \in H; p = 1, ..., P. \tag{1.6}$$

2. **Preasignaciones de cursos**. En esta variante se imponen como restricción dura que un curso se programe en un bloque horario específico. La restriccón (1.7) impone que un curso c sea programado en el bloque horario p cuando el parámetro $t_{cp} = 1$. En caso contrario, la restricción (1.8) impone que el curso c no pueda estar programado en el bloque horario q cuando el parámetro $r_{cq} = 1$.

$$y_{cp} \geq t_{cp} \tag{1.7}$$

$$y_{cq} \leq (1 - r_{cq}) \tag{1.8}$$

3. **Cursos con distinta duración**. Esta extensión del problema original considera que un curso puede tener 1, 2 o 3 bloques horarios de duración durante una semana [76]. Por ejemplo, si consideramos que el bloque horario en que comienza el curso c_1 es p y para el curso c_2 es $q > p$, existirá un conflicto horario entre ambos cursos si se cumple la condición $q - p < g_{c_1}$, en donde, g_{c_1} es la duración del curso c_1.
4. **El problema de asignar solo salas de clase**. Esta variante considera solo el problema de asignar un conjunto de salas de clase con diferentes capacidades a cursos con horarios previamente definidos. Uno de los primeros trabajos que aborda esta variante lo presentan [50], quienes muestran que el problema es *NP-Hard* cuando se impone como restricción dura que todas las sesiones de un mismo curso sean asignadas a una misma sala de clase. Cabe destacar que existen problemas similares en otras áreas en donde también se deben programar actividades o recursos como por ejemplo como la asignación de choferes y flotas a viajes.
5. **El problema de asignar estudiantes a secciones**. Otra extensión al problema original busca resolver el problema de asignar estudiantes a cursos con múltiples

secciones, considerando que los horarios de los cursos están fijos. En este caso, se busca minimizar los conflictos horarios respetando: 1) la selección de cursos hecha por los estudiantes, 2) manteniéndo el número de estudiantes total por cada sección de forma balanceada y, 3) respetándo el número máximo de estudiantes por sección. En [7] incluyen otras condiciones especiales, como por ejemplo: respetar las preferencias de los estudiantes respecto del lenguaje (español y catalán) en que se dictan los cursos, evitar las ventanas horarias entre cursos que debe tomar un mismo estudiante o minimizar la distancia total recorrida por los estudiantes cuando las salas de clase de sus cursos están dispersas en diferentes edificios. En [65] se incluye que los estudiantes tengan una carga de trabajo similar y que se incorporen las preferencias de los estudiantes por ciertos cursos. En este tipo de modelos utilizan variables binarias x_{ci} que toman valor igual a 1 si el curso c es asignado al estudiante i y toman un valor igual a 0 en caso contrario.

Cabe destacar, que en todos los modelos de optimización presentados anteriormente, la variable de decisión principal asigna de manera independiente cada sesión de un curso a un bloque horario y sala de clase particular y, mediante el uso de restricciones, imponen las configuraciones horarias deseadas. Mientras que los modelos de optimización presentados en este trabajo de tesis, poseen otro tipo de variable de decisión la cual asigna todas las sesiones de un curso simultáneamente a un solo patrón horario. Este patrón horario es un vector que está conformado por dos o más componentes. Por ejemplo, si el curso c está asignado al patrón (L.A-J.A,S1), significa que las sesiones del curso c se dictan los *Lunes* y *Jueves* en el bloque horario A y que la sala de clase asignada será la *S1*. La principal ventaja de utilizar un modelo de optimización basado en patrones es que la relajación lineal es más ajustada (tighter) [125] facilitando la resolución del modelo al utilizar herramientas comerciales como *CPLEX* [91]. El hecho que las relajaciones lineales sean más ajustadas se basa en que para cada solución de la relajación lineal del modelo basado en patrones, es posible construir una solución equivalente para el modelo que asigna las sesiones individuales. Sin embargo, esta relación no es cierta en el otro sentido, ya que existen soluciones factibles de la relajación del modelo que asigna sesiones individuales para las cuales no hay soluciones equivalentes para el modelo basado en patrones. Si una solución de la relajación del modelo basado en patrones tiene la variable que asigna el curso c al patrón (*Lunes.A - Jueves.A, S1*) con un valor igual a 0.7, se puede construir una solución equivalente para la solución de la relajación del modelo de clases individuales asignado el valor 0.7 a las dos variables que asignan a cada una de las sesiones individuales del curso. Es decir, se le asigna el valor 0.7 a la variable que asigna el curso c al bloque (*Lunes.A)* y sala de clase *S1* y 0.7 a la variable que asigna el curso c al bloque (*Jueves.A)* y sala de clase *S1*. En caso contrario, si consideramos otra solución fraccionaria para la relajación lineal del modelo que utiliza sesiones individuales, con la variable que asigna el curso c al bloque (*Lunes.A)* y sala de clase *S1* igual a 0.7 y la variable que asigna al curso c al bloque (*Jueves.A)* y sala de clase *S1* con un valor igual a 0.2, no es posible encontrar una solución equivalente en la relajación lineal del modelo basado en patrones.

1.5.2 Planificación de la capacidad en universidades

La planificación de la capacidad tiene una importancia estratégica para las universidades, pues comprometen una cantidad sustancial de recursos y permite direccionar los esfuerzos y el crecimiento de largo plazo. Debido al carácter estratégico de las decisiones, si éstas son mal tomadas afectarán de forma negativa y transversalmente a toda la institución educativa. Por un lado, si la planificación de la capacidad está muy por debajo de los requerimientos de las mallas curriculares, disminuirá considerablemente la calidad del servicio, no habrá espacio disponible para programar ciertos cursos, ni tampoco profesores idóneos para dictar sus sesiones. Este escenario, puede derivar en el arriendo de salas de clase externas, la programación de cursos en salas de clase no adecuadas, utilizar módulos horarios no deseados, o que los cursos centrales no se dicten por especialistas. Por otro lado, si planificamos la capacidad por sobre la demanda de requerimientos, ocasionará un aumento considerable en los costos de inversión, costos de operación y mantenimiento de la infraestructura, y posiblemente un aumento innecesario de las contrataciones de nuevos profesores. Estas situaciones, como es evidente, provocarán un descontento general en los estudiantes, los profesores y directivos, afectando negativamente la imagen de estas entidades educativas.

La planificación de la capacidad contempla decisiones en diferentes ámbitos:

- **Decisiones asociadas a la infraestructura.** Dentro de este ámbito se deben tomar una serie de decisiones, como:
  - *Remodelación de la infraestructura existente.* En este ámbito se debe determinar si es necesario remodelar la infraestructura ya existente, y decidir qué salas de clases serán remodeladas para adaptarla a los nuevos requerimientos de los cursos. En muchas ocasiones los espacios no fueron diseñados originalmente para ser salas de clase o en su defecto las salas de clase construídas no fueron diseñadas para nuevos requerimientos pedagógicos.
  - *Arriendo de infraestructura externa.* En ciertos casos es posible arrendar a otras unidades académicas salas de clase para hacer frente a aumentos inesperados de la demanda de cursos.
  - *Construcción de nueva infraestructura.* En este ámbito se debe determinar si es necesario construir un nuevo edificio o campus cuando la demanda por espacio crece demasiado. Dependiendo del tamaño de la nueva infraestructura puede ser construído en una o más fases durante el horizonte de planificación. Además se debe determinar el número y tipo de salas de clase que contemplará este nuevo edificio, en términos del tamaño, forma y equipamiento tecnológico.
- **Decisiones asociadas al staff de profesores.** En este caso se debe determinar el número y tipo de profesores necesarios para poder dictar los cursos de las diferentes mallas curriculares que imparte la universidad. Es importante mencionar que los requerimientos de profesores en este trabajo están asociados a la carga docente, la cual no necesariamente se alinea con los requerimientos de las áreas de investigación u otras actividades académicas.

Asumiremos que la planificación de la capacidad tiene como objetivo la minimización de una función de costos totales. Esta función objetivo puede contemplar los siguientes términos: 1) Costo total asociado a la remodelación de las salas de clase en el tiempo; 2) Costo total de mantenimiento de la infraestructura construída; 3) Costo total asociado al arriendo de salas de clase externas; 4) Costo fijo total por la construcción de una fase de la nueva infraestructura; 5) Costo total de construir salas de clase nuevas y; 6) costo total asociado a la contratación y honorarios de los profesores.

Cabe destacar que al tratarse de un modelo de largo plazo, los parámetros asociados a la función de costos debe considerar factores de descuento y tasas de interés que permitan calcular el valor presente de los costos asociados a los distintos términos de la función objetivo en el tiempo. Se puede incluir un beneficio al final del período de planificación por la venta de parte de la infraestructura.

Modelos para la planificación de la capacidad

Uno de los primeros trabajos que presenta una revisión de modelos utilizados para planificar capacidad en universidades lo presenta [26]. Los modelos presentados en este trabajo se pueden categorizar en tres grupos: a) modelos de proyección del crecimiento del número de estudiantes; b) modelos de dotación de profesores y; c) modelos de optimización y asignación de recursos.

Los modelos tipo a) tienen como objetivo planificar el crecimiento del número de estudiantes al ingreso de una malla curricular y mientras éstos cursan su especialidad. Uno de los enfoques más utilizados para pronosticar el número total de estudiantes que potencialmente se inscribirán en las mallas curriculares que imparte una universidad en el tiempo son los modelos de series de tiempo. Por ejemplo [55] utiliza la lógica difusa para construir el modelo utilizándo como información de entrada el número histórico de estudiantes que se inscribieron en cada malla curricular, así como otras series históricas asociadas a las pruebas de ingreso. Además, estos modelos buscan estimar cómo es el paso de los estudiantes dentro de una malla curricular para determinar el número total de “horas-asiento-profesor"que se necesitarán dentro del horizonte de planificación. Específicamente, es necesario analizar qué cursos se comparten entre varias mallas curriculares, analizar las reprobaciones históricas de los cursos, así como, las políticas utilizadas para ingresos especiales de estudiantes. Diferentes técnicas han sido utilizadas para pronosticar el flujo de estudiantes dentro de una malla curricular siendo posible mencionar por ejemplo: modelos de simulación [134, 30], modelos de flujo en red [127] o cadenas de Markov [18], entre otras.

Los modelos tipo b) permiten resolver problemas asociados al crecimiento y composición del claustro de profesores, así como, estimar los costos que debe incurrir una universidad en términos de los salarios. Estos modelos han sido utilizados principalmente para asignar recursos a los departamentos acedémicos para su normal funcionamiento. Por ejemplo, [154] presenta un modelo que busca estimar una proyección de los salarios de una facultad incorporando criterios de equidad y restricciones presupuestarias. Este modelo caracteriza a los profesores de acuerdo a su jerarquía, su disciplina, años dentro de la universidad y por las diferencias entre salarios promedios con otros profesores de la misma disciplina. En otro contexto, [71] presentan un modelo de optimización de flujo en red que permite resolver la tarea de asignar la carga docente asociada a los cursos que se deben dictar en una universidad. Cabe destacar que en este modelo no se considera la asignación de salas de clase para los cursos asignados ni tampoco sus horarios. No obstante, con este tipo de enfoque se puede estimar las disciplinas que tienen falta o exceso de profesores, así como distribuir la asignación de recursos entre las distintas unidades académicas respecto de la carga de trabajo generada por los cursos que se deben dictar.

Finalmente, los modelos tipo c) tienen como objetivo modelar la forma en que una universidad es administrada y, por ende, tiene como objetivo asignar oṕtimamente los recursos que tiene disponibles para realizar las actividades académicas que deben ser desarrolladas. [11] presentan un modelo de programacón matemática que permite determinar los niveles óptimos de ingreso de estudiantes de primer año en la Universidad de Yale para asegurar su normal funcionamiento. Este tipo de enfoque también permite proyectar el número de estudiantes que ingresarán a una universidad en el tiempo. [17] proponen un enfoque de solución basado en un modelo de programación lineal que determina el área oṕtima que debe ser asignada a cada unidad acádemica. Esta asignación la realiza considerando las actividades de extensión, docencia e investigación que debe realizar cada unidad. El modelo de optimización contempla varios objetivos cuya importancia relativa fue determinada mediante la técnica *Analytic Hierarchy Process* (AHP). Este trabajo no trata directamente la programación de eventos o cursos dentro de las salas de clase, más bien se resuelve el problema de asignación total de espacio de manera general sin considerar cambios en el tiempo. Según los autores, este tipo de enfoque funciona mejor cuando se trata de asignar el espacio a un solo tipo de usuario (como por ejemplo las oficinas), a diferencia de la asignación de espacio a salas de clase, donde éstas últimas son utilizadas de manera transitoria por distintos perfiles de estudiantes y con un fin asociado a la enseñanza y el aprendizaje. Otro trabajo interesante que trata el problema de la planificación de la infraestructura lo presenta [23]. En este trabajo presentan un enfoque de solución basado en un modelo de programación lineal que permite asignar y remodelar la infraestructura existente teniendo como objetivo aumentar la utilización de las salas de clase. Este enfoque considera la programación de múltiples tipos de eventos, como por ejemplo tutoriales, cátedras o seminarios, considerando que los eventos tienen un horario fijo. En este trabajo se considera que los cursos tiene horarios fijos y no contempla la construcción de nueva infraestructura solo se considera la remodelación de la existente.

Cabe destacar que no se ha encontrado trabajos que determinen simultáneamente la planificación de la capacidad en términos de la infraestructura y profesores, ni tampoco trabajos que incorporen directamente la proyección del número de estudiantes y eventos a programar en el tiempo.

1.6. Metodología de la investigación

La metodología utilizada en este trabajo de tesis fue la siguiente:

1. **Definición de los problemas y recolección de la información relevante**. En esta etapa se realizó una revisión acabada de la literatura existente, tanto el problema de programación de horarios y asignación de salas de clase, como para el problema de planificación de la capacidad en el largo plazo para las universidades. El resultado de esta revisión bibliográfica permitió tener una visión general de los problemas estudiados por otros investigadores haciendo posible identificar las similitudes y diferencias respecto de los enfoques de solución propuestos en esta tesis. Particularmente, se analizaron los objetivos que persiguen los directivos de diferentes unidades académicas de educación superior, en el corto y largo plazo, así como, se identificaron las condiciones obligatorias y deseables que se deben cumplir para el normal funcionamiento de una unidad. Además, en esta etapa se identificaron las fuentes de información relevantes para la aplicación de los enfoques propuestos en tres casos reales. Con esta información se construyó una base de datos con la cual fue posible definir los parámetros de entrada de los modelos de optimización, así como, fue posible crear filtros para descartar soluciones infactibles a priori.
2. **Diseño y desarrollo de los enfoques de solución basados en modelos de programación lineal entera**. En esta etapa se diseñaron los enfoques de solución para una clase amplia de problemas que fueron abordados en esta tesis. En esta etapa se formularon los diferentes modelos matemáticos que permiten resolver, tanto el problema de la programación horaria y asignación de salas de clase en el corto plazo, así como, para resolver el problema de planificación de la capacidad en el largo plazo para una universidad. Para cada caso se definieron los patrones que serían utilizados para las variables de decisión, así como, se formularon las restricciones y objetivos de los modelos de optimización propuestos en esta tesis.
3. **Obtención de las soluciones de los problemas mediante la utilización de los modelos**. En esta etapa se adaptaron los modelos de programación lineal entera formulados en la etapa anterior para aplicarlos sobre 3 casos de unidades académicas. Para la resolución de los modelos de optimización se utilizó algoritmos genéricos contenidos en un paquete comercial, como por ejemplo, *Branch and Bound*. Particularmente, en esta etapa se determinó la programación de horarios y asignación de salas de clase para una instancia particular, así como la planificación de la capacidad, en términos del número de salas de clase y profesores que se necesitan para hacer frente a cambios en demanda de cursos. Además, se cuantificaron los beneficios obtenidos por los enfoques de solución propuestos en diferentes instancias utilizando diversos indicadores de desempeño.
4. **Pruebas y validación de los modelos de programación lineal entera**. En esta etapa se validaron las soluciones entregadas por los modelos de optimización propuestos, en términos de la coherencia con el problema real. Con esta validación se realizaron ajustes a los modelos, al igual que se identificaron potenciales fallas en los programas computacionales que permiten la resolución del problema de optimización.

5. **Implementación de los modelos**. En esta etapa se realizó el diseño e implementación de los sistemas computacionales que resuelven el problema de corto plazo. Específicamente, se detallaron cuáles son los componentes estratégicos y módulos principales de los sistemas implementados.

1.7. Aportes originales de esta tesis

En este trabajo de tesis ha abordado algunos elementos centrales y distintivos los cuales se detallan a continuación:

1. En términos prácticos, se modelaron algunos elementos no presentadas en la literatura previamente. En primer lugar se consideró que los cursos tienen diferente duración y que las fecha de inicio difieren curso a curso. Estos dos elementos generan la necesidad de programar semana a semana los cursos dentro del horizonte de planificación no siendo posible programar una semana tipo replicable. Esta situación genera que eventualmente todas las semanas sean diferentes.
2. En los modelos implementados las variables de decisión principales asignan todas las sesiones de un curso y las salas de clase que utilizan de manera simultáneamente en un solo patrón. Un patrón es una combinación factible de horarios y salas de clase para un curso. La principal ventaja de utilizar un modelo de optimización basado en patrones es que la relajación lineal es más ajustada (tighter) facilitando la resolución del modelo al utilizar herramientas comerciales como *GUROBI* [85] y *CPLEX* [90].
3. A pesar de que el problema de programación de horarios ha sido estudiado extensamente en la literatura, en la práctica, solo un número pequeño de enfoques de solución ha sido implementado como un sistema de soporte a la toma de decisiones. En este trabajo de tesis se debió enfrentar una serie de problemas del mundo real incluyendo la resistencia de la organización a adoptar nuevas teconologías, poco compromiso de la organización, percepciones negativas de la usabilidad del sistema, así como, el tratamiento del grado de dificultad del problema que se estaba soportando. Estos elementos permiten evidenciar la necesidad de conformar equipos multidisciplinarios, contemplar un mayor nivel de experiencia de los usuarios y una activa participación de los mismos, en el desarrollo del sistema computacional.

Cabe destacar que en este trabajo de tesis no se aborda de manera profunda el tratamiento de la incertidumbre del problema de planificación en el largo plazo. Particularmente, se desarrollan algunos modelos de pronósticos que permiten estimar algunos de los parámetro utilizados por los modelos de optimización. Por tanto, una extensión natural de esta tesis podría estar relacionado al tratamiento probabilístico de la incertidumbre o el estudio de diferentes escenarios posibles que permitan reducir la incerteza del resultado de las decisiones en este ámbito.

Como resultado de esta tesis se elaboraron los siguientes artículos científicos:

- **Miranda J.**, Espinoza D., Weber R. (2014) *A MIP Approach for Infrastructure Capacity Planning in Higher Education: The effect of Instructor Timetable Availabilities*. Enviado a la revista *OR Spectrum*.
- **Miranda J.**, Rey P., Robles J.M.(2012). *udpSkeduler: A Web architecture based decision support system for course and classroom scheduling*. Decision Support Systems 52, pp. 505-513. Factor de impacto promedio 5 años 3.035 (actualizado al 22 de abril del 2014).
- **Miranda J**.(2010). *eClasSkeduler: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile*. Interfaces 40(3), pp. 196-207. Factor de impacto promedio 5 años 1.016 (actualizado al 22 de abril del 2014).

Adicionalmente, se presentaron los siguientes trabajos en distintas conferencias y congresos:

- Un Modelo de Optimización para Planificar la Capacidad en una Universidad: Un Caso de Estudio del Efecto de las Disponibilidades Horarias de los Profesores, Congreso Chileno de Investigación de Operaciones X OPTIMA, Concepción, Chile, 2013 [113].
- Un Modelo de Programación Entera Basado en Patrones para la Asignación de Salas de Clases para una Facultad de Medicina, Congreso Latinoamericano de Investigación Operativa (CLAIO), Rio de Janeiro, Brasil, 2012 [114].
- *eClasSkeduler* 2.0: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile. International Federation of Operational Research Societies Annual Meeting-IFORS, Melbourne, Australia, 2011 [112].
- An Integer Programming Approach for Classroom Infrastructure Planning, European Conference on Operational Research-EURO XXVI, Lisboa, Portugal, 2009 [113].

Parte I

Programación de horarios y asignación de salas de clase

Capítulo 2

eClasSkeduler: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile

Abstract

Each October, the Executive Education Unit at the Universidad de Chile develops its course schedules for the following year. By 2008, the complexities of increasing enrollments and course offerings had rendered its manual timetabling process unmanageable. Inconvenient and inflexible scheduling decisions were causing discontent among instructors and students, making the need for a more efficient system of assigning classrooms patent. Three characteristics distinguish the unit's situation from the classic university course timetabling problem. First, its courses vary in duration, ranging between 15 and 30 weeks. Second, its course start dates are spread over the academic year. Finally, each course's start date is flexible and must fall within a window defined by the earliest and latest start dates. This paper presents an automated computational system that generates optimal timetables and classroom assignments for all the unit's courses, minimizing both operating costs and schedule conflicts. When we compared the schedules it generated with the unit's manually generated timetables, we found that our system yielded average cost savings of 35 percent; in addition, it reduced execution times (for generating schedules) from two weeks to less than 30 minutes.

2.1. Introduction and context

The Executive Education Unit (EEU) is a unit of the Universidad de Chile. *EEU*'s mission is to offer study programs in management control, information systems and technology, business administration, and economics. Its courses are aimed largely at professionals, high-level executives, and companies interested in employee training and development. Between 2003 and 2008, 7,000 students attended *EEU* courses. The *EEU* has historically scheduled its courses and assigned classrooms manually by trial and error, a slow and tedious process with average execution times that had ballooned to four weeks by 2008. Execution time is defined as the time needed to complete the final course schedules for an academic year, and involves the following three stages.

1. Collection and entering of the necessary data, which require an average of one week;
2. construction of timetables and generation of reports (ETTR), which require an average of two weeks;
3. validation of schedules by *EEU* administrators, which requires an average of one week.

Since 2003, a significant increase of both enrollments and number of courses taught (Figure 2.1) has added to the complexity of the manual process; the inflexibility of the schedules generated and the inefficient resource allocation of the schedules have made this manual process increasingly unsustainable.

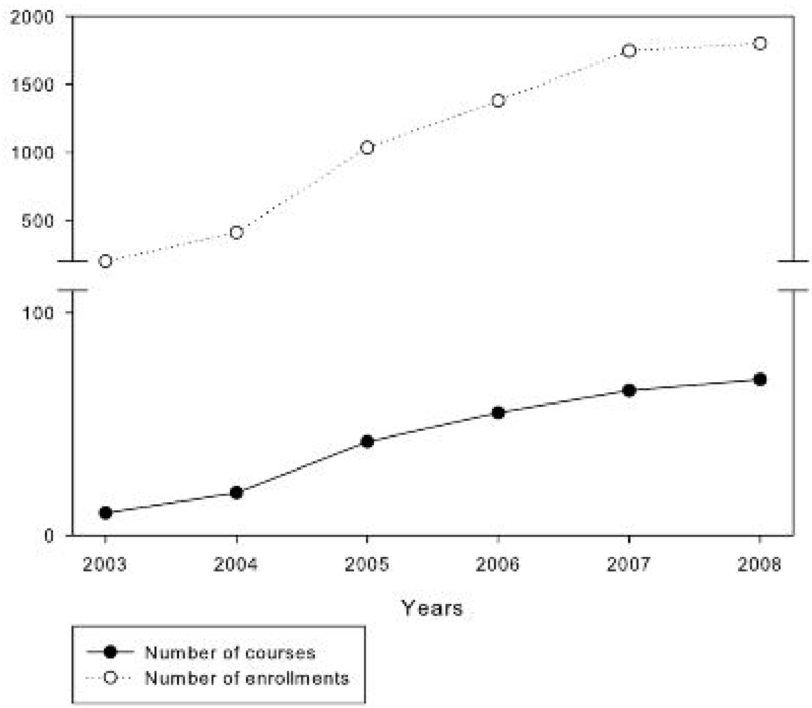


Figura 2.1: Trend in annual enrollments and annual number of courses.

Since 2005, the *EEU* has also provided closed courses-courses requested by specific companies and open only to employees of these companies. These courses, which may be requested at any time during the year without prior notification, are strategically important to the *EEU*, allowing it to build strong relationships with the country's major industries. Thus, when the *EEU* receives such a request, it must quickly update and retimetable (i.e., reschedule) the schedules of all courses that have not yet begun. If the request is for a date on which no classrooms are available and none of the previously allocated courses can be rescheduled, the *EEU* will offer the company a different date that has classroom availability. The growth in demand for these courses has led to a corresponding increase in rescheduled courses (Table 2.1).

Year	Nº Courses	Nº Students
2008	25	600
2007	15	280
2006	17	250
2005	2	50

Tabla 2.1: Trend in annual number of courses and enrollments: *closed* courses for specific companies.

A range of critical organizational problems had also begun to surface under the prevailing system causing much frustration and wasted time for *EEU* administrators, instructors, and students. These problems, which included assignment of classrooms with insufficient capacity, schedule conflicts (overlaps) between courses taught by a specific instructor, frequent changes in classroom assignments, and excessive assignment of special classrooms such as laboratories (labs), were often not detected promptly, and were extremely difficult to resolve.

These problems affected the *EEU* negatively and perceptions of disorganization damaged its image. The timetable conflicts increased operating costs because they necessitated course rescheduling, compensation mechanisms for affected students, and rental of classrooms outside of *EEU* premises. Rates for these unscheduled rentals were 200 percent higher than standard (classroom) rates. The *EEU* also incurred an additional cost of transferring students to the off-premise locations; in addition, if the rental required a lab, the expense of equipping it also had to be added to the cost.

To address these difficulties, the *EEU* embarked on developing *eClasSkeduler*, an automated computational system that generates optimal course timetables and classroom assignments. The objective of this paper is to describe this system and the use of an integer programming (IP) model within it to provide an optimal solution that minimizes operating costs and schedule conflicts. Implementing this system has reduced the time required for the timetable construction and report generation (i.e., ETTR, the second stage), which previously took an average of two weeks and now averages less than 30 minutes. Additional benefits include greater automation of the scheduling process, faster and higher-quality timetabling, the ability to explore multiple scheduling scenarios, and more efficient use and assignment of the *EEU*'s resources.

2.2. Literature review

The twin problems of timetabling and classroom assignment faced by educational institutions have been widely studied in the literature [2, 9, 100, 136, 139]. Both are classified as NP-hard [138]. There exists a range of techniques for solving them, most notably a series of optimization models based primarily on linear and integer linear programming [60, 61, 115] as well as various types of heuristics and metaheuristics [73, 100, 124, 131]. A number of researchers have used specific heuristics such as tabu search, [8], simulated annealing [73], ant colony algorithms [146], genetic algorithms [160], multi-neighborhood local search [80], multi-agent methods [118, 152] and hyperheuristics [40, 42, 128]. Yet other approaches employ hybrid heuristics that combine two or more of the techniques named above, such as [56, 162], or offer designs built around constraint programming [1, 83, 132].

[62] describes a series of problems that arise in the mathematical modelling of course scheduling, instructor assignment and examination timetabling. Though adopting a theoretical perspective, the article lays the basis for developing future applications that address similar problems and actual study cases.

Some researchers have systematized their solution approaches and implemented computational systems that provide support for decision-makers at specific educational institutions. [29] propose a system that uses a heuristic procedure to solve a course scheduling problem. They show that it has low implementation costs and significantly reduces execution times. Whereas the equivalent manual process took three weeks to deliver a solution, their system did so in two days.

Ten years later, [54] implemented a system on a PC to resolve a course scheduling problem at an adult training school. They solve their network-flow model using a heuristic procedure that generates initial solutions in a matter of minutes. [93], implemented a computational course scheduling system at the *Loughborough University Business School*. The paper describing the system, which allows the user to estimate the effects and consequences of a given scheduling solution, shows that efficient management of course databases leads to better-quality schedules. The schedules generated are not necessarily feasible or optimal; however, the authors stress their system's usefulness in permitting greater automation of the processes.

[75] present *SHAPIR*, a system that uses heuristic algorithms to solve an IP model and allows the user to manually adjust the schedules it generates. Three educational institutions of different sizes have tested the system; in each case, the results were good and execution times were reasonable.

[150],developed a system to automate course timetabling at UCLA's Anderson School of Management. In this approach, the courses are separated into two sets, core and noncore. The system uses an *IP* model whose objective function maximizes the schedule preferences of the core-course instructors; it then schedules the noncore courses using a heuristic procedure.

[79] developed *SlotManager*, which has a similar structure to our model, for use with a microcomputer. It has three components-interface, database, and model base. Its interface enables a user to enter information, e.g., type of student group, list of classrooms, and course characteristics, into the system using a series of menus. Its database is relational and contains the information that defines the schedules for a given semester or year. Fi-

nally, the model base contains the system's ïntelligence,"which solves instructor-schedule conflicts, classroom-capacity violations, and resource underuse. [149] originally presented this structure.

[88] created *SchedulExpert*, a system that also incorporates an *IP* model that they implemented for course timetabling at *Cornell University's School of Hotel Administration.* The set of decision variables in the model assigns courses to patterns, which are combinations of class days, time slots, durations, and classroom-equipment requirements. The type of variable used is similar to the type that we used in our formulation. Their model addressed all the schedule-conflict conditions that Cornell imposed; in addition, the time required to generate the timetables dropped from several weeks to a few hours.

Other studies [8, 48, 66, 86, 116, 118] have reported diverse decision support systems that incorporate mathematical programming models for generating course timetables and classroom assignments. In each case, the system contributed numerous improvements to the decision-making process.

2.3. The research problem

In its simplest form, course timetabling is defined as the assignment of a set of courses to different time slots and classrooms while satisfying certain requirements [88]. In this paper, we discuss a model that addresses a course timetabling solution for a university.The course timetabling models may be further broken down into three groups: school timetabling, examination timetabling, and university timetabling. The first group deals with scheduling at the primary and secondary school levels, the second group is used for setting examination times and the third provides timetable solutions for post-secondary establishments. The last of these will concern us here.

Three special characteristics distinguish the *EEU*'s situation from the classic university course timetabling problem. First of all, the institution's courses are of varying duration, ranging anywhere from 15 to 30 weeks of classes. This is because they differ in the depth of treatment of their subject matter, with introductory level courses generally being shorter than the specialization courses, the latter tending to have more classes. Also, the courses have different start dates which are spread over the entire length of the academic year. Finally, each course's start date must fall within a temporary window defined by earliest and latest start dates. Any starting date is flexible, in the sense that it can be any day of previously defined 3-weeks window time. All of this implies that the number of courses starting or ending in any given week will fluctuate. The Unit may schedule a course for any date within the planning period as long as the end date is within it as well. Because of these characteristics, different courses may start and end in any given week. The timetabling problem must therefore be formulated so that the solution schedules the entire academic year simultaneously rather than just a single base week that is then repeated over the entire calendar. Each week will thus have its own timetable.

The problem must also incorporate other requirements and conditions to ensure that the solution is appropriate. Each course consists of two classes, either lecture or lab, per week. A lecture can be given in any classroom; a lab class must be held in a computer lab. The two weekly classes must be held in three-hour time slots on different days between Monday and Saturday; the days must remain fixed throughout the course's duration. Moreover, all classes for a given course must be scheduled over consecutive weeks, leaving no intermediate weeks without a class. The classroom assigned to a course must also be the same for the entire planning period. However, this restriction is relaxed for lab classes. Dates for the latter are defined *a priori* for each course at the start of the planning horizon.

A key aspect of the timetabling requirements is that for any given course, there is a set of *conflict-excluded courses* (i.e., excluded courses) that must not be scheduled on the same day. The courses in the set are those that a student would be most likely to take or an instructor would be most likely to teach during the same academic term. The *EEU* administration determines the excluded course sets and communicates them to the timetable programmers prior to scheduling.

In summary, generating the *EEU* course and classroom timetables is extremely complex. With the growing number of courses and students plus the need to schedule closed courses on an ongoing basis, manual timetabling was no longer viable and its continued use would lead to scheduling errors and inefficiencies in resource allocation.

2.4. Description of the system and optimization model

According to the classification defined by [102], we can consider *eClasSkeduler* as supporting decision making based on mathematical models because it generates an optimal solution that is consistent with a series of constraints. It uses a menu-based interface that allows the user to modify all the necessary parameters (Figure 2.2).

The input *information module* is a database that stores all information relating to courses, classrooms, and instructors. For each course, it includes data items such as the following. 1) Estimate of student enrollment, which is derived from an historical demand forecast; the estimate is adjusted as registration proceeds; 2) course duration; 3) window defined by the start dates; 4) lab class dates; and 5) corresponding excluded courses. The database stores each classroom's capacity, rental cost, and any technological resources that it contains.

The user *interface module* serves as the system control mechanism. It transforms the input data kept in the input information module as specified by the requirements and formats of the *IP* optimization model. It also handles the definition of the weight coefficients that set the relative importance of the different objectives in the objective function. Finally, it allows various technical parameters to be defined, including the maximum number of iterations, the gap between the linear relaxation and the best integer solution, and the numerical tolerances.

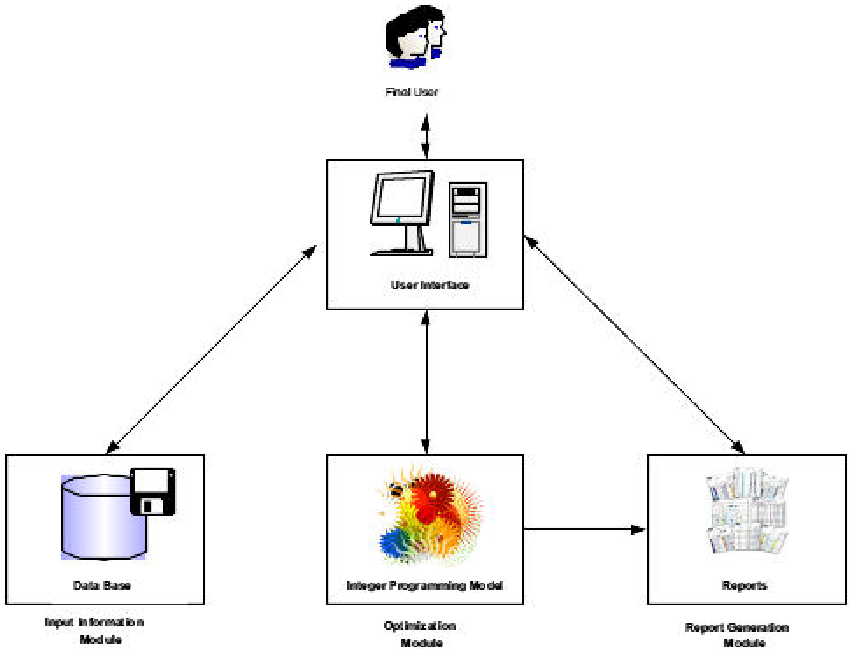


Figura 2.2: The flowchart shows the four main modules that comprise *eClasSkeduler*.

The *optimization module* contains the code for the IP model and a directory that stores the specific models of previously solved instances. These may be classified according to the combination of objectives defined in each case. A *CPLEX* 9.0 solver [91] is used to manipulate and execute the models and find the optimal solutions.

The *report generation module* receives the solution results as input; it processes and transforms them to produce management reports, including reports on course schedules for specific fields of study, the use of classrooms, and schedules for individual instructors. It also generates performance indicators for management control and resource allocation such as unused classroom capacity for each time slot, extra lab rentals, and the number of schedule conflicts between excluded courses.

The optimization model

The *optimization module* contains an integer linear programming model that defines the optimal timetables for all *EEU* annual courses. The model incorporates all the *EEU*'s objectives, as well as the requirements and constraints for ensuring its normal operation. It attempts to assign available *EEU* classrooms efficiently relative to cost and capacity, avoiding the use of external premises and schedule conflicts between excluded courses wherever possible.

The model assigns each course a unique timetable pattern defined by a vector with three components-a classroom, a time slot for two different days, and a period of consecutive weeks over which the course is taught. For a course to be assigned to a given timetable pattern, its classroom-capacity component must be equal to or greater than the number of registered students, the period of consecutive weeks must equal the course duration, and the first class must fall within the window defined for the course start date. If the pattern meets all these conditions, it is classified as a feasible pattern for that course.

We begin the presentation of our scheduling model by introducing the necessary notation. The set of courses is denoted C and the set of available classrooms (excluding laboratories) is defined as S. A *timetable pattern* is a set of timetables and classrooms in which each element consists of a triplet defined by a classroom, a period consisting of consecutive weeks and a time slot. For example, if a course c is assigned the pattern $1903/6-25/Mon-We$d, its classes are all held in Room 1903, begin in Week 6 and end in Week 25 (a duration of 20 weeks) and are given in the Monday-Wednesday time slot. In the EEU, there are over 5,000 unique patterns.

Note that courses can only be assigned to a pattern that is feasible, which means it must fulfill certain conditions. In specific terms, a pattern is *feasible* for a course $c \in C$ if the restrictions listed below are satisfied:

1. All classes are scheduled in the same classroom, which has sufficient capacity for the course's enrollment. A classroom's capacity is denoted $s \in S$ and enrollment is represented by the parameter D_c for course c.
2. The first week of classes falls within the course's time window.
3. The number of scheduled weeks matches the duration of the course.

If we define P as the set of feasible patterns, then $P(c)$ is the set of feasible patterns for course c and $P = \bigcup_{c \in C} P(c)$. Let us denote $t = 1, \ldots, T$ any day in the academic calendar on which classes may be held. We also define a_{pst} as a parameter that takes the value 1 if pattern p includes a class held on day t in room s, and 0 otherwise. Furthermore, for each day t and for each $n \in \mathbb{N}$, the set $TL(t, n) \subseteq P$ is the set of patterns p such that day t is the $n-$ th class of the pattern.

For each course c the term $Lab(c)$ is the set of lab classes for course c, which are counted correlatively with the lecture classes. Thus, if course c includes two lab classes, one in the third week and one in the ninth, the set of them is defined as $Lab(c) = (3, 9)$.

Finally, for each c we also define a corresponding set $CI(c)$ of *excluded* courses that, to the extent possible, are not scheduled on the same day. Note that if $a \in CI(c)$ and $b \in CI(c)$ are *excluded* courses with respect to course c they are also *excluded* relative to each other.

The model contains three sets of decision variables. The first set consists of binary variables that are activated if a given course is assigned to a certain timetable pattern. More specifically, for each course c and pattern $p \in P(c)$, the variable:

$$x_{cp} = \begin{cases} 1 & \text{if course } c \text{ is given as determined by pattern } p, \\ 0 & \text{otherwise.} \end{cases}$$

The second set are non-negative integer variables that represent the use of external lecture rooms and laboratories. Counter variable z_t and y_t tracks the number of extra lecture rooms and labs needed on day t respectively. In the present context, laboratories all have the same capacity, equipment and other resources, as opposed to the lecture rooms whose capacities vary and whose use costs will therefore depend on their assignments. Counter variable q_{ct} tracks, for each course c, the number of schedule conflicts on day t between c and its corresponding *excluded* courses. And finally, the third set is composed of non-negative integer variables that count the number of schedule conflicts among *excluded* courses. Note that the last two decision variable sets were included given that on certain occasions not all of the operating requirements and conditions can be completely satisfied. In such cases the model will incorporate slack variables.

As can be seen, the objective function, represented by Equation (2.1), combines four different objectives. The first term of the function represents total classroom rental cost of the course-classroom assignment. This factor depends on a classroom's characteristics such as years of use and installed equipment; the latter typically includes data show projectors, personal computers, electronic blackboards, and audiovisual equipment. The parameter F_{cp} in this term is defined as the cost of scheduling course c according to pattern $p \in P(c)$. The second objective is the opportunity cost of assigning a classroom to a course with fewer registered students than the room's capacity. The difference between course enrollments and nominal classroom capacity is defined as unused capacity. The *EEU* administration valued the cost of an empty seat based on the loss that the *EEU* suffered when a student could not register in a closed course because of a lack of classroom capacity. The term $Cap_s - D_c$ represents the difference between the capacity of a given classroom assignment and the number of enrollments in a particular course. Note that because closed courses are not scheduled, we could not know the number of their enrollments with certainty. The third term in the objective function is the total rental cost of external classrooms. This objective minimizes the cost per day of using extra lecture rooms and laboratories. The parameters g_t and d_t are defined as the unit cost of using a lecture room and lab room per day t respectively. The fourth term is the total cost associated with the schedule conflicts between excluded courses. The *EEU* administrators also quantified this because the *EEU* must hire an extra instructor when a conflict occurs. Note that they determined this cost and the opportunity cost of an empty seat based on historical data.

$$\text{mín} \quad z = C_1 \sum_{c\in C} \sum_{p\in P(c)} F_{cp}x_{cp} + C_2 \sum_{c\in C} \sum_{p\in P(c)} (Cap_s - D_c)x_{cp} + C_3 \sum_{t\in T} [\mathrm{d}_t y_t + g_t z_t] + C_4 \sum_{c\in C} \sum_{t\in T} q_{ct} \tag{2.1}$$

Note that the objective function terms all have individual weight coefficients C_{i} (i=1,...,4) that reflect the relative importance of the different objectives as defined by the system user. The function includes criteria for resource assignment efficiency minimizing operating costs and unused classroom capacity as well as operability criteria that minimize course conflicts.

As regards the constraints, the Equation (2.2) ensures that each course $c \in C$ is scheduled using just one of the possible feasible patterns defined in the set $P(c)$.

$$\sum_{p \in P(c)} x_{cp} = 1 \quad \forall\, c \in C. \tag{2.2}$$

The second set of constraints represented by Equation (2.3) avoids schedule conflicts in the use of lecture rooms, ensuring that no more than one class is held in each classroom $s \in S$ on a given day $t \in T$. If more lecture rooms are needed, the slack variable z_t is activated and the objective function value is penalized.

$$\sum_{c \in C} \sum_{p \in P(c)} a_{pst} x_{cp} \leq 1 + z_t \quad \forall\, s \in S,\ \forall\, t \in T. \tag{2.3}$$

The third constraint set represented by Equation (2.4) is analogous to the preceding one but applies to laboratories. It guarantees that the number of courses using a lab on a given day $t \in T$ does not exceed the number available, defined by the parameter $NLabs$. If more labs are needed, the slack variable y_t is activated and the objective function value is penalized.

$$\sum_{c \in C} \sum_{n \in Lab(c)} \sum_{p \in TL(t,n) \cap P(c)} x_{cp} \leq \mathrm{NLabs} + y_t \quad \forall\, t \in T \tag{2.4}$$

The fourth and final constraint set defined by the Equation (2.5) requires that wherever possible, for any day $t \in T$ and course c not corresponding to *excluded* course as defined by the set $CI(c)$ be given on the same day. If a schedule conflict in such a case cannot be avoided, the variable q_{ct} is activated, again imposing a penalty on the objective function value.

$$\sum_{j \in CI(c)} \sum_{p \in P(j)} \sum_{s \in S} a_{pst} x_{jp} \leq 1 + q_{ct} \quad \forall\, c \in C,\ \forall\, t \in T. \tag{2.5}$$

With the necessary notation now defined, we set out the proposed integer programming model can be set out in its entirety:

$$\text{mín} \quad z = C_1 \sum_{c \in C} \sum_{p \in P(c)} F_{cp} x_{cp} + C_2 \sum_{c \in C} \sum_{p \in P(c)} (Cap_s - D_c) x_{cp} + C_3 \sum_{t \in T} [\mathrm{d}_t y_t + g_t z_t] + C_4 \sum_{c \in C} \sum_{t \in T} q_{ct}$$

s.t.

$$\begin{aligned}
&\sum_{p \in P(c)} x_{cp} = 1 && \forall\, c \in C\,, \\
&\sum_{c \in C} \sum_{p \in P(c)} a_{pst} x_{cp} \leq 1 + z_t && \forall\, s \in S\,,\ \forall\, t \in T\,, \\
&\sum_{c \in C} \sum_{n \in Lab(c)} \sum_{p \in TL(t,n) \cap P(c)} x_{cp} \leq \mathrm{NLabs} + y_t && \forall\, t \in T\,, \\
&\sum_{j \in CI(c)} \sum_{p \in P(j)} \sum_{s \in S} a_{pst} x_{jp} \leq 1 + q_{ct} && \forall\, c \in C\,,\ \forall\, t \in T\,, \\
&x_{cp} \in \{0,1\} && \forall\, c \in C\,,\ \forall\, p \in P(c)\,, \\
&z_t, y_t, q_{ct} \in \mathbb{Z}+ && \forall\, t \in T\,,\ \forall\, c \in C\,.
\end{aligned}$$

2.5. Experiments and results

In this section, we analyze the results of an experiment we conducted to study the strengths of the system; we compared the timetables it generated for the academic years 2007 through 2009 with those developed using the *EEU*'s manual process. In each case, the manual schedules had been drawn up the previous October; our model used the same data available to the *EEU*'s schedulers. The planning horizon for each year was 40 weeks, with six weekly time slots (six days and one block per day). In the analysis, we looked at a series of performance indicators (Table 2.2) that reflect various management and resource- allocation criteria; we included all the objectives in the IP objective function and weighted them equally.

Indicator	Description
ETTR	Total time required to generate the course timetables and classroom assignments.
Total rental cost	Rental cost associated with the assigned classrooms.
Unused classroom capacity	Total number of empty seats in all classrooms assigned by the system. This value is calculated as the difference between the capacity of a classroom and the number of enrollments in the course assigned to it.
Number of courses with schedule conflict	Number of excluded courses that conflict with each other.
Number of external lecture rooms	Number of external lecture rooms assigned by the system. These classrooms are located off EEU premises.
Number of external labs	Number of lab rooms assigned by the system that are located off EEU premises.

Tabla 2.2: The performance indicators.

The time needed for timetable construction and report generation (ETTR) was cut dramatically from an average of two weeks to less than 30 minutes (Table 2.3), thus enabling the *EEU* to respond quickly to requests from private companies for closed courses. High-speed scheduling also means that multiple scenarios using different objective weights can be analyzed and compared.

Instance	N° Courses	N° Classrooms	N° *excluded* course sets	N° Lab classes	CPU Time
2007	30	20	12	112	19.6 min.
2008	38	25	20	170	22.6 min.
2009	55	25	30	182	28.7 min.

Tabla 2.3: Problem specification and computational data.

In comparison to the manual system, our system provided a substantial decline in classroom rental costs because two of the four objective function terms indicated lower total operating costs. *eClasSkeduler* boosts efficiency in using economic resources, thus also increasing profits. Table 2.4 shows the decline in the total rental costs.

	2007	2008	2009
EEU's manual process	60,300	69,420	90,230
eClasSkeduler	44,500	41,290	55,440
Difference	26 %	41 %	39 %

Tabla 2.4: Total cost ($US) of schedules by timetabling system.

Our system also resulted in a significant reduction in unused classroom capacity. Its method of assigning courses to classrooms left fewer empty seats per academic year; thus, it reduced rental costs because they are directly proportional to nominal capacity (Table 2.5). Under the manual system, the number of available classrooms grew each year, resulting in an increasing underuse of total nominal capacity.

	2007	2008	2009
EEU's manual process	314	413	575
eClasSkeduler	251	132	138
Difference	21 %	61 %	74 %

Tabla 2.5: Unused classroom capacity by timetabling system.

Using the timetabling system, we also observed a decline in the number of excluded course-schedule conflicts (Table 2.6). Students had more flexibility to register for advanced courses in a given subject field, thus increasing their satisfaction levels.

	2007	2008	2009
EEU's manual process	9	12	34
eClasSkeduler	1	5	15
Difference	8	7	19

Tabla 2.6: Number of *excluded* courses with schedule conflicts by timetabling system.

eClasSkeduler also generated schedules that required fewer external classrooms and labs. The manual schedules assigned 18 external classrooms and eight additional labs between 2007 and 2009; in contrast, our system generated schedules requiring only three classrooms and one lab (Tables 2.7 and 2.8). Our system virtually eliminated the logistical problems of equipping off-premise classroom space and providing transportation, thereby sparing the *EEU* all the associated costs and the damage to its corporate image.

	2007	2008	2009
EEU's manual process	9	5	4
eClasSkeduler	2	1	0
Difference	7	4	4

Tabla 2.7: Number of external lecture rooms by timetabling system.

	2007	2008	2009
EEU's manual process	1	2	5
eClasSkeduler	0	0	1
Difference	1	2	4

Tabla 2.8: Number of external laboratory rooms by timetabling system.

2.6. Discussion of results

The objective function of the model minimizes the total cost expressed as a weighted sum of its various terms-Equation (2.1). In our experiment, we gave each term a weight of 1 in line with the desire of *EEU* administrators to minimize the total cost of the timetable scheduling. Note, however, that if we ranked each term in accordance with its contribution to the final cost function, the most important objective would be the one with the largest contribution. The *EEU* administrators validated our results and concluded that the timetables generated made efficient use of resources at minimum cost. In the light of this performance, the *EEU* implemented the timetables that *eClasSkeduler* produced for 2009. We did not directly approach the search for the most appropriate weighting factors for each term in the objective function; however, we intend to incorporate a method of determining the relative weights for these terms in the decision-making process in a future study. Existing research reports several interesting solution approaches that address the multiobjective nature of certain applications and business problems [15, 130, 155], suggesting this is a fertile area for further research and development.

2.7. Conclusions

This paper has presented a timetabling system that uses an IP model to solve the problem of scheduling classes and assigning classrooms at the *EEU*, a branch of the Universidad de Chile. *eClasSkeduler* generated satisfactory results for all involved participants in the *EEU*. In particular, it curtailed operating costs and lowered unused classroom capacity- the two most important findings from the institution's viewpoint because they lead to more efficient use of economic and infrastructure resources. In addition, the system produced fewer course schedule conflicts and off-premise classroom assignments, thus allowing students to have more options in selecting courses in the same academic term and increasing the *EEU*'s attractiveness to students. In addition, the decreased need to use external resources also reduced the need for students and instructors to travel from the *EEU*'s premises.

Based on these results, we conclude that incorporating our method into the *EEU*'s decision-making process would be beneficial. First, it would enable the *EEU*'s management to choose the best of a series of options by evaluating multiple scenarios through simple manipulation of the system parameters. Second, the system would considerably shorten execution times from several weeks under manual scheduling to mere hours. This would allow the *EEU* to respond rapidly to changing requirements, significantly reducing the risks and uncertainties inherent in the manual approach. Third, the system automates what is otherwise an extremely slow, tedious, and complicated task of scheduling by hand.

Capítulo 3

udpSkeduler: A Web architecture based decision support system for course and classroom scheduling

Abstract

A key process for post-secondary educational institutions is the definition of course timetables and classroom assignments. Manual scheduling methods require enormous amounts of time and resources to deliver results of questionable quality, and multiple course and classroom conflicts usually occur. This article presents a scheduling system implemented in a Web environment. This system generates optimal schedules via an integer-programming model. Among its functionalities, this system enables direct interaction with instructors in order to gather data on their time availability for teaching courses. The results demonstrate that significant improvements over the typical fully manual process were obtained.

3.1. Introduction and context

For decades, higher educational institutions have relied on computer-based systems to support a wide range of administrative functions such as course registration, student record management and storage, and personnel and financial management. The use of such systems has significantly improved these institutions' organizational agilities, allowing them to achieve and maintain a considerable degree of administrative and operational efficiency.

A basic recurring process in colleges and university administration is the generation of timetables for course offerings. In its simplest form, this course scheduling process assigns each course to a time slot and a classroom subject to a series of requirements and constraints [150]. These conditions typically include the type and size of the required classroom, instructor time availability, and the avoidance of timetable conflicts for courses students are required to take in the same semester, in addition to any other conditions relating to specific objectives that have been defined by the institution administrators. The scheduling problem created by such a set of circumstances clearly poses an interesting intellectual challenge.

Given the combinatorial nature of the problem and its computational complexity, which has been categorized as *NP-hard*, manual solution approaches based on trial and error are naturally considered to be inefficient in scenarios with large numbers of courses and classrooms [112]. In real applications, the methods used must address not only the computational complexity but also a series of interconnected factors that vary over time, such as the instructors' time preferences, course prerequisites, and the creation of new courses and study programs.

The problem of generating course schedules and classroom assignments is by no means a new one, having been studied extensively in the literature and addressed through various solution approaches based on operations research [100, 139]. In practice, however, only a small number of these studies have been implemented as decision support systems (DSS). A series of interconnected factors underlie this dearth of real-world applications, including organizational resistance to change and the adoption of new technologies [57], organizational attitudes and lack of commitment [82], negative perceptions of the systems' user-friendliness [136], the degree of problem difficulty supported by the DSS [84, 156], the need for a multi-disciplinary work team to perform system-related tasks [14, 82, 137], as well as the required levels of users' experience [129, 135], training [156] and participation in the development of the system [82, 137].

The present article describes a scheduling decision support system that is based on a Web architecture called *udpSkeduler*. It was implemented in the Faculty of Engineering (hereafter referred to as "the Faculty") at Universidad Diego Portales (UDP) in Santiago, Chile, where it has generated a series of benefits that will be described in the following sections. The system uses an integer-programming model in order to define optimal schedules, which simultaneously incorporates all of the problem's requirements and constraints, as well as any additional objectives that are set by the Faculty Board of Directors. Further benefits of the *udpSkeduler* implementation include a reduction in the time that is required for scheduling due to process automation, improvement in schedule quality with fewer human errors and timetable conflicts, the ability to easily and quickly explore and analyze multiple scenarios through a user-friendly interface, and a reduction in work load for planning staff, which frees them for other tasks.

The structure of this part is as follows. Section 3.2 reviews the literature regarding the role of computer-based support systems in course schedule decision-making at educational institutions. Section 3.3 introduces the Faculty's timetabling problem and the various stages of the course scheduling process. Section 3.4 describes the *udpSkeduler* system and its constituent modules. Section 3.5 presents the results of the implementation, focusing on quantitative evidence of the improvements that are provided by the system. Finally, Sections 3.6 and 3.7 discuss the computational experiments that were conducted with the system and present our conclusions.

3.2. The importance of computer-based systems based on mathematical models in higher education course scheduling

Decision support systems based on mathematical programming models have been supporting course-scheduling processes for more than two decades [54]. The most commonly used solutions employ linear and integer programming techniques. The study by [62], a seminal work in this area, offers a theoretical analysis of a series of optimization problems that have inspired a variety of real-world problem applications. Solution approaches that are built around integer programming are presented in several studies such as [60, 61, 66, 115, 125]. Most of these works propose models in which the individual weekly sessions for each course are separately assigned, and the time slot configuration conditions are imposed by applying constraints. In the model to be presented here, the course sessions are assigned simultaneously to a single pattern made up of two components: a set of time-slots and a classroom.

Indeed, most of the above-mentioned studies concentrate on theoretical development and do not analize an actual computer-based implementation; however, some researchers have integrated their solution approaches into practical applications by including decision support systems for the timetabling process. In 1989, [54] developed a computer-based system for a scheduling problem at an adult education institute that delivers initial solutions with relatively few course conflicts in a matter of minutes which can then be manually fine-tuned. They report that this iterative process produces a satisfactory schedule in no more than an hour.

In 1993, [93] implemented a computer-based scheduling system at the *Loughborough* University Business School that efficiently manages course information through a range of queries to a database. One of the principal benefits for the School has been the greater standardization of the process, which has been completely automated. Similar benefits are achieved in the system we propose here through partial structuring and automation of the process and the efficient management of information by means of database filters. These filters organize the information to fit the structure of the pattern assignment variable in the integer programming model, mapping the patterns within a reduced solution space and eliminating all a priori infeasible combinations.

A computer-based system that was implemented by Ferland and Leurent in 1994 [75], known as *SAPHIR*, adopts a heuristic approach to solve an integer-programming model, and provides an interactive optimization procedure that is similar to that in [54].

[150] developed a computer-based course scheduling system for Anderson School of Management at the University of California at Los Angeles (UCLA) that generates schedules in two stages. The first stage exclusively deals with the core courses whereas the second schedules all of them. This allows the system to generate multiple versions of the core course schedule reflecting different criteria. A sensibility analysis can then be conducted and the best version can be chosen, with the end result being a better overall schedule.

The method also integrates information management and report-generating functions that facilitate the process of fine-tuning the timetables.

Another decision support system for course scheduling, known as *SlotManager* [79], follows an approach that was originally designed by [102] by using a three-component structure that consists of interfaces, databases, and an optimization model. This system features a user-friendly interface and functionalities for obtaining information on course registrations, available classrooms, and course characteristics. The general structure of our *udpSkeduler* system is derived from this work but includes an additional module that supports direct interaction with the teaching staff, capturing instructors' time availability for giving classes and related preferences.

[66] report on a computer-based system that was developed at the Athens University of Economics and Business that utilizes both a mathematical programming model and heuristic procedures in order to arrive at definitive course and examination schedules. The system is structured around five modules: a data module that manages information, a control system module that includes the user interface and generates data to be fed to the models, an optimization module that solves the mathematical programming model and executes the heuristics, a report generating module that reports on class timetables for use in decision-making, and an evaluation module that examines the quality of the generated schedules. According to the authors, this system provides better timetables for the students, a better use of classrooms, and a satisfaction of instructors' time preferences.

In 2002, [88] describe a computer-based system they developed, known as *SchedulExpert*, that uses an integer-programming model to generate course schedules for Cornell University's School of Hotel Administration. The principal benefits of this system are that it defines schedules that meet the objectives of the institution's governing body and reduces the time that is needed to produce them. In particular, it eliminates timetable conflicts between required core courses and certain sets of elective courses while, at the same time, minimizes conflicts between certain electives. Finally, this model also supports instructor course assignment decisions.

[112] presents a computer-based system named *eClasSkeduler* for courses that are given by the Executive Education Unit at the Universidad de Chile. Because of the way courses are taught by the Unit, with different course startup dates and varying course durations, the problem the system must solve requires that each week be individually and simultaneously scheduled. It therefore departs considerably from the problem addressed in the present work in which a single weekly schedule is replicated for an entire semester. Furthermore, although *eClasSkeduler* uses a course pattern assignment variable, the pattern structure differs from the one to be used here in that the former is defined by three components: a set of time slots, a classroom and a period of consecutive weeks over which the course is taught. As for the reported benefits of *eClasSkeduler*, they include a reduction in off-site classroom rental costs, fewer course conflicts, and efficient classroom use.

Finally, we note that all of the surveyed papers of [139] and [100] clearly illustrate the significant benefits of incorporating computer-based systems into the decision-making process for course scheduling.

The *udpSkeduler* contains three important elements that distinguish it from the systems discussed above. The first element has to do with systematizing the capture of instructors' time availabilities for giving classes and associated preferences through a user-friendly interface. The interface allows instructors to interact directly with the system, thus streamlining the process of inputting this key information. The second element is the formulation of an integer programming model with decision variables based on the assignment of courses to patterns for the scheduling problem of an institution's undergraduate programs. Finally, the third distinguishing element is the efficient management of the database storing the system input information.

3.3. The research problem

The Faculty at *UDP* offers four undergraduate programs: industrial engineering, computer and telecommunications engineering, civil engineering, and civil engineering technology. The first three academic programs are normally completed in 12 semesters and the fourth program in 10. Each engineering program requires the students to take a relatively rigid set of courses that are distributed over the duration of the program according to curricula that define which courses must be offered in each semester and which courses must not be scheduled in the same time slots. Many of the courses are common to more than one curriculum, thus increasing the potential for time conflicts that must be avoided.

Approximately 300 courses must be scheduled each semester. The various sessions (classes) of a single course may be of three types: lectures, laboratory classes and tutorials (recitations), the first two considered as one for scheduling purposes. Lectures and lab classes are taught by instructors previously assigned to courses whereas tutorials are given by teaching assistants. The exact number of courses that are taught from one semester to the next depends on a range of factors, such as course demand, instructor time availability, and the creation of new courses. Each course consists of one, two, or three 90-minute lecture sessions that can be scheduled in four different weekly time slot configurations: 1) one session at any time; 2) two sessions, given on different days at the same time; 3) two consecutive sessions given on a single day; and 4) three sessions, given on three different days at the same time. The sessions for courses following the fourth configuration may be held either on Mondays, Wednesdays, and Thursdays or Tuesdays, Wednesdays, and Fridays.

An interesting element that contributes to the complexity of the problem is that certain multisession courses must alternate between different classroom types. For example, a physics course might have one session in a typical classroom for theory lectures, whereas the second session is held in a laboratory for conducting experiments.

The Faculty's infrastructure consists of 45 classrooms; however, the number that is actually available for courses changes from semester to semester because of the various nonteaching activities such as, seminars, conferences, and workshops, which compete for the use of space. There are five types of classrooms: lecture rooms, physics labs, computer labs, civil engineering labs, process simulation labs, and auditoriums. These spaces may be further differentiated by size and installed equipment.

There are 270 instructors on the Faculty's staff, who are categorized by their full-time or part-time status. Part-time instructors are well-known practitioners from industry who give professionally oriented courses whereas full-time instructors are devoted to teaching and research. Both the number of instructors and their time availabilities vary from semester to semester.

Before the implementation of *udpSkeduler*, the course scheduling process was handled by a team of three planners who generated the timetables and a coordinator who exclusively worked on assigning classrooms. The previous scheduling process was manual and used the schedule from the previous semester as a base format. Errors were frequent, causing numerous scheduling conflicts that were primarily caused by a tendency to assign too many courses to the same time slots, thus, overwhelming the supply of available classrooms. As a result, some courses had to be assigned to external (off-site) classroom space, which presented an awkward situation that led to a series of administrative headaches.

3.3.1 The scheduling process and course registration

The scheduling process for a given semester consists of a series of stages that involve the interaction of all of the Faculty's departments and other academic units. The **first stage** is devoted to gathering all of the information that is needed to decide the schedules. It begins with the identification of the available classrooms and the courses that will be offered that semester. Then, for each course, the corresponding academic unit specifies the technology and classroom requirements and the instructors' time availabilities. An estimate of course demand is also made using a procedure that is based on historical registration data and failure rates.

Once all of this information has been collected, the process enters the **second stage**. Here, a set of preliminary schedules is compiled. Each academic unit's schedule planner independently defines the schedules for that unit's courses, incorporating the information on each instructor's time availability. The three planners (one for each program, with civil engineering and engineering technology are considered to be a single unit) and the coordinator frequently meet in order to harmonize their respective schedules, eliminating any classroom "double-bookings" or timetable conflicts for courses that are common to more than one program grid. Most of these scheduling inconsistencies are due to the non-centralized nature of the decision-making process.

The objective of the **third stage** is to capture the real demand for each course. The preliminary course schedule is published and students register in the various offerings during a four-day registration period. This is followed by the **fourth stage**, in which a new and definitive classroom assignment is conducted. All four stages in the timetabling process are shown in Figure 3.1. Stages II and IV are automated by *udpSkeduler*, as described in the next section.

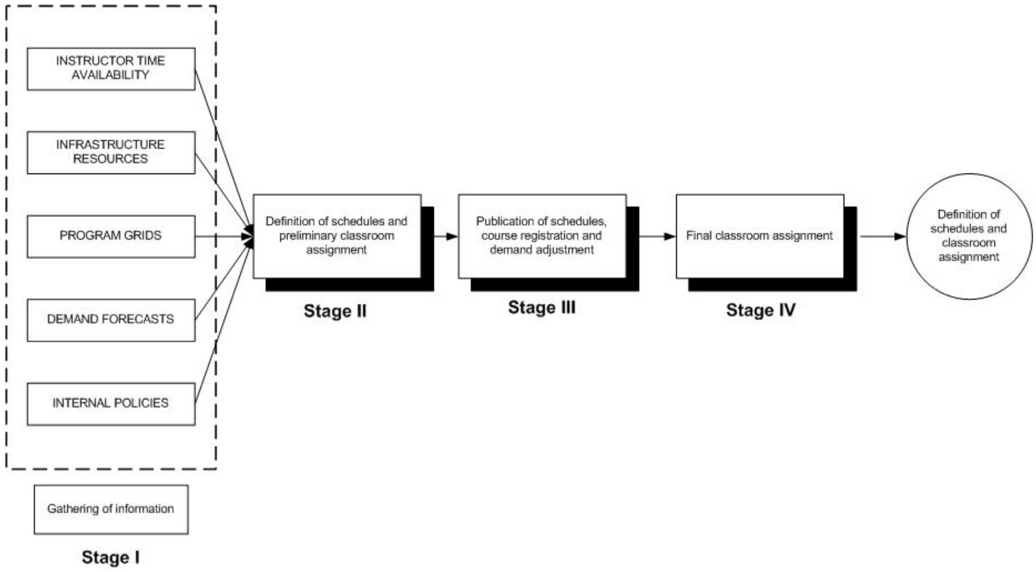


Figura 3.1: The four stages of the course scheduling process.

3.4. Description of the system and optimization model

udpSkeduler is a decision support system that is based on mathematical programming models [102]. Built in a Web environment, it generates optimal course schedules by taking into account all of the constraints and internal Faculty policy objectives.

The *udpSkeduler* architecture consists of five main modules, which are depicted in Figure 3.2. The first is the *user interface module*, which is a mechanism for checking and adjusting the system's parameters and options and for managing information. Through this interface, the user can define the program grids and specify each course's characteristics in terms of its technology and infrastructure requirements. Various reports relating to student record management can also be generated.

The second module is the *data input module*, which stores all of the information that is necessary for generating the course schedules in a relational database. Some examples of these data are instructor time availability, classroom type, program grids, and course characteristics. The schedules for previous semesters are also stored here. Various filters are provided that quickly discard infeasible assignments, thus reducing the computation time that is required to find a solution. One of the filters prevents a course from being assigned to a classroom with insufficient seating capacity, whereas another filter prevents a course from being assigned to a time slot that is outside of the instructor's time availability. Figure 3.3 depicts the structure of the database with its main tables.

The third module is the *time availability module*. As one of the most important elements of the *udpSkeduler* system, it supports interaction between the instructors and the Faculty's planners. Through a user-friendly interface that can be accessed via the Internet, instructors can enter their personal accounts and specify their time availability.

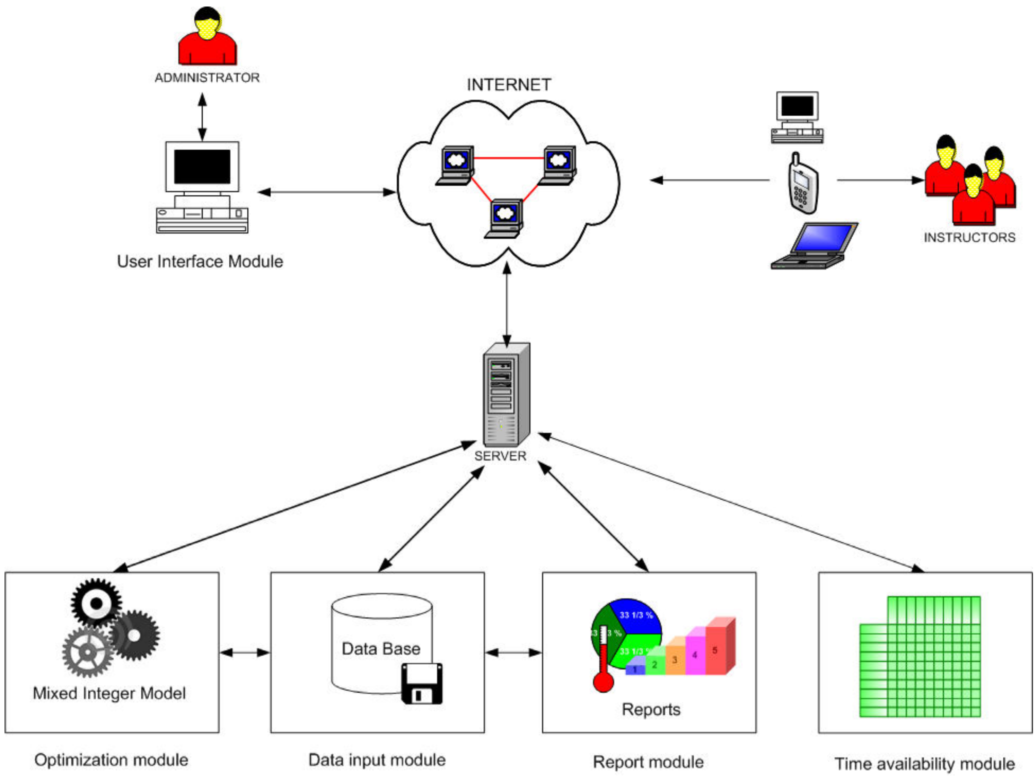


Figura 3.2: General architecture of the *udpSkeduler* system.

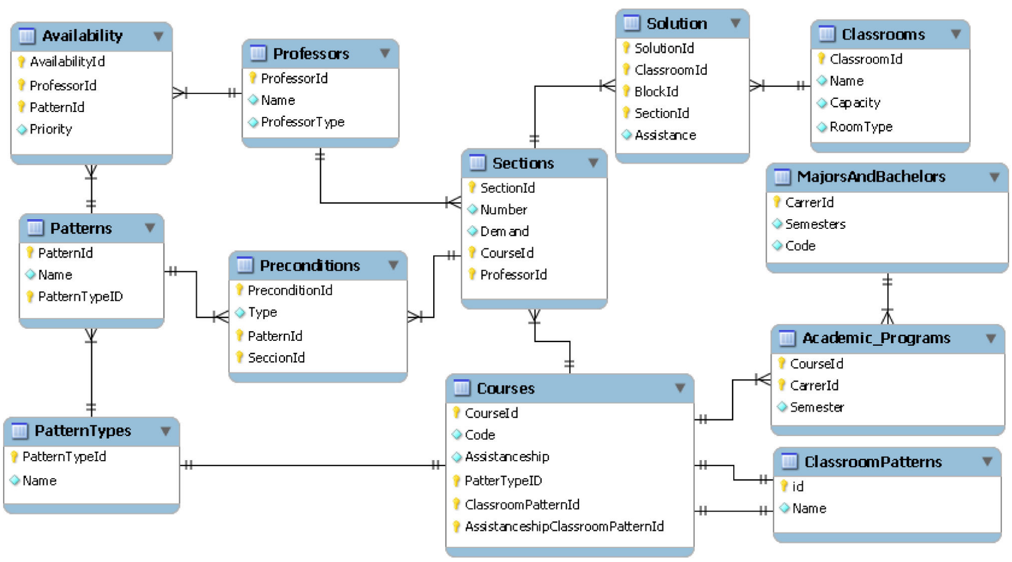


Figura 3.3: Architecture of the *udpSkeduler* database.

As can be seen in Figure 3.4, the interface for an account displays a table that represents the various time slots. The slots for which the instructor has already indicated availability are displayed in green, and the currently selected ones are shown in red. Timetable preferences can be expressed by ordering the slots by priority from the most preferred to the least preferred (see the dashed-line box in Figure 3.4). If they wish, instructors can also indicate that they have no preferences.

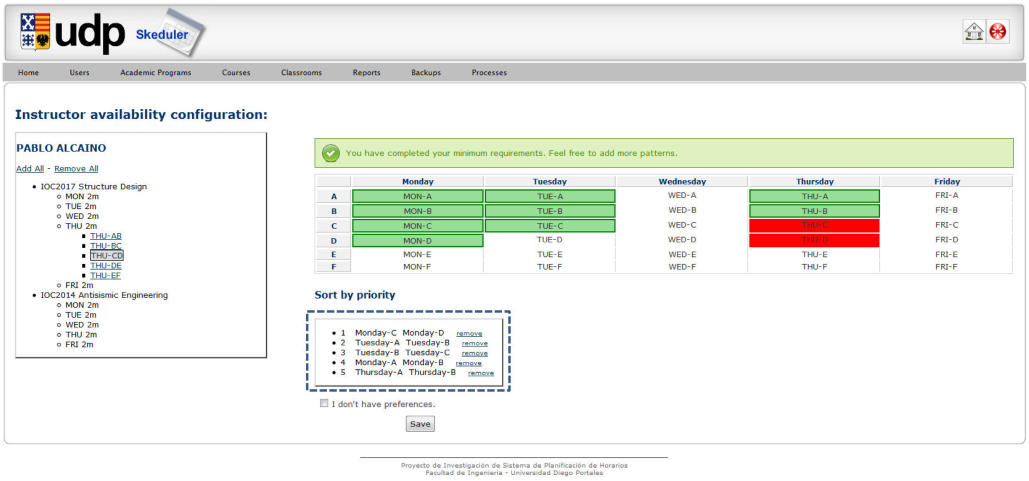


Figura 3.4: Screen shot of the instructor account interface for the *udpSkeduler* time availability module.

The fourth module is the *optimization module*, which contains the code for the integer-programming model. The model is solved using the *CPLEX* software package [90]. Finally, the fifth module is the report model, which produces a series of management reports after each execution of the optimization module. These reports present a series of indicators for evaluating the schedules that have been generated by the module, such as the average classroom use, the number of classrooms available for each course session, and course conflicts, as well as model indicators, such as the objective function value, the number of variables and constraints, and the solution computation time.

The optimization model

The integer-programming model assigns each course a unique timetable pattern defined by a vector that contains two components, one representing a set of time slots (one for each session) and the other representing a classroom. For example, if course c is assigned the pattern (*monday.A - thursday.A, FDI315*), it will be given on Mondays and Thursdays in slot A, and both sessions will be held in room $FDI315$.

The advantage of a pattern-based formulation is that its linear relaxation is tighter [125], thus facilitating the solution of the model by commercial solver packages such as *CPLEX* [90]. The greater tightness is due to the fact that for each linear relaxation solution, an equivalent solution can be constructed that assigns individual course sessions. This relationship does not necessarily work both ways, however, as there exist feasible solutions of the model relaxation that assign sessions but which have no equivalent solutions for the pattern-based model. As an example, if we consider the solution of the linear relaxation of the model in which the variable that assigns course c to pattern (*monday.A - thursday.A, FDI315*) takes a value of 0.7, the equivalent solution for the relaxation of the individual course session model is constructed by assigning the same value to the variables that assign each session. In other words, a value of 0.7 is assigned to both the variable that assigns course c to slot (*monday.A)* and classroom *FDI315* and the variable that assigns course c to slot (*thursday.A)* and classroom *FDI315*. If, on the other hand, we consider a fractional solution of the individual session model in which the variable that assigns course c to slot (*monday.A)* and classroom *FDI315* takes a value of 0.7 while the variable assigning course c to slot (*thursday.A)* and classroom *FDI315* takes some other value such as 0.2, no equivalent solution for the linear relaxation of the pattern-based model can be found.

For each course c, a feasible pattern set exists within the total set of patterns. A feasible pattern for a given course is one for which the previously assigned instructor has time availability and the number of registrants is less than or equal to the classroom capacity. The definition of the feasible pattern subset is determined using the database filters, and the total number of feasible patterns may exceed 100,000 in some cases. For larger instances, however, explicit generation of the set of feasible patterns will produce too many sets and the problem will not be computationally solvable. To address such situations, a dynamic pattern generation approach could be employed by implementing a branch-and-price algorithm [16].

We began our exposition of the model by defining and explaining the necessary notation. The set of courses is C, the set of classrooms is S, the set of instructors is L, the total set of time slots is T, a feasible set of timetable patterns is P and the set of such patterns for a given course $c \in C$ is $P(c)$.

A timetable pattern (hereafter simply "pattern") is a vector containing two components, one representing a set of time slots and the other a classroom where the sessions (classes) of a course are to be given. A feasible pattern p for a course $c \in C$ is one that satisfies the following conditions:

1. The classroom capacity is high enough to accommodate the number of students in the course.
2. The classroom is of the right type for the course (all classroom spaces are classified into lecture rooms, various types of laboratories and auditoriums).
3. The set of time slots follows the weekly time slot configuration defined for the course.

There are two types of course sessions, lectures and tutorials (the former including lab classes), which may be scheduled according to different patterns. Of the three sets of decisions variables in the model, the first two are binary and are activated if the lectures or tutorials of course $c \in C$ are assigned to pattern $p \in P$. More specifically, for each $c \in C$ and pattern $p \in P$ we define the following variables:

$$x_{cp} = \begin{cases} 1 & \text{if course } c \text{ lectures are scheduled according to pattern } p, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$y_{cp} = \begin{cases} 1 & \text{if course } c \text{ tutorials are scheduled according to pattern } p, \\ 0 & \text{otherwise.} \end{cases}$$

This use of binary decision variables to assign a course to a unique pattern differentiates the model from those that use binary variables for assigning each course session to a time slot and constraints requiring the individual sessions to be given in predefined sets of time slots [60, 61, 75, 115, 150].

The third set of decision variables contains non-negative integer counter variables that register the number of timetable conflicts among groups of courses students are required to take in the same semester. The set of these course groups is denoted R. For each group $r \in R$ we define a counter variable z_{rt} that registers the number of timetable conflicts among the courses in that group for time slot $t \in T$.

The model's objective function contains five terms that represent the various criteria of concern to the Faculty- see Equation (3.1). The first term attempts to minimize the use of the Faculty's auditoriums. For this purpose we define the set $P(\text{aud})$ to contain all of the patterns that use an auditorium as a classroom. The second term minimizes the assignment of tutorial sessions outside the preferred times, reflecting the internal Faculty policy of scheduling all tutorials to the best possible extent on a single day of the week. The set of non-preferred slots is denoted NM. The third term minimizes the number of timetable conflicts in order to ensure more flexibility for students who are repeating courses or taking courses ahead of the normal program grid sequence. The fourth term minimizes the number of external classrooms. The set of all patterns assigning an external classroom is denoted $P(\text{ext})$. Finally, the fifth term attempts to maximize the sum of the time preferences that have been entered by instructors in the time availability module. The parameter g_{lp} defines the time preference of instructor l for pattern p and the set of courses taught by instructor l is denoted $C(l)$. Each indicated preference is weighted by a parameter that assigns a value of 10 to the most preferred time slots and 1 to the least preferred. A value of 1 is assigned when an instructor indicates no preference. Each objective function term has a coefficient that reflects its importance relative to the other terms. The values for these coefficients were defined by the Faculty Board.

$$\text{mín } z = C_1 \sum_{p\in P(\text{aud})} \sum_{c\in C} x_{cp} + C_2 \sum_{p\in NM} \sum_{c\in C} y_{cp} + C_3 \sum_{t\in T} \sum_{r\in R} z_{rt} + C_4 \sum_{p\in P(\text{ext})} \sum_{c\in C} (x_{cp} + y_{cp}) - C_5 \sum_{l\in L} \sum_{c\in C(l)} \sum_{p\in P(c)} g_{lp} x_{cp}. \tag{3.1}$$

Constraints (3.2) and (3.3) respectively impose that for each course $c \in C$, the lectures and tutorials are scheduled using a single pattern. This ensures that every course is or will be assigned a set of time slots and a classroom.

$$\sum_{p\in P(c)} x_{cp} = 1 \quad \forall c \in C. \tag{3.2}$$

$$\sum_{p\in P(c)} y_{cp} = 1 \quad \forall c \in C. \tag{3.3}$$

The second set of constraints is expressed by (3.4). It attempts to avoid instructor time conflicts by prohibiting the assignment of more than one lecture to an instructor in a single time slot. Note that this constraint set does not apply to teaching assistants (responsible for tutorials and lab classes), who are not chosen until after the final timetable schedules are generated.

$$\sum_{c\in C(l)} \sum_{p\in P(c)} x_{cp} \leq 1 \quad \forall l \in L. \tag{3.4}$$

The third set of constraints is defined by (3.5). Its role is to prevent the assignment of a classroom more than once to the same time slot. The set of feasible patterns for course $c \in C$ that assigns classroom $s \in S$ to time slot $t \in T$ is denoted $Q(c, s, t)$.

$$\sum_{c \in C} \sum_{p \in Q(c,s,t)} x_{cp} + \sum_{c \in C} \sum_{p \in Q(c,s,t)} y_{cp} \leq 1 \quad \forall s \in S\,, t \in T. \tag{3.5}$$

The fourth set of constraints is given by (3.6). It imposes that the lectures and tutorials for course $c \in C$ may not be assigned to the same time slot $t \in T$. The set of feasible patterns for course $c \in C$ that contain time slot $t \in T$ is denoted $U(c, t)$.

$$\sum_{p \in U(c,t)} x_{cp} + \sum_{p \in U(c,t)} y_{cp} \leq 1 \quad \forall c \in C. \tag{3.6}$$

The fifth and last set of constraints is specified by (3.7). It imposes that to the extent possible, the courses students are required to take in the same semester are not scheduled in the same time slot $t \in T$. In practice, timetable conflicts cannot be completely eliminated, so they are penalized by the third term of the objective function. If such a conflict cannot be avoided, the z_{rt} term in the objective function is activated and penalized.

$$\sum_{c \in r} \sum_{p \in U(c,t)} x_{cp} + \sum_{c \in r} \sum_{p \in U(c,t)} y_{cp} \leq 1 + z_{rt} \quad \forall t \in T\,, r \in R. \tag{3.7}$$

The foregoing notation and other considerations are reflected in the final specification of the integer programming model, which is written as follows:

$$\begin{aligned} \text{mín } z = \quad & C_1 \sum_{p \in P(\text{aud})} \sum_{c \in C} x_{cp} + C_2 \sum_{p \in NM} \sum_{c \in C} y_{cp} + C_3 \sum_{t \in T} \sum_{r \in R} z_{rt} \\ & + C_4 \sum_{p \in P(\text{ext})} \sum_{c \in C} (x_{cp} + y_{cp}) - C_5 \sum_{l \in L} \sum_{c \in C(l)} \sum_{p \in P(c)} g_{lp} x_{cp} \end{aligned}$$

s.t.

$$\begin{aligned} & \sum_{p \in P(c)} x_{cp} = 1 \quad \forall c \in C\,, \\ & \sum_{p \in P(c)} y_{cp} = 1 \quad \forall c \in C\,, \\ & \sum_{c \in C(l)} \sum_{p \in P(c)} x_{cp} \leq 1 \quad \forall l \in L\,, \\ & \sum_{c \in C} \sum_{p \in Q(c,s,t)} x_{cp} + \sum_{c \in C} \sum_{p \in Q(c,s,t)} y_{cp} \leq 1 \quad \forall s \in S\,, t \in T\,, \\ & \sum_{p \in U(c,t)} x_{cp} + \sum_{p \in U(c,t)} y_{cp} \leq 1 \quad \forall c \in C\,, \\ & \sum_{c \in r} \sum_{p \in U(c,t)} x_{cp} + \sum_{c \in r} \sum_{p \in U(c,t)} y_{cp} \leq 1 + z_{rt} \quad \forall t \in T\,, r \in R\,. \end{aligned}$$

3.5. Results of the implementation

In this section, we review the results of the Faculty's use of *udpSkeduler*. The system was implemented in two phases over the two semesters of 2010. The last entirely manual scheduling process, which was conducted for the second semester of 2009, will be our point of comparison. Table 3.1 summarizes the elements that were involved in the analysis of the system's effectiveness. Table 3.2 describes the performance indicators that were used to compare the qualities of the schedules that were generated for the three semesters.

	Semester		
	Case 1	Case 2	Case 3
	2009-2	2010-1	2010-2
Classrooms	45	44	44
Courses	267	315	285
Course sessions	738	867	794
Instructors	248	259	255

Tabla 3.1: Characteristics of the three analyzed semesters.

Performance indicator	Description
Use of auditoriums	Number of courses that are assigned to a Faculty auditoriums
Use of external classrooms	Number of courses that are assigned to external classrooms
Non-preferred tutorials	Number of courses with tutorials that are outside of preferred times
Different classrooms	Number of courses with sessions that are assigned to different classrooms
Course conflicts	Number of conflicts between courses that are normally taken in the same semester according to the program grid

Tabla 3.2: Scheduling performance indicators.

In the first phase of *udpSkeduler* application (semester 2010-01), the course timetables were defined by the Faculty's planners using the traditional manual procedure, with *udpSkeduler* employed only to automate the assignment of classrooms in the final stage (Stage IV) of the process (see Figure 3.3). In the second phase (semester 2010-02), on the other hand, *udpSkeduler* was utilized to define both the course timetables and the classroom assignments, thus automating Stage II as well as Stage IV.

Table 3.3 presents the results for the three semesters of our comparative analysis as measured by the aforementioned performance indicators. The results demonstrate that the *udpSkeduler* system produces substantial improvements over the fully manual ap-

	Semester		
	Case 1	Case 2	Case 3
Performance indicator	2009-2	2010-1	2010-2
Use of auditoriums	8	5	0
Use of external classrooms	17	11	0
Non-preferred tutorials	11	8	3
Different classrooms	49	0	0
Course conflicts	21	13	6

Tabla 3.3: Number of undesirable situations in semester course schedules.

proach. Furthermore, they show that the amount of improvement grows with the degree of involvement of the system in the scheduling process.

More specifically, a comparison of Case 1 to Case 2 (the last fully manual schedule and the first phase of *udpSkeduler* implementation, respectively) indicates that *udpSkeduler* generated a more efficient classroom assignment. One of its primary successes was to reduce the number of courses for which weekly sessions were assigned to more than one classroom; the incidence of such situations decreased from 49 to 0 (32 % of the total) when using *udpSkeduler* instead of manual scheduling. In addition, the use of auditoriums decreased from 8 to 5 and the use of external classrooms decreased from 17 to 11; however, because *udpSkeduler's* involvement was confined to Stage IV, the timetables themselves were still defined manually without regard to classroom availability, and the complete elimination of auditorium and off-site space utilization was not possible. Thus, in certain time slots, classrooms double-booking did occur. As for the improvements in non-preferred tutorials and course conflicts, these were achieved by better course scheduling by the Faculty's planners.

If we now compare Case 1 (fully manual) to Case 3, the second phase of the *udpSkeduler* implementation, in which its use was extended to two stages (II and IV) of the scheduling process, further significant improvements in the performance indicators are evident. The use of auditoriums was completely eliminated, no course had sessions that were assigned to more than one classroom and no external classrooms were needed. This last result was the most important one, because Faculty planners had not managed to assign all courses to rooms on site since 2008. As for the other indicators, non-preferred tutorial times were reduced from 11 to 3, and course conflicts were cut from 21 to 6. These outcomes demonstrate that centralizing decision-making can achieve global optima, with highly efficient assignments of infrastructure resources.

3.6. Discussion of results

Each of the terms in the optimization model's objective function is weighted by order of importance. The weights themselves were directly decided by the Faculty Board members. The most important objective is the minimization of external classroom use, followed by the minimization of timetable conflicts and then the minimization of auditorium use. The minimization of non-preferred tutorial times and of instructor time preferences were, respectively, fourth and fifth in the importance ordering. The definitive objective function was, therefore, a weighted sum of the objectives that were established by the Faculty Board. The final schedules were validated and approved by the Board, thus completing the integration of the *udpSkeduler* system into the overall schedule generation process.

The process of finding the best configuration for the objective function weights was not explored in this study. Various studies have suggested methodologies determining the optimal objective function weights [130, 155], and this continues to be a fertile area for additional research and improvement. The present authors intend to pursue this line of inquiry in future work.

3.7. Conclusions

A Web-based, course-scheduling and classroom-assignment system using an integer-programming model was developed for the Faculty of Engineering at Universidad Diego Portales. Known as *udpSkeduler*, the system has generated satisfactory course timetables and classroom space assignments on the Faculty's site. The schedules and assignments have fulfilled policy goals such as the use of the same classroom for each weekly session of every course, the elimination of the need for external classroom rentals an end to courses given in Faculty auditoriums.

The results of the *udpSkeduler* implementation demonstrate that integration of the system into the course scheduling process has led to multiple benefits. First, the time required for certain stages of the process was significantly reduced from weeks to a couple of hours. Second, certain stages of the process were successfully automated. Third, course timetable conflicts and other human errors generated by the previous manual process were eliminated. Fourth, the system's mathematical programming model can analyze a large number of solutions and find the optimal solution. Fifth, multiple scenarios can be quickly and simultaneously evaluated in terms of several different objectives by means of simple variations in the system parameters, thus reducing uncertainty in the scheduling decisions and allowing contingency plans to be generated. Sixth, at the organizational level a new communication channel was created between Faculty instructors and planners that accelerates and simplifies the exchange of information, especially in regard to instructor time availability and timetable preferences. Finally, the rapidity of the scheduling process using *udpSkeduler* has allowed planners to devote more time to revising and fine-tuning definitive schedules and other tasks.

Parte II

Modelos para la planificación de la capacidad en universidades

Capítulo 4

A MIP approach for infrastructure capacity planning in higher education: The effect of instructor timetable availabilities

Abstract

Solution approaches based on operations research provide support for a range of academic and administrative problems at higher education institutions. This paper presents an approach to the capacity planning problem based on a mixed integer linear programming model that determines the number and type of classrooms and course instructors needed for an institution's program and course offerings. The model incorporates certain short term conditions that affect decisions in the long term. The proposed approach was applied at the School of Economics and Business of University of Chile to test the effect on the School's planning process of different instructor timetable availability scenarios. The experiments conducted found that restrictive availabilities increased both costs and the number of new classrooms that would have to be built.

4.1. Introduction and context

An essential task in the administration of higher education establishments is the planning of capacity to address future changes in program and course enrollments. Capacity planning refers here to the correct estimation of the resources required to efficiently run a post-secondary institution's teaching activities. This process is of strategic importance given that it determines the fate of substantial resources and guides institutional efforts and growth patterns over the medium and long terms. Poor capacity planning decisions will negatively affect every aspect of a college's operations. If planned capacity is below what is required by the courses offered the quality of service will be significantly impaired, with certain programs facing space shortages and a lack of qualified instructors. This in turn could force a resort to various less-than-ideal arrangements including off-site space rentals, inappropriate classrooms, awkward class schedules and core courses taught by non-specialists. If, on the other hand, the institution's planned capacity exceeds requirements, it will suffer the effects of over-investment, high operating and infrastructure maintenance costs and the possibility of unnecessary instructional staff increases. Either situation is likely to lead eventually to general discontent among students, instructors and administrators alike, not to mention damage to the institutional image.

Since capacity planning decisions orient an institution's goals and efforts in the long term, some factors affecting future outcomes will not be known with certainty at the time these decisions are made. One such factor is the demand forecasts, which attempt to establish the numbers of students registering in courses over the planning horizon. These numbers are used to estimate the numbers and sizes of courses to be offered and on the basis of these, the total number of "instructor-seat-hoursrequired in each term. With these estimates, planners can also determine the size and composition of the instructional staff needed to teach the courses in the various programs the institution offers. Obviously, the accurate forecasting of course registrations hinges upon a clear understanding of how students progress through the programs' course configurations as well as certain other historical information.

Although overestimating capacity is costly, the evidence suggests many institutions have low classroom utilization rates [23, 22], defined as the ratio of the number of seat-hours taken up by all course offerings to the total number of available seat-hours [22]. For example, studies have shown that utilization rates vary from 20 % to 54 % in the UK [109, 68] and are almost 60 % in the US [148, 151, 159]. Some of the causes given for these low rates are: (a) poor choice of criteria for determining the number and size of courses; (b) course timetabling performed independently of classroom assignment; (c) the need to avoid schedule conflicts for courses normally taken simultaneously by students in a given program; and, as will become apparent below, (d) the limited availability of instructors as regards teaching timetables.

Although the objectives of capacity planning differ from those for course timetabling, there is an important relationship between them given that to properly determine long-term resource needs, the short-term scheduling of courses and infrastructure must be efficient [142]. Course scheduling assigns to each instructor-course a set of time slots and an available classroom while satisfying a series of constraints and operating requirements. There is an extensive operations research literature proposing a range of solution approaches for course scheduling, many of which are surveyed in [62]. Generally speaking, they are based on integer programming models [41, 6] or heuristic methods [126, 117, 42] and all of them attempt to maximize the use of infrastructure resources and instructional staff, but none of them employ infrastructure or enrollment growth projections over time.

The present article presents a solution approach that addresses the long-term capacity planning problem of post-secondary institutions. The approach consists of three sequential stages. The first two stages generate estimates that create the input information on course demand forecasts to be used in the third stage, which solves a mixed integer linear programming model (MIP). This model includes operating conditions that affect short-term decisions, such as avoiding course schedule conflicts, and long-term decisions, such as building new infrastructure. The proposed approach was applied by the School of Economics and Business at the University of Chile as a support tool for planning its capacity over an 8-year horizon. Each term, the School must schedule an average of 300 courses and 4 programs of study taught by a staff of 290 instructors in 35 classrooms of different types. A central element of the approach that will be emphasized here is the importance to capacity planning of the instructors' time slot availabilities. Instructors typically have time preferences which we intuitively assume are the periods in which most courses will be scheduled and for which, therefore, the greatest number of classrooms will be needed. Our experiments have shown that in practice, instructors are only available at certain hours and the number of time slots in which classrooms are effectively usable is relatively limited, prompting institutions to overbuild their infrastructure and thereby incur unnecessarily high investment costs.

The remainder of this article is organized into seven sections. Section 4.2 reviews the relevant literature. Section 4.3 introduces the general capacity planning problem for a post-secondary institution, defining the information requirements, objectives and various operating conditions to be satisfied. Section 4.4 discusses the generation of the course demand forecasts (the first two stages of the proposed approach), followed in Section 4.5 by a detailed description of the MIP model (the third stage) that generates the planning problem solutions. Section 4.6 describes the real case the proposed approach was applied to at the School of Economics and Business of the University of Chile. Section 4.7 sets out the results obtained, and finally, Section 4.8 presents our main conclusions and some ideas for future research.

4.2. Review of the literature

Long-term capacity planning involves decisions relating to the remodeling of existing infrastructure, the rental and construction of new buildings and the appointment of new instructors in response to changes in enrollment over time. One of the earliest works to survey higher-education capacity planning models was [142] and some years later by [26]. The formulations presented there may be divided into three classes: (a) enrollment growth projection models; (b) instructional staffing models; and (c) optimization and resource allocation models.

Class (a) models seek to plan the growth in students enrolling in a program of study and as they move through its constituent courses. One of the most common approaches for predicting the potential numbers of program enrollments over time employs time-series models [145]. An example is [55], which applies fuzzy logic to build a model using as input the historical number of students registering in each program and other historical series related to entrance examination scores. Time-series formulations also attempt to estimate students' movement through the programs' courses in order to determine the total number of instructor-seat-hours that will be required over the planning horizon. This will involve establishing which courses are shared by multiple programs and analyzing historical course failure rates as well as policies on special student admissions. Various techniques have been brought to bear in forecasting student flows through programs, including simulation models [134, 30], network flow models [127] and Markov chains [18, 106].

Class (b) models solve problems related to the growth and composition of instructional staffs and estimating instructor payroll costs. These formulations have been used mainly to allocate operating resources to academic departments. For example, [154] proposes a model that attempts to project a faculty payroll based on equity criteria and budget constraints. Instructors are characterized according to rank, subject area, years at the institution and average salary differences from other instructors teaching in the same subject areas. In a different context, [71] present a network optimization model that solves the teaching schedule problem by assigning instructors to courses. The model does not assign either classrooms or time slots but the approach described does estimate which subjects areas have a shortage or surplus of instructors and how to distribute resources between teaching units on the basis of the teaching load generated by the courses to be offered.

As for class (c) models, their objective is to model the way an institution is managed [143]. Thus, they attempt to optimally allocate available resources to its academic activities. [11] present a mathematical programming model that determines the optimal first-year enrollment levels at Yale University. The model can also generate projections of future enrollments over time. [17] propose a solution approach built around a linear programming model that identifies the optimal floor space to be assigned to each academic unit. The allocation is based on the teaching, research and extension activities each unit is responsible for. The optimization model has various objectives whose relative importance is determined using the analytic hierarchy process (AHP). Instead of directly scheduling events or courses in classrooms, it solves the total space assignment problem in general terms without considering future changes. The authors state that this approach works better when space is assigned to just one type of user (e.g., offices) as opposed to the assignment of space to classrooms, which are used on an alternating basis by different categories of students for teaching- or learning-related activities. In another interesting paper on the problem of planning infrastructure, [23] offer a solution approach in which a linear programming model assigns and remodels the existing infrastructure to increase classroom utilization but does not allow for new construction. The model schedules multiple types of events such as lectures, tutorials and seminars assuming they have a set timetable. Finally, no works were found in our literature survey that simultaneously plan capacity as regards infrastructure and instructors or that directly incorporate projections of enrollments or events to be scheduled over time.

4.3. Description of the problem

Consider the problem facing a higher-education institution that offers n programs of study and must plan its capacity over a horizon of S terms. Assume that the institution meets the infrastructure and instructional staff requirements of the programs using only its currently existing resources. Furthermore, assume that there is a set of classrooms grouped into K different types defined by layout and size. Each type is associated with certain teaching activities of the different courses and course enrollment. Instructors are responsible for teaching the courses making up the programs and are categorized into P types according to their full-time/part-time status, subject area and time slot availability.

Thus, the capacity planning process involves decisions in both the infrastructure and instructional staff areas:

- **Infrastructure decisions.** The decisions in this area are the following:

  - *Remodeling of existing infrastructure.* Decisions include whether remodeling of existing infrastructure is needed and which classrooms will be remodeled to adapt the infrastructure to the changing requirements of the courses. In many cases, the spaces to be used were not originally designed to be classrooms, or were built as classrooms but do not fulfill the latest teaching requirements.

  - *Rental of off-site premises.* Decisions on whether to rent classrooms located off the campus where this is possible in order to satisfy predicted increases in course demand.

  - *Construction of new infrastructure.* Decisions on whether to construct a new building or campuses if the demand for space has grown significantly. New infrastructure can be built in one or more stages depending on the scale of the construction and the changes in the requirements of the courses. The number of classrooms to be constructed and their type in terms of size, layout and technological equipment must also be decided.

- **Instructional staff decisions.** Decisions in this area determine the number and type of instructors that must be hired to teach the courses offered. In this work instructional staff requirements are defined only in terms of teaching load, which may not coincide with the needs of planned research or other academic activities.

The proposed approach assumes that the objective of the capacity planning exercise is to minimize a total cost function. This objective function may include the following terms: (1) total cost of remodeling classrooms over time; (2) total cost of built infrastructure maintenance; (3) total cost of off-site classroom rentals; (4) total fixed costs for the construction of a phase of new infrastructure; (5) total cost of building new classrooms; and (6) total payroll cost of instructional staff. Note that since the model is intended for long-term planning, the cost function parameters incorporate discount rate to capture the present values of the various objective function cost terms.

In what follows we introduce the main elements of the problem and the conditions and constraints that will be considered in the proposed model.

4.3.1 Programs and courses

The demand for instructor-seat-hours is associated with the activities of the courses in a program of study. Each such program is made up of a series of interconnected courses leading to an academic degree. The structure or configuration of a program determines which courses a set of students must register in simultaneously and what are the prerequisites for each course. Changes may be made to the program structure over the planning period.

A typical program structure is depicted by the network shown in Figure 4.1. I and F are fictitious nodes that represent the start and end, respectively, of the program. The arcs in the network represent each course's prerequisites, which can be classed as core or non-core. A core prerequisite for a course c is a course that precedes c in the same subject area and thus covers the background material for c while a non-core prerequisite for c also precedes it but is not in the same subject area and covers material that is complementary. In the figure, the core prerequisites of a given course are connected to it with a solid line whereas the non-core prerequisites are connected to it with a broken line.

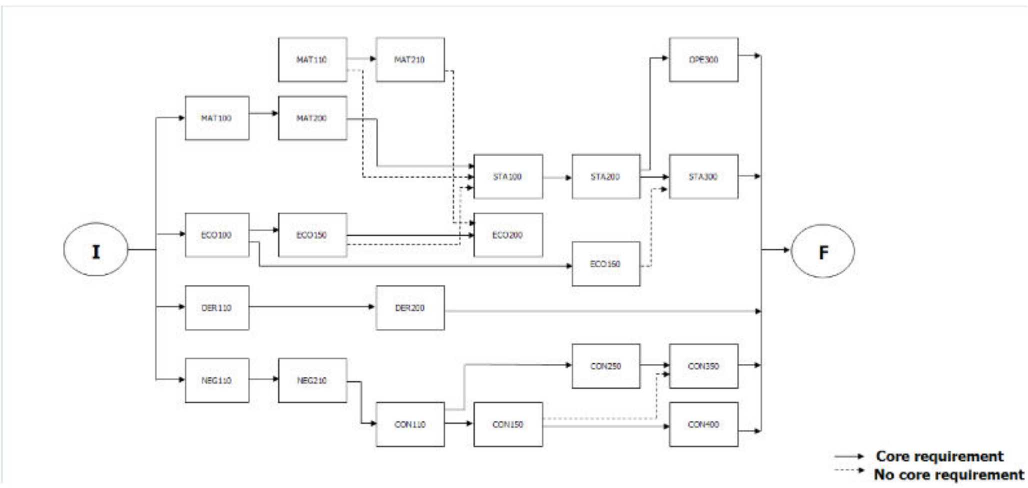


Figura 4.1: Program structure and course prerequisite relationships.

Each course in a program may have one or more sections, which may be taught by one or more professor types. Generally speaking, courses with large numbers of students will be divided into multiple sections. A course may also have different types of sessions such as lectures, tutorials or seminars, which must be scheduled at different times. In certain cases, however, a certain type of session may be scheduled at the same time for all sections of a course. For example, the tutorial sessions of every section may be scheduled at the same time so they can be used to hold tests or exams simultaneously for all students in the course. However, constraints can also be placed on the number of sessions of the same type to be scheduled in a given time slot in order to increase the students' registration options and thus give them greater flexibility. Also, courses can be classified according to (1) subject area (marketing, operations management, human resources, etc.); (2) classroom type (computer lab, flat-floor, auditorium, etc.); (3) number of weekly class hours; or (4) importance in the program (core, non-core). These characteristics implicitly determine the instructor and classroom resource requirements.

4.3.2 Class assignment and adaptation of infrastructure

We assume the institution has a set of classrooms that can be categorized into K types based on their physical characteristics such as size and layout. Size is defined by the seating capacity while layout refers to the arrangement and mobility of the seating or any special equipment that may be installed. Each course can then be assigned the type of classroom that is most appropriate for the number of students registered in it and the particular characteristics of its teaching activities.

This assignment is not unique in the sense that for a given course there may be various possible size and layout options. Preferences over these two characteristics may therefore be defined. For example, a course that could be held in a small classroom could also be held in a medium or large one, though this would result in more empty seats and in general would be considered an undesirable assignment.

One of the options for adapting infrastructure to new requirements over time is to remodel existing classrooms. For example, a large classroom could be converted into two small ones or a computer lab, or vice versa. Alternatively, if existing classrooms cannot meet seat-hour requirements, additional classrooms could be rented or a new building constructed. In the former case, construction can be spread over multiple phases as noted earlier, and in the latter case, due consideration must be given to the vagaries of the rental property market from term to term. As regards costs, the construction of a new building has a fixed cost and a variable cost component, the latter depending on the number and type of classrooms. In general we may assume that the cost of constructing classrooms is greater than that of remodeling existing ones and various times greater than that of renting extra ones. Furthermore, as the number of classrooms rises the maintenance costs such as cleaning and utilities will become increasingly significant.

4.3.3 Characteristics of the instructional staff

The institution is assumed to have a set of instructors that can be categorized into P types according to full-time/part-time status, subject area or academic rank (assistant professor, associate professor, full professor). Although instructors typically carry out research and extension activities, here we consider directly only their teaching load, which may vary depending on their type. Certain types may have higher teaching loads while others might have heavier loads in research or other activities, and this will influence the maximum number of courses they can teach. Moreover, the importance of a given course in a program may restrict which types of instructor can be assigned to it. For example, core courses might be given by full-time instructors while those of a more practical nature might be taught by part-timers who also work professionally in their fields.

Another consideration is the time slot availabilities of the different instructor types, which in turn determine the times at which certain courses can be scheduled. In practice, certain time periods tend to be preferred to others. This is a key factor given that our approach takes different time slot availabilities into account in the capacity planning process.

4.4. Forecasting of course demand

The proposed solution approach has a planning horizon of S terms and includes two sequential stages to forecast course demand. The first stage (subsection 4.4.1) estimates the number of students who enroll in the first term of each program offered by the institution using a time-series model [145], while the second stage (subsection 4.4.2) estimates the number of students who register in each course in each term (except the first) of the various programs using a network flow model similar to the one used by [127]. This model incorporates information on the prerequisites of the courses, the interactions between programs and the courses' historical failure rates.

4.4.1 Forecasting the number of students entering the first term of a program

There are various ways of building a model to forecast the number of students enrolling in the first term of a program. Since this mechanism is not the focus of the present work, we employed the relatively simple and widely used multivariate time-series VAR(p) model with several lags [145].

The model predicts the number of students entering term s (Y_n^s) as a function of a series of factors or explanatory variables (X_n^s). Among these factors are: (1) the historical number of students enrolled in the program, that is, the lags of the variable to be predicted (Y_n^{s-t} with $t = 1, ..., s-1$); (2) the maximum number of places offered in each program and term; (3) the number of students enrolling in related programs; (4) the number of institutions planned for the jurisdiction's education system over time; (5) the historical time series of the minimum and maximum entrance exam scores of admitted students; (6) the number of secondary school graduates in the jurisdiction; and (7) the total number of applicants. The objective variable Y_n^s is given by the VAR(p) model as follows:

$$\vec{Y}_n^s = \Gamma + \sum_{\mathrm{i}=1}^{p} \theta_{\mathrm{i}} \vec{Y}_n^{s-\mathrm{i}} + u_n^s, \tag{4.1}$$

where $\vec{Y}_n^s = (Y_n^s, X_n^s)$' is a vector that contains Y_n^s, the dependent variable to be predicted which is defined as the number of students that will enroll in the first term s in program n, and X_n^s, a matrix that contains all the remaining dependent variables; Γ is a constant; $\vec{Y}_n^{s-\mathrm{i}}$ are the lags of the historical series Y_n^s and X_n^s, respectively; and u_n^s is the error term.

The output of the model is the input data for predicting the number of students per course and the number of courses to be scheduled. This in term will be used to estimate the number of students per course in each term.

4.4.2 Forecasting the number of students per course in each subsequent term

The second stage of the proposed approach determines the number of students who register in each course of each program in each term subsequent to the first one just derived above in 4.4.1. Generally, a student registering in a course must have passed a series of previous courses of which one can be identified as a core prerequisite and the others as non-core prerequisites.

The predictive model in this stage is based on two fundamental assumptions:

1. The number of students registering in course c in a given term s is explained by the number of students who passed the core requisite course for c in the previous term and the number who failed c in term $s-2$ if it is an annual course or $s-1$ if it is offered every term. For simplicity we assume that non-core requisites do not significantly effect course c registration.
2. The failure rate of each course holds constant between terms s and $s+2$. In other words, the proportion of students passing or failing course c is assumed not to change significantly from one term to the next.

The model itself is a network flow model that represents the program structure by a graph $G[N, A]$ in which each node $n \in N$ is a course and each arc $a \in A$ is a prerequisite (see Figure 4.1). If we define A_c^s as the number of students registering in course c in term s, r_c^s la as the failure rate of c in term s, $\bar{c}$ as the core prerequisite of c and η_c^{s+1} as the number of special registrants in c, the number of students predicted to register in c in term $s+1$ is given by

$$A_c^{s+1} = \lceil (1 - r_{\bar{c}}^s) A_{\bar{c}}^s + r_c^s A_c^s + \eta_c^{s+1} \rceil \,. \tag{4.2}$$

Note that the special students term refers to those enrolled in other programs or exchange students, the number of the latter generally being determined by the institution's administration. The third and final stage of the proposed solution approach, consisting of an MIP model that estimates the institution's capacity, is presented in the next section.

4.5. Formulation of MIP model

The main inputs to the MIP model are the courses to be taught in each term and the course registration forecasts generated by the models just discussed above. The model treats each session type making up a course as independent events and therefore with requirements that can be different. For example, if course c has lectures, tutorials and seminars, it may be that both lecture and seminar sessions must be given in an auditorium and tutorials in computer labs, and furthermore, that each of them must be taught by a different instructor type. Constraints will then be used to avoid schedule conflicts between the different session types or to make sure certain other desirable conditions are satisfied. A schedule conflict occurs when two or more courses that must be taken by the same set of students have sessions that are scheduled in the same time slot.

The following subsections set out the model's sets, parameters and decision variables together with their notation.

4.5.1 Time slot and classrooms patterns

The decision variables of the model are based on the assignment of each course to a set of time slot and classroom patterns. Time slot patterns are combinations of one or more feasible time slots determined by instructors' time slot availability while classroom patterns are combinations of one or more types of classrooms as required by the sizes and activities of the courses to be assigned to them. This grouping of time slots and classrooms generates modeling benefits in terms of tighter linear relaxations [125], a smaller number of variables and fewer symmetries, and also simplifies the solution of MIP models by commercial mathematical programming solvers such as Gurobi [85]. More specifically, the use of classroom patterns allows decisions on the size of course sections to be incorporated directly. If the number of students registered in a course exceeds the capacity of the largest classroom, the course will have to be divided into more than one section.

The decision variables associated with the assignment of courses to time slots and classrooms are based on patterns. In the case of time slot patterns, the model attempts to assign all sessions of a course simultaneously to one or more of them, which must satisfy (1) the course's total number of weekly hours, and (2) instructor availabilities.

The classroom patterns assign all sections of a given course to a set of classrooms of one or more sizes or layouts. This set must satisfy the following conditions: (1) have a total seating capacity at least as large as the number of students registered in the course; and (2) be of an appropriate type for the course activities. Note that the classroom patterns implicitly determine the size of a course's sections given that pattern contains a set of classrooms of equal or different sizes and therefore a certain total number of seats. A course that has 200 registered students, for example, could be assigned to a pattern containing 4 classrooms seating 50 students each or 3 classrooms with 70 seats each.

4.5.2 Model sets and parameters

We assume that all the weeks in a term are equal. The scheduling of a term then becomes just the scheduling of a standard week that is replicated over all the weeks in the term.

Figure 4.2 shows a standard week of 5 days with 6 time slots each.

To describe the standard week, the following sets are defined:

- T: The set of time slots making up the entire week. The first time slot on Monday, for example, is referred to as Mon.1 (see Figure 4.2).
- B: The set of time slot patterns in which courses can be scheduled. The element $b \in B$ is a pattern consisting of a combination of one or more time slots $t \in T$. For example, the pattern (Tue.4,Fri.4) is a pattern with two sessions that must be scheduled for the fourth time slot on Tuesdays and Fridays (see Figure 4.2).
- $B_t \subseteq B$: The set of time slot patterns containing time slot $t \in T$. This set will be used in the formulation of certain constraints.

	Mon	Tue	Wed	Thu	Fri
1					
2					
3					
4					
5					
6					

Figura 4.2: The time slots of a standard week.

For each course c and term s the following sets and parameters are also defined:

- C_s: The set of all courses to be scheduled in term s.
- C_{sk}: The set of courses in term s that can use a classroom of type $k \in K$.
- H_s: The set of course families whose schedules must not conflict in term s. For example, an element $h \in H_s$ can consist of all courses in a given program that must not be scheduled in the same time slot.
- φ_{cs}: The number of students registered in course $c \in C_s$ in term s.
- λ_{cs}: The maximum number of sections of course $c \in C_s$ that can be scheduled in the same time slot in term s.

As regards classrooms, we assume that those which are remodeled, rented or newly constructed in term s can be used beginning with term $s + 1$. To describe the classrooms, the following sets and parameters are defined:

- K: The set of classroom types.
- R_k: The set of classrooms of type k belonging to the institution at the start of the planning horizon. The R_k constitute a partition of the total set of classrooms $R = \cup_k R_k$.
- L_{cs}: The set of feasible classroom patterns for course $c \in C_s$ in term s.
- K_{cs}: The set of classroom types that can be assigned to course $c \in C_s$ in term s. Note that $c \in C_{sk} \iff k \in K_{cs}$.
- α_l: The number of classrooms in classroom pattern $l \in L$.

- ϕ_l: Total number of seats in classroom pattern $l \in L$.
- θ_{lk}: The number of classrooms of type $k \in K$ in classroom pattern $l \in L$.
- β_{ks}: The maximum number of classrooms of type $k \in K$ that can be rented in term s.
- γ: The maximum number of stages for constructing a new building within the planning horizon.

To describe the instructors, the following sets and parameters are defined:

- P: The set of instructor types.
- P_{cs}: The set of instructor types that can teach course $c \in C_s$ in term s.
- C_{ps}: The set of courses a type $p \in P$ instructor can teach in term s. Note that $p \in P_{cs} \iff c \in C_{ps}$.
- E_s: The set of courses that can be taught by more than one instructor type in term s.
- B_{ps}: The set of time slot patterns that are feasible for a type $p \in P$ instructor teaching a course in term s. This set determines the time slot availabilities for each instructor type $p \in P$ in term s.
- δ_p: The number of type $p \in P$ instructors on the institution's staff at the start of the planning horizon.
- π_{ps}: The maximum number of courses a type $p \in P$ instructor can teach in term s.
- ψ_{cps}: The minimum number of type $p \in P$ instructors that must be assigned to course $c \in C_s$ in term s.

The model's objective function uses the following parameters:

- Y_s: The fixed cost of a single stage in the construction of a new building in term s.
- F_{ks}: The unit cost of constructing a new type $k \in K$ classroom in term s.
- A_{ks}: The unit cost of renting a type $k \in K$ classroom in term s.
- $Q_{k'ks}$: The unit cost of remodeling a classroom from type $k' \in K$ to type $k \in K$ in term s.
- L_{ks}: The term maintenance cost of a classroom of type $k \in K$ in term s.
- G_{ps}: The term payroll cost of a type $p \in P$ instructor in term s.

4.5.3 Decision variables

The decision variables in the proposed model are the following:

- x^p_{cbs}= The number of type p instructors that teach course c in time slot pattern b in term s.

-

$$z_{cbs}^{l} = \begin{cases} 1 & \text{If course } c \text{ in time slot pattern } b \text{ uses classroom pattern } I \text{ in term } s \\ 0 & \text{otherwise.} \end{cases}$$

- e_{ks} = The number of type k classrooms that will be rented in term s.

- w_{ks} = The number of type k classrooms that will be constructed in term s.

- $u_{k'ks}$ = The number of classrooms that will be remodeled from type k' to type $k \in K$ in term s.

- v_{ks} = The number of type k classrooms that will not remodeled in term s.

- n_{ps}= The number of type p instructors that will be needed in term s.

-

$$r_{cbs} = \begin{cases} 1 & \text{If any section of course } c \text{ is scheduled in time slot pattern } b \text{ in term } s. \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{d}_s = \begin{cases} 1 & \text{If a stage of a new building is constructed in term } s. \\ 0 & \text{otherwise.} \end{cases}$$

4.5.4 Model description

With the notation defined in the above subsections, the mixed integer programming model can now be set out as follows:

$$\text{mín} \sum_{s=1}^{S} Y_s \mathrm{d}_s + \sum_{k\in K}\sum_{s=1}^{S} F_{ks}w_{ks} + \sum_{k\in K}\sum_{s=1}^{S} A_{ks}\mathrm{e}_{ks} + \sum_{s=1}^{S}\sum_{k,k'\in K} Q_{k'ks}u_{k'ks} + \sum_{k\in K}\sum_{s=1}^{S} L_{ks}(v_{ks} + \sum_{k'\in K} u_{kk's}) + \sum_{s=1}^{S}\sum_{p\in P} G_{ps}n_{ps} \tag{4.3}$$

s.t.

$$\sum_{l\in L_{cs}} \phi_l z_{cbs}^{l} \geq \varphi_{cs} \quad \forall\ c \in C_s, \forall\ s = 1, ..., S. \tag{4.4}$$

$$x_{cbs}^{p} \leq n_c r_{cbs} \quad \forall\ c \in C_s,\ \forall\ b \in B,\ \forall\ p \in P,\ \forall\ s = 1, ..., S. \tag{4.5}$$

$$\sum_{c\in h}\sum_{b\in B_t} r_{cbs} \leq 1 \quad \forall\ h \in H_s, \forall\ t \in T,\ \forall\ s \in S. \tag{4.6}$$

$$\sum_{p\in P_{cs}}\sum_{b\in B_t} x_{cbs}^{p} \leq \lambda_{cs} \quad \forall\ c \in C_s,\ \forall\ t \in T,\ \forall\ s = 1, ..., S. \tag{4.7}$$

$$\sum_{p\in P_{cs}} x_{cbs}^{p} = \sum_{l\in L_{cs}} \alpha_l z_{cbs}^{l} \quad \forall\ c \in C_s,\ \forall\ b \in B, \forall\ s = 1, ..., S. \tag{4.8}$$

$$\sum_{l\in L_k}\sum_{c\in C_s}\sum_{b\in B_t} \theta_{lk} z_{cbs}^{l} \leq v_{ks-1} + \sum_{k'\in K} u_{k'ks-1} + w_{ks-1} + \mathrm{e}_{ks-1} \quad \forall\ k \in K,\ \forall\ t \in T, \forall\ s = 1, ..., S. \tag{4.9}$$

$$v_{ks} + \sum_{k'\in K} u_{kk's} = v_{ks-1} + \sum_{k'\in K} u_{kk's-1} + w_{ks-1} \quad \forall\ k \in K,\ \forall\ s = 1, ..., S. \tag{4.10}$$

$$v_{k0} + \sum_{k' \in K} u_{kk'0} = R_k \quad \forall\, k \in K. \tag{4.11}$$

$$\mathrm{e}_{ks} \le \beta_{ks} \quad \forall\, k \in K, \forall\, s = 1, ..., S. \tag{4.12}$$

$$w_{ks} \le M\mathrm{d}_s \quad \forall\, k \in K, \forall\, s = 1, ..., S. \tag{4.13}$$

$$\sum_{s \in S} \mathrm{d}_s \le \gamma \tag{4.14}$$

$$\sum_{c \in C_s} \sum_{b \in B_t \cap B_s(p)} x^p_{cbs} \le \delta_p + n_{ps} \quad \forall\, p \in P, \forall\, t \in T, \forall\, s = 1, ..., S. \tag{4.15}$$

$$\sum_{c \in C_s} \sum_{b \in B_{ps}} x^p_{cbs} \le \pi_{ps} \quad \forall\, p \in P, \forall\, s = 1, ..., S. \tag{4.16}$$

$$\sum_{b \in B_{ps}} x^p_{cbs} \ge \psi_{cp} \quad \forall\, c \in E_s, \forall\, p \in P, \forall\, s = 1, ..., S. \tag{4.17}$$

$$x^p_{cbs}, z^k_{cbs}, v_{ks}, u_{kk's}, w_{ks} \in \mathbb{N} \quad \forall\, c \in C_s, \forall\, b \in B, \forall\, p \in P, \forall\, k, k' \in K, \forall\, s = 1, ..., S. \tag{4.18}$$

$$r_{cbs}, \mathrm{d}_s \in \{0, 1\} \quad \forall\, c \in C_s, \forall\, b \in B,\ \forall s = 1, ..., S. \tag{4.19}$$

The objective function of the model, given by (4.3), is a cost function with six terms. The first term is the total fixed cost of one or more stages in the construction of a new building. The second term is the total construction cost of new classrooms of each type $k \in K$. The third term is the total rental cost of off-site classrooms while the fourth term is the total cost of remodeling a type $k \in K$ classroom. The fifth term is the total classroom maintenance cost, and finally, the sixth term is the total payroll cost of instructional staff over the planning horizon.

Constraint (4.4) ensures that the classroom pattern assigned to a course has sufficient capacity to seat the total number of students registered in it. A pattern's capacity is defined by parameter ϕ_l while the number of registered students in a course is set by parameter φ_{cs}.

Constraint (4.5) identifies whether, in term s, any section of course $c \in C_s$ is scheduled in timetable pattern $b \in B$, activating binary variable r_{cbs}. This variable is used to impose the elimination of schedule conflicts among a family of courses $h \in H_s$ **v**ia constraint (4.6), which prevents schedule conflicts between courses that must be taken by the same set of students. For example, a family $h \in H_s$ may be a set of courses in a single term s or consecutive terms (s and either s-1 or s+1), or they may be courses represented by the different types of sessions.

Constraint (4.7) imposes that no more than λ_{cs} sections of course c may be scheduled in each time slot $t \in T$ in term s. This ensures that courses with relatively large numbers of sections are assigned to a variety of time slots, thus giving students greater flexibility.

Constraint (4.8) relates decision variable x^p_{cbs}, which counts the number of instructors, to decision variable z^l_{cbs}, which assigns a single classroom pattern to each course. This restriction imposes that the number of instructors teaching the various sections of course $c \in C_s$ in time slot pattern $b \in B$ in term s be equal to the number of classrooms used for teaching these sections. Also, $\alpha_l z_{cbs}^l$ counts the number of classrooms to be used for c in time slot pattern $b \in B$ in s.

Constraint (4.9) identifies the number of new and rented classrooms and the number that will be remodeled in a given term in order to meet the demand for all the courses. The left-hand side of the constraint identifies the total number of classes of type $k \in K$ used to teach the various sessions of all the courses in each time slot by calculating the term $\theta_{lk} z_{cbs}^l$ while the right-hand side identifies the number of classrooms that will satisfy the demands for classroom space in each term. The type k classroom requirement in term s is satisfied by assigning: (i) classrooms of type k that were not remodeled in term s-1 (v_{ks-1}); (ii) all classrooms of type $k' \neq k$ that were remodeled in term s-1 ($\sum_{k' \in K} u_{k'ks-1}$); (iii) new classrooms constructed in term s-1 (w_{ks-1}); and (iv) classrooms rented in term s-1 (e_{ks-1}). Constraint (4.10) is the flow conservation condition between consecutive terms for each type k classroom. More specifically, the left-hand side quantifies the total number of type k classrooms available in term s, which must be equal to the total number of classrooms available in the previous term ($s-1$). Constraint (4.11) imposes the boundary condition for the start of the planning period and constraint (4.12) imposes that the maximum number of type k classrooms that can be rented in s cannot exceed β_{ks}. A diagram depicting the flow conservation condition for a type k classroom between consecutive terms is shown in Figure 4.3.

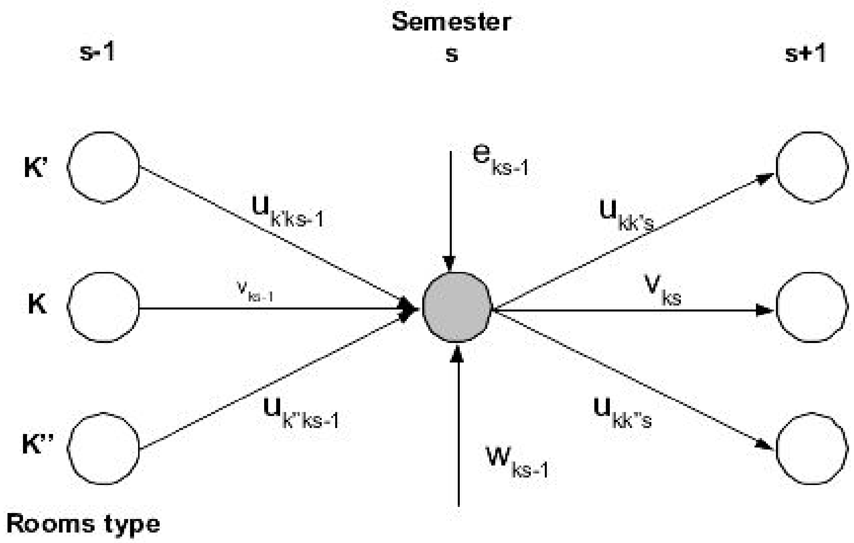


Figura 4.3: Flow conservation for a type k classroom between consecutive terms.

Constraint (4.13) determines when a new building construction stage should be initiated. This occurs when the demand for classroom space cannot be met by rented or remodeled classrooms of type $k \in K$ in term s, in which case binary variable d_s is activated. Parameter M is a constant set by the modeler. Constraint (4.14) limits the number of construction stages to a level within the planning horizon.

As regards instructors, constraint (4.15) counts the total number of instructors of type

$p \in P$ that are needed to teach all the courses to be scheduled in term s. If the number at the start of the planning horizon (δ_p) is exceeded, integer variable n_{ps} is activated. Set $B_s(p)$ defines the time slot availabilities of instructors of type p in s. Constraint (4.16) sets the maximum number of courses a type p instructor can teach in s to an amount π_{ps}. Conditions ensuring that certain courses have a minimum number of a given type of instructor were also incorporated into the model. For example, it may be desirable to require that the core courses of a program always have at least one full-time instructor. Constraint (4.17) imposes that for courses that may be taught by different types of instructor (E_s), there will always be at least ψ_{cp} instructors of type p assigned in term s. Finally, constraints (4.18) and (4.19) define the nature of the model variables.

4.6. Description of real case

The University of Chile's School of Economics and Business (FEN in its Spanish initials) is one of the most prestigious schools in its field in Latin America [108] and the demand for admission to its programs is among the highest of any university in the country. Out of more than 4,000 applications received in 2012, for example, only about 450 were accepted. It may be safely assumed that such institutions will always be able to fill the number of places set by the administration. In the case of FEN, the final number of students admitted has historically been slightly higher than the number of places offered each year (see Figure 4.4) due to special student enrollments and factors relating to the admitted students' maximum and minimum entrance exam scores.

4.6.1 Characteristics of FEN programs and courses

FEN offers four programs: (1) Business (Bus); (2) Economics (Eco); (3) Management control and information systems (Mcis); and (4) Accounting (Acc). Each program has 70 to 76 courses, averaging in total more than 300 course sections per term. To ensure all students receive a similar basic preparation, a large part of the courses in the first four terms are common to all of the programs. This increases the potential for timetable conflicts and thus considerably complicates the process of scheduling time slots and classrooms.

The courses are categorized according to (a) their importance in the programs (core or non-core), and (b) their subject area (operations management, economics, marketing, etc.). These characteristics of the courses determine to a large extent the type of professor required to teach them. The numbers of core and non-core courses in each program are shown in Table 4.1; the individual courses total 167 (after eliminating multiple-counting of courses common to more than one program) and are distributed by subject area in Figure 4.2.

All courses are scheduled in 90-minute time slots of which there are six per day from Monday through Friday, for a total of 30 slots per week. The sessions for each course may be scheduled in either one of four time slot pattern families: (a) Family 1, for courses with

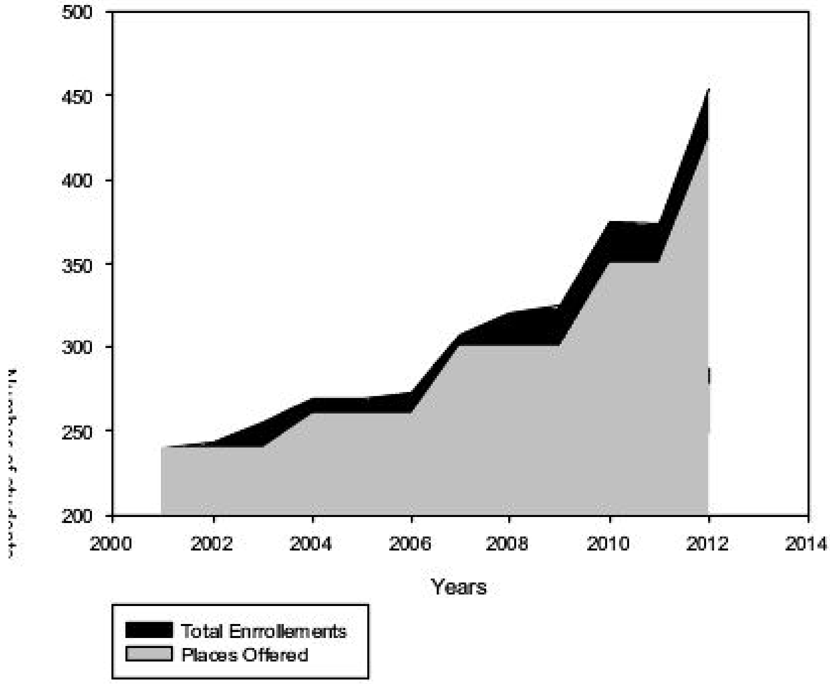


Figura 4.4: Places offered by FEN versus total FEN enrollment during the period 2001-2012.

Programs	Core	Non-core	Total
Bus.	55	18	73
Eco.	54	19	73
Mcis	58	18	76
Acc.	58	12	70
Total	225	67	292

Tabla 4.1: Number of core and non-core courses by program.

Subject area	Courses		Subject area	Courses
Auditing	6		Strategy	4
Accounting	22		Marketing	6
Management control	6		Mathematics	8
Strategy	4		Commerce	12
Economics	37		Operations management	7
Statistics	7		MIS	21
Human Resources	8		Taxes	7
Electives	5		English	6
Entrepreneurship	1			

Tabla 4.2: Number of courses by subject area.

one session per week that may be scheduled in any time slot; (b) Family 2, for courses with two sessions per week on different days but in the same time slot. Permitted day configurations for Family 2 are Monday and Thursday or Tuesday and Friday; (c) Family 3, for courses with two consecutive time slots on a single day (e.g., Monday.2-Monday.3); and (d) Family 4, for courses with three sessions per week on different days but in the same time slot. Permitted day configurations for Family 4 are Monday, Wednesday and Thursday or Tuesday, Wednesday and Friday (e.g., Monday.2- Wednesday.2-Thursday.2). The numbers and percentages of courses using these different patterns are shown in Table 4.3. Tutorials are scheduled as a Family 1 pattern.

Time slot pattern	Courses	% of total
1	179	54.74 %
2	137	41.90 %
3	5	1.53 %
4	6	1.83 %
Total	327	100 %

Tabla 4.3: Number and percentage of courses in each time slot pattern.

FEN course sessions can be of two types, lectures and tutorials, though some only have lectures. If a course with tutorial sessions has more than one section, the tutorials for all of them are scheduled in a single weekly time slot so that exams can be held for all students simultaneously.

4.6.2 Characteristics of classrooms

FEN has 35 classrooms categorized by size and layout. The size classifications are: medium (M), for 40 to 50 seats; large (L), for 51 to 65 seats; and very large (VL), for 66 to 100 seats. The layout classifications are: flat-floor (F), horseshoe (H), auditorium (A) and computer laboratory (Lab). The layout options are designed to meet the needs of the different types of course activities. For example, horseshoe classrooms facilitate face-to-face interaction and are thus used mainly for courses that involve discussion and analysis of case studies while in flat-floor classrooms, tables and chairs can be rearranged easily for working in groups. Thus, for a given course the desired classroom type can be assigned to each type of session, or alternatively, it can be specified that any type of classroom may be assigned. Table 4.4 shows the sizes and layouts of FEN's current stock of classrooms while Table 4.5 indicates the number of School courses assigned to each classroom type.

4.6.3 Characteristics of instructional staff

FEN has 80 full-time (FT) and 205 part-time (PT) instructors as of 2013. Core courses with only one section must be taught by a full-time instructor, but those with multiple

Size \ Layout	F	H	Lab	A	Total
M	2	–	4	–	6
L	8	7	–	3	18
VL	3	4	–	4	11
Total	13	11	4	7	35

Tabla 4.4: Number of classrooms by size and layout.

Classroom type	Courses	% of total
Horseshoe	110	65.87 %
Laboratory	10	5.98 %
Flat-floor	9	5.39 %
Any type	38	22.76 %
Total	167	100 %

Tabla 4.5: Number and percentage of courses by type of classroom used.

sections may have just one section given by a full-timer and the rest by part-timers. In such cases, the full-time instructor has the additional role of course coordinator to ensure consistency across all sections. Non-core courses may be given by either full-time or part-time instructors. The distribution of FEN's instructional staff by full-time/part-time status and subject area is set out in Table 4.6.

Subject area	Code	FT	PT
Accounting	AC	9	13
Entrepreneurship	EN	1	5
Strategy	ST	3	5
Finance	FI	6	8
Marketing	MK	9	8
Management	MN	3	8
Human Resources	HR	5	8
MIS	MO	6	18
Management control	MC	5	10
Economics	EC	33	57
English	EN	0	13
Mathematics	MA	0	34
Statistics	ST	0	18
	Total	80	205

Tabla 4.6: Number of instructors by subject area in 2013.

Historically, the time slot availabilities of full-time instructors have been concentrated in time slots 2, 3 and 5, which have accounted for more than 80 % of availabilities in the last 6 terms. For part-time instructors, on the other hand, who are often combining their teaching hours with other professional activities, availabilities have been concentrated in time slots 1, 4 and 6, that is, early in the morning, at lunch-hour and at the end of the standard working day.

4.7. Real case experiments

In this section we present the results of the experimental application of our approach to the real case just described above. Although the planning horizon was 8 years, we analyzed only the autumn term of each year when on average 5 % more courses are offered than in the spring term and the number of schedule conflicts is also greater. The experiments assumed an annual rise in the number of places in each program of 10 %.

As regards the construction of new classrooms, the three possible sizes were: (1) medium, 50 seats; (2) large, 65 seats; and very large, 90 seats. The three possible layouts were flat-floor, horseshoe and computer laboratory. The construction cost of a laboratory is twice that of a horseshoe classroom and four times that of a flat-floor classroom of the same size. Remodeling was considered to affect only a classroom's layout, not its size, and because of laboratories' special technical characteristics they cannot be remodeled from other layouts but must be purpose-built. The cost of remodeling a classroom was assumed, in light of estimates based on real data, to be 20 % of the construction cost of a new one. Rentals were not considered due to FEN internal policies. The term payroll cost of a full-time instructor is four times the cost of a part-time one regardless of the subject taught (note that all cost data in this paragraph were defined in consultation with FEN).

4.7.1 Forecasting of course demand

To forecast first term enrollment a multivariate time series model VAR(p) time series model was built for each FEN program. The explanatory variables of the model were: (1) the lags of the dependent variable, that is, the historical number of enrollments in program (Y_n^s); (2) the total number of applicants (Pos_{Bus}^s); (3) the number of places offered each term (Vac_n^s); and (4) the historical time series of the maximum and minimum entrance exam scores of admitted students ($PMax_n^s$) and ($PMin_n^s$), respectively. As an example, the model estimated for the business (Bus.) program is shown in Table 4.7. The number of lags is $p = 2$ and the variables associated with places offered (Vac_{Bus}^s) and minimum scores were omitted due to multicollinearity, the first being highly correlated with the objective variable and the second with maximum scores ($PMax_{Bus}^s$). For all programs combined, the projection of enrollments in the first term of each year is shown in Figure 4.5.

The enrollment in each course c and term s was estimated using the model presented in Section 4.4.2. The failure rate for each course was derived from historical information and calculated as the arithmetic average of the rates for the last 6 years. The proportion of enrollment registering as special students was very small (less than 1 %) and for simplicity's sake was ignored. The number of events to be scheduled for each session type over the planning horizon is shown in Table 4.8.

Variable	Coeff.	Std. Err.	95 % Conf. Interval	
Y_{Bus}^{s-1}	0.74	1.26E-14	0.74	0.75
Y_{Bus}^{s-2}	1.22	1.03E-14	1.22	1.23
Pos_{Bus}^{s-1}	-0.03	4.70E-16	-0.03	-0.02
Pos_{Bus}^{s-2}	-0.01	6.68E-16	-0.02	-0.01
Vac_{Bus}^{s-1}	(omitted)			
Vac_{Bus}^{s-2}	(omitted)			
$PMax_{Bus}^{s-1}$	-2.3	9.54E-15	-2.23	-2.22
$PMax_{Bus}^{s-2}$	0	9.13E-15	-1.79E-14	1.79E-14
$PMin_{Bus}^{s-1}$	(omitted)			
$PMin_{Bus}^{s-2}$	(omitted)			
Γ	1,642.77			

Tabla 4.7: Enrollment forecast estimates and statistical test results

	Lectures	Tutorials	Total events
2014	201	192	393
2015	220	210	430
2016	236	226	462
2017	256	244	500
2018	273	260	533
2019	300	286	586
2020	333	317	650
Total	1,819	1,735	3,554

Tabla 4.8: Number of events to be scheduled over planning horizon.

FEN imposes the section sizes a course can be partitioned into based on which term in the program a course is given, the idea being that section sizes should get smaller as students advance through the program years. Note that this definition implicitly determines the size of the classroom that must be assigned to the course. The general section size maxima set by the School are shown by term in Table 4.9, with some exceptions being defined where teaching or methodological considerations so dictated. For example, courses in English language had a section maximum of 25 students.

Term	Size section
1-4	Very large (90) or Large (65)
5-8	Large (65) or medium (50)
9-10	Medium (50)

Tabla 4.9: Permitted values for course section size partitions.

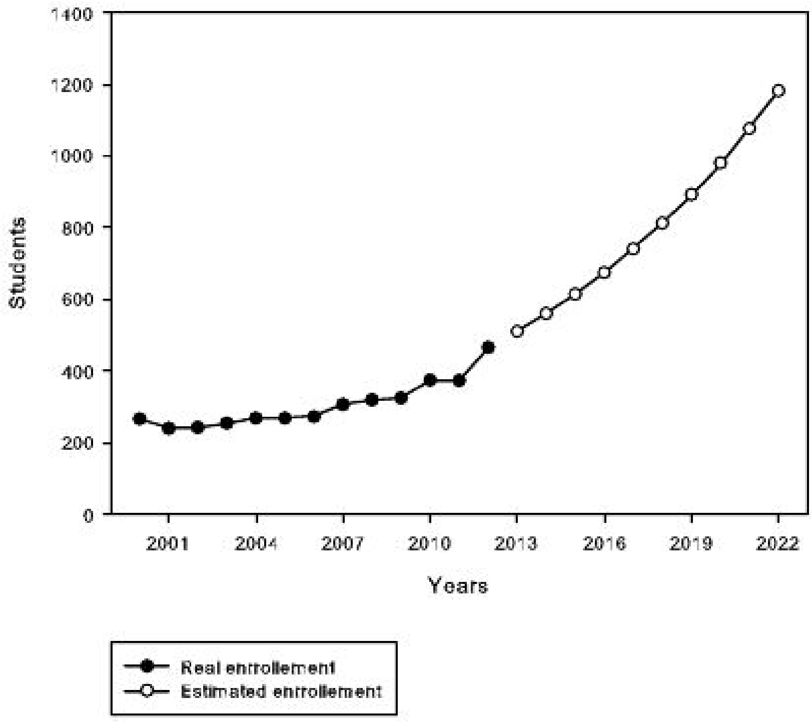


Figura 4.5: Total enrollment forecast over the planning horizon.

4.7.2 Results of the MIP model

In this subsection we present the results of the MIP model developed in Section 4.5. We begin by describing the scenarios determined by the different timetable availabilities of the instructor types (full time, part time). Then we set out the impacts of these availabilities on costs, the demand for construction of new classrooms and instructional staff size, respectively. Finally, we examine the changes to the existing classroom infrastructure between the start and the end of the planning horizon.

Instructor time slot availability scenarios

Four instructor time slot availability scenarios were analyzed:

1. Scenario 1 (S1): The baseline scenario reflecting FEN's current operation. Full-time instructors are available only for time slots 2, 3 and 5 while part-time instructors are available only for time slots 1, 4 and 6. This scenario is the most restrictive of the four in that the total number of time slots in which instructors can teach is the lowest.
2. Scenario 2 (S2): Full-time instructors are available only for time slots 2, 3 and 5, as in the baseline scenario, but part-time instructors have complete flexibility to teach in any of the 6 time slots.

3. Scenario 3 (S3): Full-time instructors have complete flexibility to teach in any of the 6 time slots but part-time instructors are available only for time slots 1, 4 and 6, as in the baseline scenario.

4. Scenario 4 (S4): Both full-time and part-time instructors have complete flexibility to teach in any of the 6 time slots. This scenario is the most flexible of the four in that both instructor types can teach in any time slot.

The optimization problems for each scenario were solved using the Gurobi 3.0.1 [85] mathematical programming solver running on a Dell Power Edge 1900 computer with a 1066 MHz processor and 8 GB of RAM. The size of the problems solved in terms of numbers of variables and constraints, the integrality gap and the computational time of the best solution found are shown in Table 4.10. The stopping criterion was a maximum time of 120 or GAP equal to 10 %.

	S1	S2	S3	S4
CPU Time (hr.)	54.46	53	85	120
Discrete variables	2,516,869	2,576,863	2,586,127	2,662,518
Constraints	147,157	122,027	147,157	149,411
Gap	9.7 %	9.4 %	9.6 %	12.2 %

Tabla 4.10: Characteristics of problem instances by scenario.

Impact of instructor timetable availabilities on costs

Instructor timetable availabilities had a considerable impact on the total costs of capacity planning. As can be seen in Table 4.11, there was a difference of 40 % between S1, the most restrictive scenario, and S4, the most flexible one. More specifically, in S4 the remodeling cost was higher but the construction cost was lower. This strongly suggests that S4 adapted to changing course by undertaking more infrastructure remodeling and less new construction.

As regards instructor payroll cost, the same comparison is observed: S4 had the lowest payroll and S1 the highest. This may be explained by the fact that the total number of instructors was subject to their availabilities (Constraint 15) so that the fewer the availabilities the greater the number of required instructors for a given time slot and therefore the higher the term payroll cost. The complete data for the objective function cost terms together with their values and the cost differences between the baseline and the other scenarios in both absolute and percentage terms are summarized in Table 4.11.

	S1	S2	S3	S4
Construction cost	4,887	3,907	3,421	2,959
Remodelling cost	729	810	1162	921
Maintenance cost	856	783	696	699
Payroll cost	184	180	173	166
Objective function (M$US)	6,656	5,680	5,452	4,745
Difference (M$US)	0	976	1,204	1,911
Difference (%)	0	17 %	22 %	40 %

Tabla 4.11: Objective function costs ($MUS) by term and scenario.

Impact of instructor time slot availability on new classroom construction

Instructor time slot availabilities had a considerable impact on the number of new classrooms to be constructed. In general terms, the lower is instructor availability, the greater is the number of new classrooms to be built. In the present case, the number for each scenario is given in Table 4.12. In the baseline scenario 25 classrooms were to be constructed while in S2, S3 and S4 the numbers to be built were 21, 18 and 14, respectively.

Forma	S1	S2	S3	S4
F	6	4	4	3
H	15	13	11	8
Lab	4	4	3	3
Total	25	21	18	14

Tabla 4.12: Number of new classrooms to be constructed by layout and scenario.

With these figures, the cost of inflexibility in the timetable availability of instructor types can be estimated. For example, it was found that the increase in flexibility of part-time instructors in S2 reduced new construction compared to the baseline scenario by 4 classrooms. The increase in flexibility of full-time instructors in S3 resulted in 7 fewer classrooms to be constructed compared to the baseline, and the flexibility increase in both instructor types in S4 brought the greatest reduction of all at 11 classrooms. These results stem from the fact that with less availability, demand for space in certain time slots is greater.

Impact of instructor time slot availability on number of instructors

Time slot availability does not significantly affect the total number of instructors needed each term over the planning horizon. The reason is that the number of courses to be taught each term is fixed and the growth in instructional staff adapts to these numbers regardless of instructor availability. This can be appreciated in Figure 4.6, which shows a sustained growth in the number of instructors that closely follows the increase in the number of courses year on year.

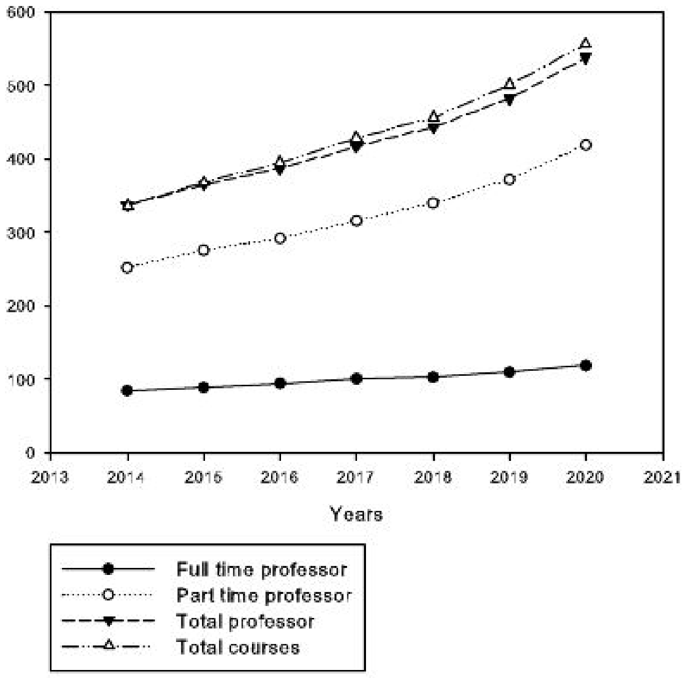
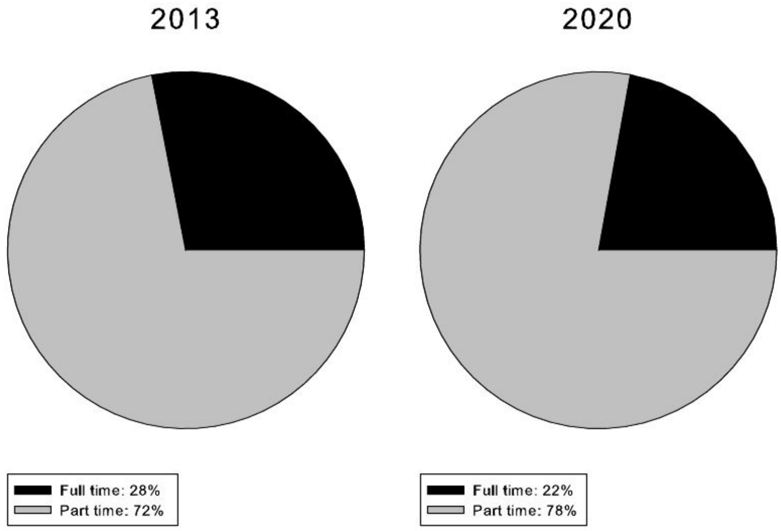


Figura 4.6: Trend in number of instructors over planning horizon for scenario S4.

There were, however, significant differences in the growth rate of each instructor type. The growth in part-timers was clearly higher than for full-timers, a phenomenon explained by the fact that for courses with more than one section that can be taught by both instructor types, only one full-timer is required, and due to the cost (i.e., payroll) differences part-timers will be preferred for the remaining sections.

The distribution of FEN's instructional staff by full-time/part-time status is depicted in Figure 4.7 for both instructor types. As can be seen, the proportion of part-time staff was projected by the model to rise over the planning horizon from 72 % in 2013 to 78 % in 2020. Figure 4.8 shows that the distribution of instructors by subject area will remain unchanged over the same period. This suggests that the instructor types were well-aligned with instructor requirements in terms of course numbers.

Figura 4.7: Distribution of instructors by full-time/part-time status in S4 over planning horizon (2013-2020).



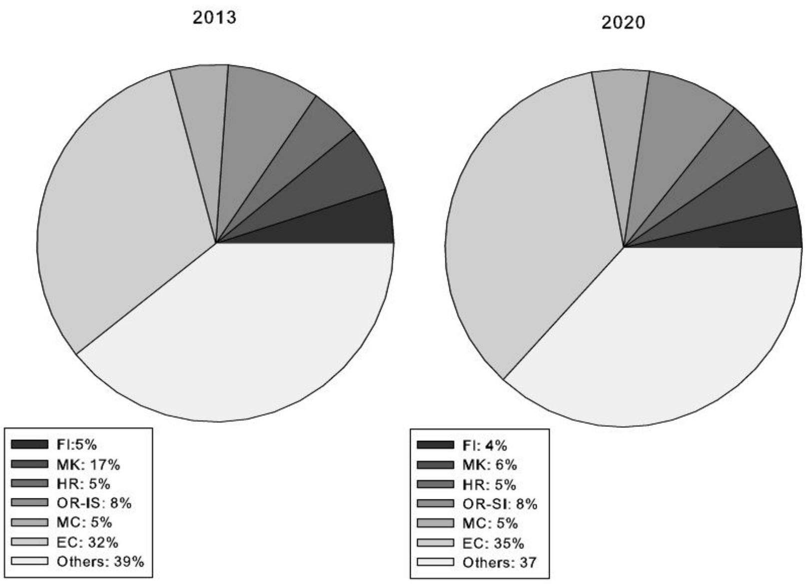


Figura 4.8: Distribution of instructors by subject area in S4 at start and end of planning horizon (2013-2020).

Remodeling of classroom types

We now examine the results of the decisions on remodeling of existing classroom infrastructure. The numbers of classrooms by physical characteristics for S4 is set out in Table 4.13. The first and second digits in each cell represent the number of classrooms in the indicated size and layout combination at the start and at the end, respectively, of the planning horizon. By the end of the horizon new classrooms have been built and some existing ones have been remodeled, resulting in changes to the initial distribution of classroom types. As an example, if a cell reads "2/3", there were 2 classrooms of the indicated size and layout at the beginning of the period and 3 by the end of it.

Size\Form	F	H	Lab	Total
M	2/3	0/6	4/4	6/13
L	8/2	7/13	0/2	15/17
VL	3/3	4/8	0/1	7/12
Total	13/8	11/27	4/7	28/42

Tabla 4.13: Number of classrooms by size and layout in S4 at start and end of planning horizon.

The initial and final distribution of the classrooms by layout are shown in Figure 4.9. The pie charts reveal a jump in horseshoe-type classrooms from 39 % of the total at the start of the planning horizon to 64 % at the end. Over the same period, computer labs increased from 14 % to 17 % of the total while flat-floor classrooms declined dramatically from 47 % to just 19 %. The results for classroom distribution by size, given in Figure 4.10, show that whereas large classrooms fell from 54 % of the total to 40 % over the planning horizon, medium classrooms rose considerably from 21 % to 31 % and very large classrooms grew less significantly from 25 % to 29 %. These outcomes demonstrate how the proposed approach adapts the existing infrastructure to the classroom requirements of the courses (see Table 4.5).

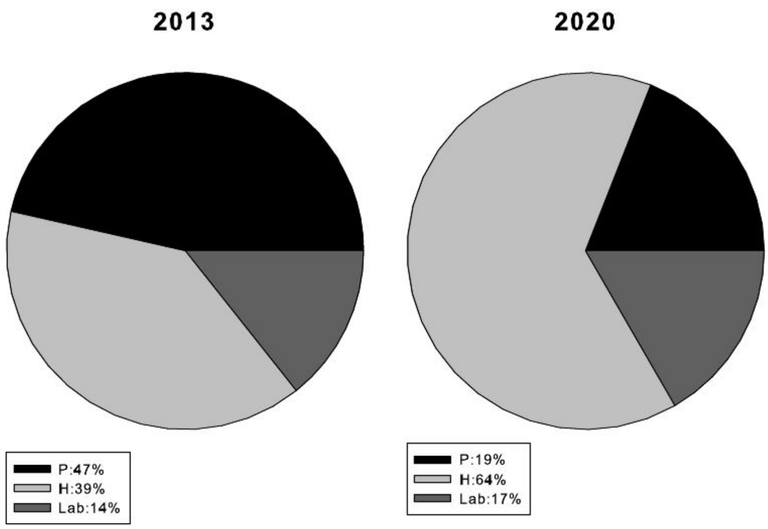


Figura 4.9: Distribution of classrooms by layout in S4 at start and end of planning horizon.

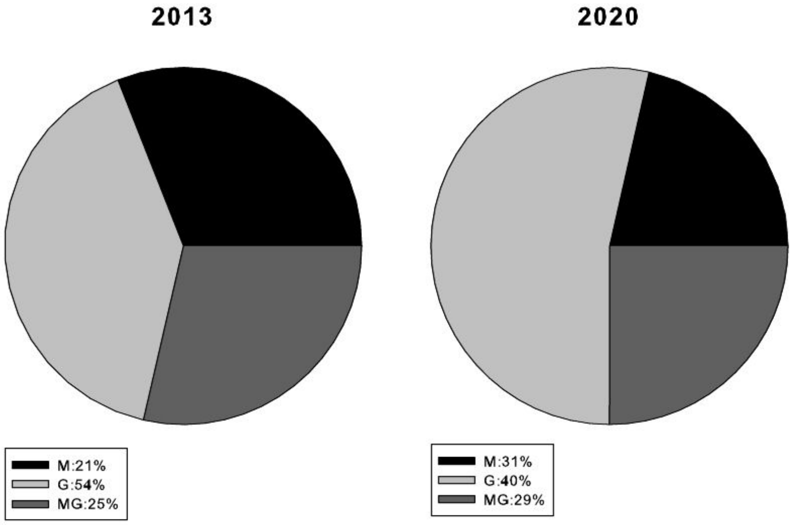


Figura 4.10: Distribution of classrooms by size in S4 at start and end of planning horizon.

4.8. Conclusions

This article presented a solution approach culminating in an MIP model that generates essential information for supporting the capacity planning process of a higher education institution. The methodology builds on two preliminary forecasting models that provide estimates of total demand for “instructor-seat-hours.ºver time. The first of these formulations estimates the number of enrollments in the first year of each program of study offered by the institution while the second one calculates the number of students expected to register in each program course. The MIP model is then inputted with these estimates and solved, and the results are used to guide key capacity planning decisions regarding the construction, rental and remodeling of infrastructure and the appointment of instructional staff.

The approach was applied to a real case of capacity planning at the School of Economics and Business of University of Chile and various instructor timetable availability scenarios were tested. The results showed that considering the number of instructors was directly related to the number of courses, the impact of the availability scenarios was not significant. But restrictive scenarios did significantly increase the institution’s costs, for reasons relating more to a consequent overbuilding of infrastructure than to higher instructor payrolls. Yet despite the real influence on costs, it was found that the model indicated a change in the distribution of instructor types, with a rise in the relative weight of part-time versus full-time instructors.

As for the infrastructure itself, it was also demonstrated that the need for different classroom types evolves considerably over time and that classroom planning must consider the importance of adapting infrastructure policies to the changing needs of course characteristics and enrollments. A failure to keep classroom construction, rental and remodeling decisions aligned with these needs would risk transforming the planning process into a costly exercise fraught with complications negatively impacting the institution over the long run. Regarding future research, exact methods based on branch and price could be applied to the capacity planning problem to reduce the gaps and solution times of the MIP model. Additional factors such as the sale of unused or underused infrastructure could be added to the model. Finally, the proposed approach lends itself to extensions and adaptations to other real-world capacity planning situations including hospitals, road toll systems and airlines.

Bibliografía

[1] S. Abdennadher and M. Marte. University course timetabling using constraint handling rules. *Applied Artificial Intelligence*, 14:311–325, 2000.

[2] S. Abdullah. *Heuristics approaches for university timetabling problems*. PhD thesis, The School of Computer Science and Information Technology, University of Nottingham, 2006.

[3] D. Abramson, H. Krishnamoorthy, and H. Dang. Simulated Annealing cooling schedules for the school timetabling problem. *Asia Pacific J Operational Res*, 16:1–22, 1996.

[4] P. Adamidis and P. Arapakis. Evolutionary algorithms in lecture timetabling. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1145–1151. IEEE Press, 1999.

[5] S. Al-Yakoob and S. Sherali. Mathematical programming models and algorithms for a class-faculty assignment problem. *European Journal of Operational Research*, 173:488–507, 2006.

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

[7] R. Alvarez-Valdes, E. Crespo, and J. Tamarit. Assigning students to course sections using Tabu Search. *Annals of Operations Research*, 96:1–16, 2000.

[8] R. Alvarez-Valdes, E. Crespo, and J. Tamarit. Design and implementation of a course scheduling system using Tabu Search. *European Journal of Operational Research*, 137:512–523, 2002.

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

[10] P. Avella and L. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8(6):497–514, 2004.

[11] R. F. Averill and J. Suttle. A model for university enrollment planning. *Socio-Economic Planning Sciences*, 9(6):257 – 261, 1975.

[12] M. Badri. A two-stage multiobjective scheduling model for [faculty-course-time] assignments. *European Journal of Operational Research*, 94:16–28, 1996.

[13] M. Badri, D.L. Davis, D.F. Davis, and J. Hollingsworth. A multi-objective course scheduling model: Combining faculty preferences for courses and times. *Computers and Operations Research*, 25:303–316, 1998.

[14] D.S. Bajwa, A. Rai, and I. Brennan. Key antecedents of executive information system success: a path analytic approach. *Decision Support Systems*, 22(1):31–43, 1998.

[15] C.A. Bana, J. C. Lourenco, M.P. Chagas, and J. C. Bana. Development of reusable bid evaluation models for the portuguese electric transmission company. *Decision Analysis*, 5(1):22–42, 2008.

[16] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P. H. Vance. Branch and price: column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[17] C. O. Benjamin, I. C. Ehie, and Y. Omurtag. Planning facilities at the University of Missouri-Rolla. *Interfaces*, 22(4):95–105, 1992.

[18] E.W. Bessent and A.M. Bessent. Student flow in a university department: Results of a markov analysis. *Interfaces*, 10(2):52–59, 1980.

[19] C. Beyrouthy, E.K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A.J. Parkes. Improving the room-size profiles of university teaching space. preprint, October 2007.

[20] C. Beyrouthy, E.K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A.J. Parkes. The teaching space allocation problem with splitting. In E.K. Burke and H. Rudova, editors, *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling (PATAT 2006)*, pages 228–247, Berlin Heidelberg, 2007. Springer-Verlag.

[21] C. Beyrouthy, E.K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A.J. Parkes. Thershold effects in teaching space allocation problem with splitting. preprint, April 2008.

[22] C. Beyrouthy, E.K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A.J. Parkes. Towards improving the utilization of university teaching space. *Journal of the Operational Research Society*, 60:130–143, 2009.

[23] C. Beyrouthy, E.K. Burke, B. McCollum, P. McMullan, and A.J. Parkes. University space planning and space type profiles. *Journal of Scheduling*, 13:363–374, 2010.

[24] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[25] R. Bixby, J. Gregory, I. Lustig, R. Marsten, and D. Shanno. Very large-scale linear programming: a case study in combining interior point and simplex methods. *Operations Research*, 40(5):885–897, 1992.

[26] Barbara Bleau. Planning models in higher education: Historical review and survey of currently available models. *Higher Education*, 10:153–168, 1981.

[27] N. Boland, B. Hughes, L. Merlot, and P. Stuckey. New integer linear programming approaches for course timetabling. *Computers and Operations Research*, 35:2209–2233, 2008.

[28] B. Bollobás. *Random graphs*. Cambridge University Press, second edition, 2001.

[29] S. Boomfields and M.MacSharry. Preferencial course scheduling. *Interfaces*, 9(4):24–31, 1979.

[30] J. Boronico. Quantitative modeling and technology driven departamental course scheduling. *Omega*, 28:327–346, 2000.

[31] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: forecasting and control*. Wiley, 4th edition, 2008.

[32] P.T. Brinkman and C. McIntyre. Methods and techniques of enrollment forecasting. *New Directions for Institutional Research*, 1997(93):67–80, 1997.

[33] P. Brucker, R. Qu, and E. Burke. Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3):467 – 473, 2011.

[34] E. Burke, Y. Bykov, and S. Petrovic. A multi-criteria approach to eamination timetabling. In Burke and Erben [37], pages 118–131.

[35] E. Burke and M. Carter, editors. *The Practice and Theory of Automated Timetabling II: Selected Papers from the 2nd International Conference on the Practice and Theory of Automated Timetabling*. LNCS 1408, Springer-Verlag, 1998.

[36] E. Burke and P. De Causmaecker, editors. *The Practice and Theory of Automated Timetabling IV:Selected papers from the 4th International Conference (PATAT IV)*. LNCS 2740, Springer-Verlag, 2003.

[37] E. Burke and W. Erben, editors. *Practice and Theory of Automated Timetabling:Selected papers from the 3rd International Conference*. LNCS 2079, Springer-Verlag, 2001.

[38] E Burke, K Jackson, JH Kingston, and R Weare. Automated university timetabling: The state of the art. *Computer Journal*, 40(9):565–571, 1997.

[39] E. Burke, G. Kendall, and E. Soubeiga. A Tabu Search hyper-heuristic for timetabling and rostering. *Jornal of Heuristics*, 9(6):451–470, 2003.

[40] E. Burke, B. MacCarthy, S. Petrovic, and R. Qu. Knowledge discovery in a hyper-heuristic for course timetabling using case-based reasoning. In *In: Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT 2002)*, pages 276–287, 2003.

[41] E. Burke, J. Marecek, A. Parkes, and H. Rudova. A branch and cut procedure for the udine course timetabling problem. *Annals of Operations Research*, 194:71–87, 2012.

[42] E. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu. A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176:177–192, 2007.

[43] E. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266 – 280, 2002.

[44] E. Burke and H. Rudova, editors. *PATAT 2006: Proceedings of The 6th International Conference on the Practice and Theory of Automated Timetabling*, Masaryk University, Brno, República Checa, 2006. Disponible en la página `http://patat06.muni.cz/doc/PATAT_2006_Proceedings.pdf`.

[45] E. Burke and M. Trick, editors. *The Practice and Theory of Automated Timetabling V: Selected Revised Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling*. LNCS 3616, springer-Verlag, 2005.

[46] M.P. Carrasco and M.V. Pato. A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. *European Journal of Operational Research*, 153(1):65 – 79, 2004.

[47] M. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34:192–202, 1986.

[48] M. Carter. A comprehensive course timetabling and student scheduling system at the university of waterloo. In Burke and Erben [37], pages 64–82.

[49] M. Carter and G. Laporte. Recent developments in practical course timetabling. In *In: Practice and theory of automated timetabling (PATAT) II, vol 1408. Springer, Berlin*, 1998.

[50] M. W. Carter and C. A. Tovey. When is the classroom assignment problem hard? *Operations Research*, 40:28–39, 1992.

[51] S. Casey and J. Thompson. GRASPing the examination scheduling problem. In Burke and Causmaecker [36], pages 232–244.

[52] P. De Causmaecker, P. Demeester, and G. Vanden Berghe. A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, 195(1):307 – 318, 2009.

[53] L. C. Chambers, editor. *The Practical Handbook of Genetic Algorithms*, volume 1. CRC Press, 1995.

[54] N. Chantal and D. de Werra. Extension of coloring models for scheduling purposes. *European Journal of Operational Research*, 40:32–37, 1989.

[55] S.-M. Chena and C.-C. Hsub. A new method to forecast enrollments using fuzzy time series. *Int. J. Appl. Sci. Eng.*, 23:234–244, 2004.

[56] M. Chiarandini, M. Birattari, K. Socha, and O. Rossi-Doria. An effective hybrid algorithm for university course scheduling. *Journal of Scheduling*, 9:403–432, 2006.

[57] R.B. Cooper and R.W. Zmud. Information technology implementation research: a technological diffusion approach. *Management Science*, 36(2):123–139, 1990.

[58] D. Corne, P. Ross, and H. Fang. *Evolving timetables*, pages 219–276. Volume 1 of Chambers [53], 1995.

[59] J. Correa and M. Wagner. Lp-based online scheduling: from single to parallel machines. *Mathematical Programming*, 119:109–136, 2009.

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

[61] S. Daskalaki, T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153:117–135, 2004.

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

[63] D. de Werra. Extension of coloring models for scheduling purposes. *European Journal of Operational Research*, 92:474–492, 1996.

[64] D. de Werra. The combinatorics of timetabling. *European Journal of Operational Research*, 96:504–513, 1997.

[65] J. Van den Broek, C. Hurkens, and G. Woeginger. Timetabling problems at the TU Eindhoven. *European Journal of Operational Research*, 196(3):877 – 885, 2009.

[66] M. Dimopoulou and P. Miliotis. Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, 130:202–213, 2001.

[67] M. Dimopoulou and P. Miliotis. An automated university course timetabling system developed in a distributed environment: A case study. *European Journal of Operational Research*, 153:136–147, 2004.

[68] M. L. Downie. Efficiency outcomes from space charging in UK higher education estates. *Property Management*, 3(1):33–42, 2005.

[69] G. Dueck. New optimization heuristics, the Great Deluge algorithm and the redord to record travel. *Journal of Computational Physics*, 104:86–92, 1993.

[70] G. Durán, M. Guajardo, J. Miranda, D. Sauré, S. Sourys, and R. Wolf. Scheduling the chilean soccer league by integer programming. *Interfaces*, 34(6):539–552, 2007.

[71] J. Dyer and J. Mulvey. An integrated optimization/information system for academic departmental planning. *Management science*, 2(12):1332–1341, 1976.

[72] B. Eksioglu, A. Volkan Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers and Industrial Engineering*, 57(4):1472 – 1483, 2009.

[73] M. Elmohamed, P. Coddington, and G. Fox. A comparison of annealing techniques for academic course scheduling. In Burke and Carter [35], pages 92–112.

[74] A.T Ernst, H Jiang, M Krishnamoorthy, and D Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3 – 27, 2004.

[75] J. Ferland and C. Fleurent. SAPHIR: A decision-support system for course scheduling. *Interfaces*, 24:105–115, 1994.

[76] J. Ferland and S. Roy. Timetabling problem for university as assignment of activity to resources. *Computers and Operational Research*, 12(2):207–218, 1985.

[77] P. Fizzano and S. Swanson. Scheduling classes on a college campus. *Computational Optimization and Applications*, 16:279–294, 2000.

[78] J. J. Forrest and J. A. Tomlin. Vector processing in Simplex and interior point methods for linear programming. *Annals of Operations Research*, 22:71–100, 1990.

[79] L. Foulds and D. Jonhson. SlotManager: a microcomputer-based decision support system for university timetabling. *Decision Support Systems*, 27:367–381, 2000.

[80] L. Di Gaspero and A. Schaerf. Multi-neighbourhood local search with application to course timetabling. In Burke and Causmaecker [36], pages 262–275.

[81] P.C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859, 1961.

[82] M.J. Ginzberg. Early diagnosis of mis implementation failure: promising results and unanswered questions. *Management Science*, 27(4):459–478, 1981.

[83] H. Goltz and D. Matzke. University timetabling using constraint logic programming. In *Practical Aspects of Declarative Languages*, volume 1551 of *LNCS*, pages 320–334. Springer-Verlag, 1999.

[84] T. Guimaraes, Y. Yoon, and A. Clevenson. Factors important to expert system success: a field test. *Information and Management*, 30(3):119–130, 1996.

[85] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2012.

[86] K. Haase, H. Scheel, and D. Sebastian. Management of lecture-rooms. model, method and intenet-based application for efficient course scheduling. *Wirtschaftsinformatik*, 46:87–95, 2004.

[87] E. Hartono, R. Santhanam, and C.W. Holsapple. Factors that contribute to management support system success: An analysis of field studies. *Decision Support Systems*, 43:256–268, 2007.

[88] T. Hinkin and G. Thompson. SchedulExpert: Scheduling courses at Cornell University School of Hotel Administration. *Interfaces*, 32:45–57, 2002.

[89] D. Hopkins. On the use of large-scale simulation models for university planning. *Review Of Educational Research*, 41:467–478, 1971.

[90] IBM. *Users Manual CPLEX 12.0.* IBM, 2012.

[91] IBM ILOG. CPLEX optimization studio web page, 2010. http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio (accessed: 12-10- 2010).

[92] D. Johnson. Timetabling university examinations. *The Journal of the Operations Research Society*, 41:39–47, 1990.

[93] D. Johnson. A database apprroach to course timetabling. *The Journal of the Operations Research Society*, 44:425–433, 1993.

[94] G. Kendall, S. Knust, C. Ribeiro, and S. Urrutia. Scheduling in sports: An annotated bibliography. *Computers and Operations Research*, 37(1):1 – 19, 2010.

[95] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, November 2009.

[96] G. Laporte and S. Desroches. The problem of assigning students to course sections in a large engineering school. *Computers and Operations Research*, 13(4):387 – 394, 1986.

[97] S. M. Lee and E. R. Clayton. A goal programming model for academic resource allocation. *Management Science*, 18(8):B395–B408, 1972.

[98] J. Leung and J.H. Anderson. *Handbook of Scheduling: Algorithms, Models, and*

Performance Analysis. Chapman and Hall/CRC, 2004.

[99] J.B. Levine. Application of the campus simulation models to the major planning decisions of a large university. In *Proceedings of the second conference on Applications of simulations*, pages 317–328. Winter Simulation Conference, 1968.

[100] R. Lewis. A survey of metaheuristics-based techniques for university timetabling problems. *OR Spectrum*, 30:167–190, 2008.

[101] T.P. Liang. Development of a knowledge-based model management systems. *Operation Research*, 36:849–863, 1988.

[102] T.P. Liang, C.C. Lee, and E. Turban. Model management and solvers for decision support. In *Handbook on Decision Support Systems*, volume 1, pages 231–258. Springer-Verlag, 2008.

[103] Z. Lu and J. Hao. Adaptive abu Search for course timetabling. *European Journal of Operational Research*, 200(1):235 – 244, 2010.

[104] Y. Ma, Ch. Chu, and Ch. Zuo. A survey of scheduling with deterministic machine availability constraints. *Computers and Industrial Engineering*, 58(2):199 – 211, 2010.

[105] S. Mansmann and M.H. Scholl. Decision support system for managing educational capacity utilization. *IEEE Transactions on Education*, 50:143–150, 2007.

[106] K. Marshall and R. Oliver. A constant work model for student attendance and enrollment. *Operations Research*, 18(2):193–206, 1970.

[107] H. Mausser, M. Magazine, and J. Moore. Application of an annealed neural network to a timetabling problem. *INFORMS Journal on Computing*, 8(2):103–117, 1995.

[108] América Economía. Ranking de mejores universidades en chile, 2012. Acceded:06-12-2012.

[109] Education Space Management Group. Space utilisation: practice, performance and guidelines. Technical report, UK Higher Education Space Management Group, 2006. Acceded: 06-12-2012.

[110] B. McCollum. The implementation of a central timetabling system in a large british civic university. In Burke and Carter [35], pages 237–253.

[111] F. Melicio and J. Caldeira. Timetabling implementation aspects by Simulated Annealing. *In: Jifa Gu (ed), IEEE systems science and systems engineering. Beijing. Aceite*, pages 553–557, 1998.

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

[113] J. Miranda, D. Espinoza, and R. Weber. A mip approach for infrastructure capacity planning in higher education: The effect of instructor timetable availabilities. *Submited to Management Science*, 2014.

[114] J. Miranda, P. A. Rey, and J.M. Robles. udpskeduler: A web architecture based decision support system for course and classroom scheduling. *Decision Support Systems*, 52(2):505 – 513, 2012.

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

[116] E. Mooney, R. Rardin, and W. Parmenter. Large-scale classroom scheduling. *IIE Transactions*, 28:369–378, 1996.

[117] R. Nasser, M. Ayob, G. Kendall, and R. Qu. A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 216(3):533 – 543, 2012.

[118] M. Oprea. Mas_up-uct: A multi-agent system for university course timetabling scheduling. *International Journal of Computer Communications and Control*, 2:94–102, 2007.

[119] B. Paetcher, R. Rankin, and A. Cumming. Improving a lecture timetabling system for university wide-use. In Burke and Carter [35], pages 156–165.

[120] B.B. Pal, D. Chakraborti, and P. Biswas. A genetic algorithm method to fuzzy goal programming formulation based on penalty function for academic personnel management in university system. In *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on*, pages 1 –10, july 2010.

[121] B.B. Pal and S. Sen. A linear goal programming procedure for academic personel management problems in university system. In *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*, pages 1 –7, dec. 2008.

[122] R. Perzina. Solving the university timetabling problem with optimized enrolment of students by parallel self-adaptative genetic algorithm. In Burke and Rudova [44], pages 246–263. Disponible en la página `http://patat06.muni.cz/doc/PATAT_2006_Proceedings.pdf`.

[123] S. Petrovic and E.K. Burke. *The Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter Chapter 45: University Timetabling. CRC Press, 2004.

[124] P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks. Stochastic optimisation timetabling tool for university course scheduling. *International Journal of Production Economics*, 112:903–918, 2008.

[125] A. Qualizza and P. Serafini. A column generation scheme for faculty timetabling. In E.Burke and M. Trick, editors, *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2005)*, pages 161–173, 2005.

[126] S. Rahman, A. Bargiela, E.K. Burke, E. Zcan, Barry McCollum, and Paul McMullan. Adaptive linear combination of heuristic orderings in constructing examination timetables. *European Journal of Operational Research*, 232(2):287 – 297, 2014.

[127] B. Ram, S. Sarin, and A. Mallik. A methodology for projecting course demands in academic programs. *Computers Ind. Engng.*, 12:99–103, 1987.

[128] P. Rattadilok, A. Gaw, and R. Kwan. Distributed choice function hyper-heuristic for timetabling and scheduling. In E.Burke and M. Trick, editors, *In: Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, pages 51–67, 2005.

[129] L. Raymond. Organization characteristics and mis success in the context of small business. *MIS Quarterly*, 9(1):37–52, 1985.

[130] G. Reeves and E. P. Hickman. Assigning MBA students to field study project teams: A multicriteria approach. *Interfaces*, 22(5):52–58, 1992.

[131] O. Rossi-doria, M. Sampels, M. Birattari, M. Chiar, M. Dorigo, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Statzle. A comparison of the performance of different metaheuristics on the timetabling problem. In *In: Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT 2002*, pages 329–351, 2002.

[132] H. Rudova and K. Murray. University course timetabling with soft constraints. In Burke and Causmaecker [36], pages 310–328.

[133] Anna Saiti. Management science in higher education institutions: Case studies from greece. *Interchange*, 41:45–60, 2010.

[134] R.M. Saltzman and T.M. Roeder. Simulating student flow through a college of business for policy and structural change analysis. *Journal of the Operational Research Society*, 63(4):511–523, 2012.

[135] G.L. Sanders and J.F. Courtney. A field study of organizational factors influencing DSS success. *MIS Quarterly*, 9(1):77–91, 1985.

[136] K. Sandhu. *Automating class schedule generation in the context of a university timetabling information system.* PhD thesis, School of Managementm, Griffith University, 2001.

[137] R. Santhanam, T. Guimaraes, and J. George. An empirical investigation of DSS impact on individuals and organizations. *Decision Support Systems*, 30(1):51–72,

2000.

[138] A. Schaerf. Local search techniques for high-school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics*, 29(4):368–377, 1999.

[139] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127, 1999.

[140] K. Schimmelpfeng and S. Helber. Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29(4):783–803, 2007.

[141] R. Schoeder and C. Adams. The effective use of management science in university administration. *Review of Education Research*, 46(1):117–131, 1976.

[142] R. G. Schroeder. A survey of management science in university operations. *Management science*, 19(8):895–906, 1973.

[143] R. G. Schroeder. Resource planning in university management by goal programming. *Operations Research*, 22(4):700–710, 1974.

[144] D.B. Seo and A.I. La Paz. Exploring the dark side if IS in achieving organizational agility. *Communications of the ACM*, 51:136–39, 2008.

[145] C. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980.

[146] K. Socha, M. Sampels, and M. Manfrin. Ant algorithm for the university course timetabling problem with regard to the state-of-the-art. In *Applications of evolutionary computing*, volume 2611 of *LNCS*, pages 334–345. Springer-Verlag, 2003.

[147] C. Soh, S.S. Kien, and J. Tay-Yap. Culture fits and misfits: Is erp a universal solution. *Communications of the ACM*, 43:47–51, 2000.

[148] Space-Management. Room utilization summary. Report, Space Management and Academic Scheduling department, 2009. Acceded: 06/12/2012.

[149] R.H. Sprague and E.D. Carlson. *Building effective decision spport systems*. Prentice Hall Professional Technical Reference, 1982.

[150] J. Stallaert. Automated timetabling improves course scheduling at UCLA. *Interfaces*, 27(4):67–81, 1997.

[151] Sam W. Stigall. Recommendations to improve space projection models and university space usage. *New Directions for Institutional Research*, 2007(135):29–36, 2007.

[152] D. Strnad and N. Guid. A multi-agent system for university timetabling. *Applied Artificial Intelligence*, 21:137–153, 2007.

[153] W. Stronge and R. Schultz. Models for projecting school enrollment. *Educational Evaluation and Policy Analysis*, 3:75–81, 1981.

[154] M. Sun. A multiple objective programming approach for determining faculty salary equity adjustments. *European Journal of Operational Research*, 138(2):302 – 319, 2002.

[155] D. Teodorović and E. Krčmar-Nožić. Multicriteria model to determine flight frequencies on an airline network under competitive conditions. *Transportation Sciences*, 21(1):14–25, 1989.

[156] T.Guimaraes, M. Igbaria, and M. Lu. The determinants of DSS success:an integrated model. *Decision Sciences*, 23(2):409–424, 1992.

[157] A. Tripathy. School timetabling – A case in large binary integer linear programming. *Management Science*, 30(12):1473–1489, 1984.

[158] A. Tripathy. Computerised decision aid for timetabling: a case analysis. *Discrete Applied Mathematics*, 35(3):313 – 323, 1992.

[159] Virginia-State. Space utilization report. Report, State Council of Higher Education for Virginia, 2000.

[160] Y. Wang. Using genetic algorithm methods to solve course scheduling problems. *Expert Systems with Applications*, 25:39–50, 2003.

[161] W. Weiler. A model for short-term institutional enrollment forecasting. *The Journal of Higher Education*, 51:314–327, 1980.

[162] G. White and J. Zhang. Generating complete university timetables by combining Tabu Search with constraint logic. In *PATAT 1997*, pages 187–198. Springer-Verlag, 1998.

[163] G.M. White and P.W. Chan. Towards the construction of optimal examination timetables. *IFOR*, 17:219–229, 1979.

[164] J. Witte. *Change of degrees and degrees of change: Comparing adptations of European higher education systems in the context of the Bologna process*. PhD thesis, CHEPS, Universiteit Twente, Enschede, Netherlands, 2006.

[165] D. Zhang, Y. Liu, R. MHallah, and S.C.H. Leung. A Simulated Annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203(3):550 – 558, 2010.