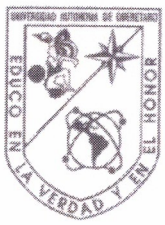


Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Ciencias (Ingeniería Matemática)

Implementación de un sistema de asignación de horarios utilizando un algoritmo híbrido basado en optimización

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Ingeniería Matemática

Presenta:

Maite Regina Benítez Escárcega

Dirigido por:

M. en C. Luisa Ramírez Granados

SINODALES

M. en C. Luisa Ramírez Granados Presidente	Firma
Dr. Luis Eduardo Urbán Rivero Secretario	Rúbrica Firma
Dr. Roberto Augusto Gómez Loenzo Vocal	Firma
Dr. José Erik Rivas Araiza Suplente	Firma
M. en C. Wilfrido Jacobo Paredes García Suplente	Firma

Dr. Manuel Toledano Ayala
Director de la Facultad

Dra. Ma. Guadalupe Flavia Loarca Piña
Directora de Investigación y Postgrado

Centro Universitario
Querétaro, QRO
México.
Mayo 2019

Abstract

At the Engineering Department of the Autonomous University of Querétaro each semester a set of undergraduate courses has to be assigned to a fixed number of classrooms. This activity is done by hand and based on experienced, therefore it is susceptible to errors. The first step is to collect the timetables of the different study programs from the Program Coordinators and then allocate courses to classrooms satisfying a list of constraints and preferences, this takes so much time that it usually had to be done during administrative vacations. A multi-objective integer programming (IP) model was design considering the department's terms and solved by programming it on Python and using Gurobi (a commercial optimization solver). The objectives are to allocate courses to rooms large enough to sit the students and to take into consideration specific room preferences. Real data from the Engineering Department's timetables and infrastructure was used to compute the assignment problem for the first semester of 2019 and the results were approved by the university's authorities.

(Keywords: integer programming, assignation problem, optimization problem solver, mathematical model.)

Resumen

En la Facultad de Ingeniería (FI) de la Universidad Autónoma de Querétaro (UAQ) se asignan por semestre un conjunto de cursos de licenciatura a un número determinado de salones de clases. Esta actividad se realiza a mano y con base a la experiencia por lo que es susceptible a errores. Después de que los coordinadores de los diferentes programas de estudio han establecido los horarios de clases, se realiza la asignación de salones satisfaciendo una lista de restricciones y preferencias lo cual solía tomar tanto tiempo que se realizaba durante vacaciones administrativas. Se diseñó un modelo de programación entera multi-obejtivo que considera las condiciones del departamento, se resolvió con un programa escrito en Python y utilizando Gurobi (un solucionador comercial para problemas de optimización). Los objetivos son mantener los cursos que pertenecen a la misma licenciatura en la menor cantidad de edificios posibles y relacionar cursos con salones de acuerdo al tamaño de ambos. Se utilizaron datos de horarios reales e infraestructura de la FI-UAQ para realizar la asignación del periodo 2019-1 obteniendo el visto bueno de las autoridades universitarias.

(Palabras clave: programación entera, problema de asignación, solucionador de problemas de optimización, modelo matemático.)

Esta tesis está dedicada a todos y todo lo que formó parte de este periodo, lo disfruté más de lo que imaginé.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por permitir que esta investigación fuera posible gracias a la manutención que me otorgó todos estos meses.

A la Universidad Autónoma de Querétaro por becar todas mis materias, a sus profesores que laboran con profesionalismo y pasión, y a los administrativos que con eficiencia y buena actitud facilitan tantos trámites. Agradezco a la Maestra Cecilia Garcia-Diego y a la Maestra Carmen Sosa por su constante ayuda, participación, disponibilidad y sobre todo buena fe en este proyecto.

A M.C.Luisa Ramírez Granados y al Dr.Roberto Gómez Loenzo porque gracias a su confianza en mí pude sobrellevar las inseguridades que me surgieron en las distintas etapas de la maestría.

Al Dr.Luis Eduardo Urbán Rivero por su paciencia, asesoramiento y colaboración en la programación y diseño del modelo de optimización. Sus habilidades y conocimientos permitieron convertir una idea en la acción que resolvió el problema propuesto.

A mis padres y hermano por hacerme sentir que soy capaz de todo.

A Ed por la compañía y motivación.

A mis compañeros por ser mis amigos y compartir más que tareas.

A todos mis profesores, sus clases me enseñaron más que teoría. Gracias por educar con pasión.

Índice general

Índice de figuras

Índice de cuadros

CAPÍTULO 1

Introducción

El problema de asignación de salones en las instituciones educativas requiere que cursos se ubiquen en salones considerando restricciones. La asignación se considera factible si y sólo si cumple con las restricciones, por ejemplo asignar un curso a una única aula en cierto intervalo de tiempo y una aula a un único curso en un intervalo de tiempo. Este problema se puede ver como un problema de asignación multidimensional (MAP por sus siglas en inglés) y por lo tanto es NP-completo [Carter & Tovey, 1992]. Por otro lado, las restricciones suaves sí pueden ser violadas en caso de no existir otra opción de asignación; un ejemplo sería que se especificara la preferencia de asignar varios cursos a una misma aula durante el mismo intervalo de tiempo, en este caso la restricción suave tendría que violarse para algunos de estos cursos pues por la restricción dura únicamente un curso puede estar asignado a esa aula en un mismo intervalo de tiempo. Este tipo de variables y la cantidad de combinaciones posibles hacen del problema de asignación de salones un problema NP-completo [Carter & Tovey, 1992].

Generalemente los problemas de asignación de salones se modelan con programación lineal entera y se resuelven con ramificación-y-acotamiento o con métodos heurísticos [Hillier & Lieberman, 2010]. Han sido ampliamente estudiados, principalmente el problema de asignación de horarios el cual incluye más niveles como la asignación de horarios de profesores, cursos y salones [Oude Vrielink et al., 2016, Burke & Ross, 1995].

Usualmente en las instituciones educativas la asignación de salones se realiza de manera central porque los salones se pueden compartir entre los diferentes programas de estudio [Constantino et al., 2010, Abdennadher et al., 2000, Phillips et al., 2015]. El problema varía de una institución a otra debido a la duración de los cursos, la infraestructura y las preferencias específicas como la asignación de cursos a salones especiales. Constantino et al. [Constantino et al., 2010] resolvió el problema de asignación de salones en una universidad de gran tamaño con el objetivo de minimizar la distancia total caminada por cada grupo hacia las diferentes aulas que se les asignaban en la semana. Utilizó penalizaciones cuando el tamaño de un salón no correspondía con el tamaño del grupo, y para cuando las preferencias de asignación determinadas por el área administrativa no se cumplían. El modelo y algoritmos de resolución propuestos por Constantino et al. [Constantino et al., 2010] no se pueden aplicar a la Universidad Autónoma de Querétaro (UAQ) porque los requisitos y las preferencias son distintas. Por lo tanto, para esta tesis se diseñó un modelo que define el problema de asignación de salones para la Facultad de Ingeniería (FI) de la UAQ de acuerdo a los objetivos y restricciones determinados. El modelo diseñado se probó e implementó para proponer una asignación de aulas para el periodo escolar 2019-1. A pesar de que el modelo cumplió con los objetivos de reducir el tiempo y hacer uso eficiente del espacio, no tomó en cuenta las restricciones suaves como asignar cursos a grupos de acuerdo al tamaño de ambos y relacionar licenciaturas con edificios de acuerdo a preferencias de asignación. La relación de cursos con edificios no es un acuerdo formal pero ha sido seguido por el área administrativa durante años, por lo que nos solicitaron se tratara de incluirla en el modelo y código del programa. En este documento presentamos una serie de modelos de programación entera y resultados obtenidos utilizando datos reales de la FI-UAQ, resueltos con Gurobi [Optimization, 2018] que es un software comercial de programación entera mixta.

El análisis de un problema del área de investigación de operaciones se inicia identificando cuál de los problemas típicos de optimización se asemeja al problema de estudio. Esta actividad es esencial para facilitar el diseño del modelo y posteriormente conocer los métodos de solución que resolverán nuestro modelo. Comunmente al modelar se utilizan cómo referencia problemas similares de la literatura para facilitar la representación matemática. Una vez modelado se busca el método que resuelva el modelo. Dicho método puede ser exacto o no, dependiendo de la calidad de las soluciones que se requieran.

La solución obtenida debe analizarse y de ser necesario complementarla con conocimientos prácticos para poder implementarla. Si la solución es inaceptable el modelo y/o el algoritmo de optimización deberán ser mejorados y repetir el proceso de toma de decisiones [Talbi, 2009]. El proceso clásico de toma de decisiones consiste en: formular el problema, modelarlo, solucionarlo, implementarlo y validarlo [Talbi, 2009]. En la práctica se recomienda iterar este proceso, como se muestra en la Figura 1.1, con el objetivo de mejorar el modelo de optimización y el algoritmo propuesto hasta que se encuentre una solución aceptable.

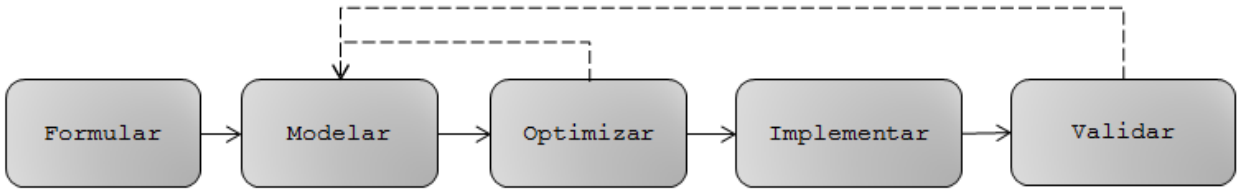


Figura 1.1: Proceso de toma de decisiones de acuerdo a Talbi [Talbi, 2009]

El problema de asignación de salones es un problema típico de la investigación de operaciones que busca asignar recursos limitados a diferentes actividades, por ejemplo: identificar qué empleados realizarán mejor ciertos trabajos de acuerdo a su aptitudes. Para plantear el modelo se necesitan las siguientes variables:

$$x_{i,j} = \begin{cases} 1 & \text{si el empleado } i \text{ realiza el trabajo } j \\ 0 & \text{otro caso} \end{cases} \tag{1.1}$$

Una matriz de costos determina los valores de que el empleado i realice el trabajo j, como se ve en el Cuadro 1.1.

Cuadro 1.1: Ejemplo de una matriz de costos para el problema de asignación de empleados a tareas.

	Tarea				
Empleado	1	2	3	4	5
1	10	11	8	9	12
2	5	7	7	4	7
3	16	14	13	18	15
4	3	5	8	7	3
5	14	9	11	12	13

Cada variable de decisión ($a_{i,j}$) se multiplica por el costo de dicha relación y se suman para generar la función objetivo, la cual se puede minimizar o maximizar según el caso.

$$minz = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j} \tag{1.2}$$

Sujeto a las siguientes restricciones:

1. Cada tarea debe ser asignada a solo un empleado.

$$\sum_{i=1}^{n} x_{i,j} = 1,\ j = 1 \dots n \tag{1.3}$$

2. Cada empleado puede realizar a lo más una tarea.

$$\sum_{j=1}^{n} x_{i,j} \leq 1,\ i = \dots n \tag{1.4}$$

Para el problema que se aborda en esta tesis, se diseñó un modelo que respeta la capacidad de cada salón, asigna un salón por curso y a la vez, un curso por salón, todo esto, considerando si los cursos deben impartirse en laboratorios, centro de cómputo, restiradores o en un conjunto específico de salones por alguna solicitud especial.

La asignación de salones de la FI-UAQ es un problema combinatorio que ha sido resuelto con métodos heurísticos o metaheurísticos, dichos métodos se han vuelto muy populares en las últimas décadas debido a que permiten obtener "buenas" soluciones en un tiempo razonable [Talbi, 2009]. La asignación de salones tienen menos niveles de asignación en comparación con el problema de asignación de horarios el cual busca la asignación de profesores a cursos en intervalos de tiempo y días a la semana.

El problema de asignación de salones es un sub-problema del problema de asignación de horarios y son pocas las investigaciones que se enfocan únicamente a la asignación de salones [Constantino et al., 2010]. El problema de asignación de salones es un problema suficientemente difícil puesto que se conoce su complejidad [Carter & Tovey, 1992]. La metodología más común es ajustar una heurística a las especificaciones de la instancia analizada [Colorni et al., 1998, Abdullah et al., 2007, Skoullis et al., 2017]. En este trabajo se presenta un modelo de programación entera para el problema de asignación de salones en la FI-UAQ; las restricciones fueron hechas a partir de los requerimientos de operación de la FI-UAQ. En las funciones objetivo se incluyeron las peticiones de preferencia de salón para ciertos cursos y la asignación acorde a la capacidad del salón. El modelo matemático se implementó en un solver de optimización comercial (Gurobi) con su interfaz de Python.

A continuación se mencionan algunos de los trabajos que investigaron el problema de asignación de salones por separado del problema de horarios en instituciones educativas. En el Cuadro 1.2 se muestra un resumen de los métodos y resultados obtenidos.

Cuadro 1.2: Métodos aplicados y resultados obtenidos en algunos de las investigaciones publicadas sobre el problema de asignación de salones en Universidades al rededor del mundo.

Autor	Año	Método	Tiempo	Número de sesiones	Número de salones	Equipo
Carter, Michael W.	1989	Relajacion Lagrangiana	15 minutos	2192	125	IBM 4341
Martinez-Alfaro y Flores-Terán	1998	Recocido simulado programado en C	14 horas	3000	190	IBM-6000
Abdennadher *et al.*	2000	Programación lógica con restricciones y un programa CHR	Pocos minutos	1000	40	-
Constantino *et al.*	2010	Método Húngaro, heurística y metaheurística	35 minutos	3978	192	PC AMD Athlon 2.4MHz 1GB RAM
Phillips *et al.*	2015	Programación entera con Gurobi 5.1	5.5 segundos	1965	72	64-bit Debian 7 3.33GHz Intel i5-2500k

Carter (1989) describió una de las primeras propuestas de solución para problemas de asignación, la cual continúa siendo relevante después de muchos años; dado un conjunto de sesiones de cursos a las que se les han asignado días y periodos de tiempo, y dada una colección de salones disponibles se debe determinar una asignación aceptable de cursos a salones de acuerdo a una variedad de factores que medirán la deseabilidad de una asignación en particular, subdividió el problema en dos componentes:

I) Una función de costo donde c_{ij} es el costo de asignar el curso i al salón j. El costo es cero si la relación entre curso i y salón j es perfecta y el costo tiene un valor alto de acuerdo a la *indeseabilidad* de la asignación. Se le da un costo infinito a las asignaciones inaceptables, por ejemplo cuando el salón es muy pequeño para el grupo. Determinar una función de costo adecuada para el problema es esencial para el buen funcionamiento del algortimo.

II) Después describe un modelo con función de costos para reflejar las preferencias generales de los profesores y la administración.

Para representar el problema de MAP consideró una restricción a la que llamó "eliminación de cambios de salón" la cual sumaba variables de curso i en diferentes salones j en periodos consecutivos $k_1 < k_2$ restringiendo la suma a un valor menor o igual a uno.

$$x_{ij_1}^{k_1} + x_{ij_2}^{k_2} \leq 1, i = 1, ..., n \quad \forall j_1 \neq j_2 \quad \forall k_1 \neq k_2 \tag{1.5}$$

donde n es el número de cursos en el subproblema (cursos impartidos en los periodos k_1 y k_2) [Carter, 1989].

Martinez-Alfaro y Flores-Terán [Martinez-Alfaro & Flores-Teran, 1998] utilizaron recocido simulado para resolver el problema de asignación de salones en el Instituto Tecnológico de Estudios Superiores de Monterrey. La universidad contaba con 190 salones de clases y alrededor de 200 periodos de clases. Consideraron la relación de cupo y tamaño, el equipo requerido por los cursos, las distancias entre los salones y el cubículo del profesor que imparte el curso, y en caso de ser edificios de varios pisos se asignaron de abajo hacia arriba. El algoritmo se programó en C y utilizaron mSQl como base de datos para los cursos, salones, y profesores. La asignación de salones se realizó en 14 horas en promedio, disminuyendo considerablemente el tiempo con el proceso manual en el cual tardaban hasta 4 meses.

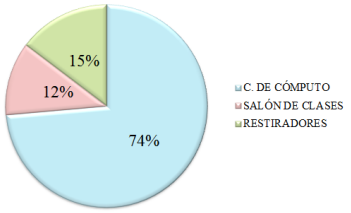
Abdennadher *et al.* [Abdennadher et al., 2000] utilizaron programación lógica con restricciones (CLP por sus siglas en inglés) la cual combina las ventajas de la programación lógica y las técnicas de resolución con restricciones. Implementaron el modelo utilizando un solucionador orientado a restricciones llamado Reglas para Manejar Restricciones (CHR por sus siglas en inglés)[Abdennadher et al., 2000]. Complementaron un solucionador de dominio finito escrito en CHR para que se adaptara a su modelo. Hicieron una interfaz escrita en JAVA a la que llamaron *RoomPlan*. Genera los datos requeridos para el solucionador CHR con Prolog facts y comienza el solucionador de CHR. Reportan que el sistema se implementó en la Universidad de Munich para 1,000 cursos y 40 salones, obteniendo un horario satisfactorio en pocos minutos en lugar de días.

Constantino *et al.* [Constantino et al., 2010] realizaron una asignación de salones en una Universidad con el objetivo de minimizar la distancia entre los salones asignados para un curso, utilizar eficientemente el espacio, satisfacer las preferencias por salones específicos y complacer otros requerimientos administrativos. Proponen dos algoritmos heurísticos basados en la resolución sucesiva para problemas lineales y de cuello de botella, y un tercer algoritmo de búsqueda de vecindades variables (VNS por sus siglas en inglés). Implementaron el algoritmo que combina el método húngaro y el algoritmo del camino de aumento más corto (CAP-A) propuesto por Carpaneto y Toth [Carpaneto & Toth, 1987] para resolver el problema lineal y el algoritmo para resolver el problema de asignación con cuello de botella de Carraresi y Gallo [Carraresi & Gallo, 1984]. El tiempo de cómputo máximo fue de 35 minutos con 27 segundos que tardó el algoritmo de VNS con 4 iteraciones, y el mínimo fue de 4 minutos con 52 segundos del CAP-A con 2 iteraciones [Constantino et al., 2010].

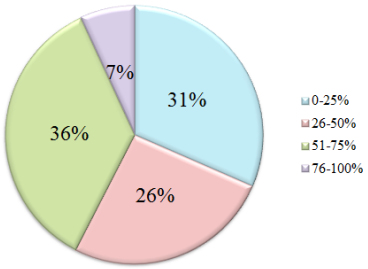
Phillips et al. [Phillips et al., 2015] diseñaron un modelo utilizando programación entera, lo probaron para dos semestres de la Universidad de Auckland y para gran parte de las instancias publicadas en la página web de la Competencia Internacional de Horarios (ITC por sus siglas en inglés) [McCollum, 2010]. Presentan gran variedad de resultados para diferentes definiciones del problema de asignación de salones y los comparan con diversos métodos publicados. Obtuvieron buenos resultados comparado con los mejores reportados, por lo tanto promueven el uso de métodos heurísticos complementados con métodos de optimización para resolver problemas difíciles. Utilizaron Gurobi 5.1 para realizar sus experimentos computacionales obteniendo resultados relativamente rápido, menos de 15 segundos, utilizado cortes y/o heurísticas.

1.1. Motivación

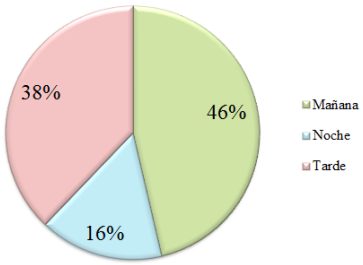
Se han propuesto numerosos métodos y modelos para resolver el problema de horarios en universidades alrededor del mundo, pero aún no se diseña un algoritmo general que pueda ser implementado sin requerir modificaciones o rediseño. Con esta investigación se pretende diseñar un modelo de optimización entera que describa el problema de asignación de salones en la FI-UAQ y resolverlo con algoritmos de optimización. Dos veces al año la administración de la FI-UAQ se da a la tarea de asignar salones con base a los horarios programados por los coordinadores. La administración unifica los horarios y revisa que no existan cruces en los horarios de los profesores. Una vez corregidas las propuestas iniciales se asignan salones, generalmente durante las vacaciones, ya que esta actividad toma mucho tiempo y es indispensable tenerlos definidos una semana antes del regreso a clases. Se llegan a perder hasta dos semanas de clases cuando profesores y alumnos necesitan terminar de aclarar detalles de horarios cuando no se cuenta con una asignación de salones factible y de calidad desde el primer día de clases. Se propone que utilizando programación lineal la asignación de salones se realizará en menos tiempo, con menos esfuerzo y se podrán cumplir los requisitos de asignar salones a cursos de acuerdo al tamaño del curso y salón, preferencias de salón y con una asignación homogénea. Durante el periodo 2018-1 se ofertaron 465 cursos equivalentes a 2,234 horas semanales y 288 profesores. Dependiendo de la materia los cursos pueden ser impartidos en salones de clases, salones con restiradores, laboratorios o centros de cómputo. El modelo de asignación que se proponga debe considerar especificaciones de asignación, relaciones de tamaño entre cursos y salones, y utilizar un formato práctico que evite retrabajos (capturar información curso por curso). Con el proceso actual la asignación de salones toma aproximadamente diez días y debido a cambios de último momento los semestres inician con horas sin salón asignado. Durante el periodo 2018-1 varios cursos que sumaron 82 horas no tuvieron salón asignado al inicio del semestre de las cuales 74 % corresponden a cursos impartidos en el centro de cómputo, como se muestra en la Figura 1.2a. Es necesario incluir en el modelo la opción de especificar en qué tipo de salón se requiere asignar el curso, y así poder considerar las diferentes características de los salones del centro de cómputo, por ejemplo los tipos de softwares que están instalados en los equipos y el software requerido por el curso.



(a) Horas sin salón asignado.



(b) Ocupación de salones.



(c) Distribución de horarios

Figura 1.2: Análisis de la asignación de salones manual del periodo 2018-1.

No existe una distribución uniforme en las horas de ocupación de los salones de clases; hay salones que están ocupados más del 50 % del tiempo y otros que no se utilizan ni un 25 % del tiempo como se muestra en la Figura 1.2b. Esto puede deberse a preferencias por ciertos salones o a la programación de horarios que no distribuye los horarios durante los 30 periodos del día. Las preferencias de salones se pueden incluir en el modelo de manera similar a las especificaciones de tipo de salón (centro de cómputo, restiradores, laboratorio, etc.). Si se debe al diseño de los horarios el modelo indicará si existe alguna solución factible, en caso de que no, será necesario analizar los periodos en discordia y retroalimentar a los coordinadores involucrados para hacer modificaciones en los horarios. La distribución de horarios determinan el número mínimo de salones requeridos ya que si la mayoría de los cursos se imparten durante la mañana, como sucedió en el semestre 2018-1 (Figura 1.2c), se requerirá mayor número de salones que si se distribuyeran equivalentemente los

cursos durante el día. El diseño de horarios no será parte del alcance de esta investigación.

La FI-UAQ cuenta con seis tipos de salones en los que se imparten los cursos:

1. Salones del Centro de Cómputo
2. Salones con restiradores
3. Laboratorios
4. Salones de tamaño > 25
5. Salones de tamaño ≤ 20
6. Salones de tamaño ≤ 25

De acuerdo al curso se asigna uno de los seis tipos de salones y se considera como una asignación *indeseada* cuando se asigna un tipo de salón distinto al requerido. Con el método manual llegaba un punto en el proceso de asignación, cuando la mayoría de los cursos tenían salón asignado, en el que los intervalos de tiempo que quedaban sin asignar eran menores a la duración de los cursos. Debido a esto se realizaban asignaciones *indeseadas* como asignar un curso que se imparte en salones del tipo 4,5 y 6 en salones del tipo 2. Para confirmar que la infraestructura de la FI-UAQ era suficiente para los cursos ofertados se realizó un análisis entre la oferta y demanda de horas-salón. La Figura 1.3a muestra el total de horas requeridas para los seis tipos de salones de acuerdo a los cursos ofertados durante el periodo 2018-1 (demanda). La Figura 1.3b muestra la cantidad de horas-salón disponibles de acuerdo al número de salones con los que cuenta la FI-UAQ y considerando que están disponibles de 7:00-22:00 de Lunes a Viernes y de 7:00-17:00 los sábados (oferta). La Figura 1.3c muestra que la diferencia entre oferta y demanda más pequeña es la del tipo 4; dado que la oferta es mayor que la demanda no debería ser necesario realizar asignaciones *indeseadas* para ningún curso.

El Cuadro 1.3 muestra el detalle de horas-salón para la oferta y la demanda de los 27 salones del tipo 4 que actualmente están disponibles en la FI-UAQ. Las 1220.5 horas de la última columna son las horas "libres" que restarían una vez realizada una asignación *deseable*. Estos datos demuestran que es posible realizar una asignación de salones deseable para los seis tipos de salones disponibles actualmente en la FI-UAQ.

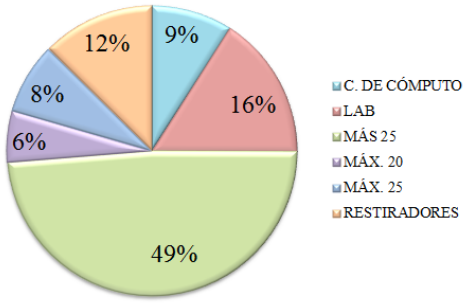
	Oferta (hrs)	Demanda (hrs)	Diferencia (hrs)
Más 25	2295	1074.5	**1220.5**

Cuadro 1.3: Diferencia entre la oferta y demanda de las horas-salón para los 27 salones del tipo 4.

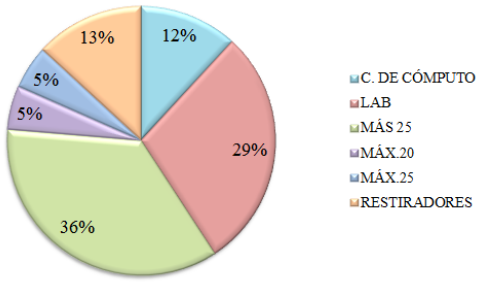
1.2. Planteamiento del problema

Los cursos de la FI-UAQ se pueden impartir en seis diferentes tipos de salones enlistados en la Sección 4.3. La asignación de salones se realiza después de que los coordinadores hayan realizado la asignación de horarios que consta de relacionar los cursos que se ofertarán, con profesores y horarios. El área administrativa recopila los horarios de todos los cursos de la FI-UAQ y revisa que estén libres de errores, como cruces en los horarios de los profesores. El problema de asignación resuelto en esta investigación no verifica los cruces de profesores o de cursos; este trabajo se enfoca en la asignación de cursos que se imparten en salones de clase a salones del tipo 4, 5 y 6.

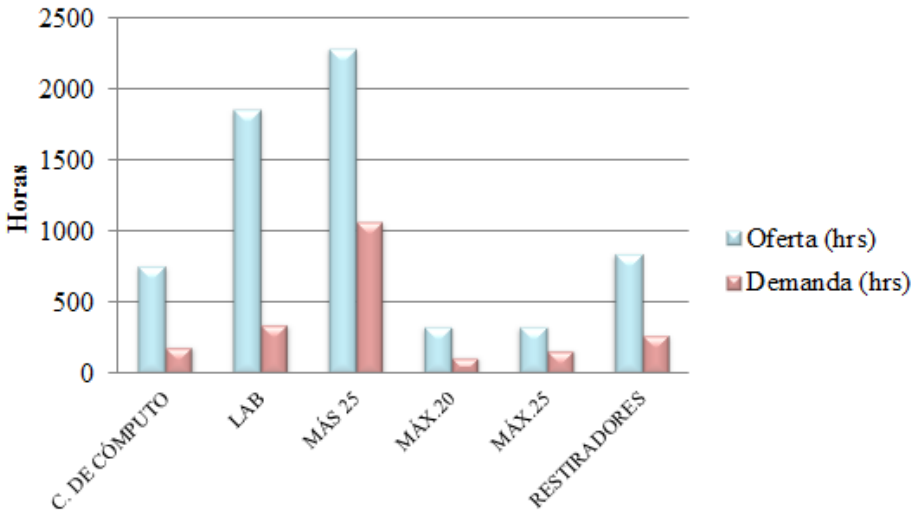
Es importante mencionar que previo a la asignación, se sabe que los programas de estudio de la FI-UAQ pueden compartir los salones de clases, por lo tanto se considerará que al inicio de la asignación los salones están 100 % disponibles. Al relacionar cursos con salones se buscará que todos los alumnos tengan asiento considerando el *tamaño del curso* (número de alumnos inscritos) y el *tamaño del salón* (número de lugares); esto se logra minimizando la relación entre tamaño de cursos y tamaño de salones. También se buscará cumplir con la petición de la administración de priorizar la asignación de ciertos cursos en ciertos salones, a estas peticiones les llamaremos *preferencias de asignación* y se describen a detalle en la Sección 3.4. La



(a) Demanda de horas-salón de acuerdo a los cursos ofertados.



(b) Oferta de las horas-salón de acuerdo a la infraestructura.



(c) Comparación entre la demanda e infraestructura de salones.

Figura 1.3: Análisis de la demanda e infraestructura del periodo 2018-1.

asignación propuesta por el programa se se imprimirán en un formato separado por comas (.csv) para el fácil manejo del usuario (área administrativa). También se imprimirán dos formatos gráficos de la asignación de salones a cursos: uno para los seis días de la semana y otro para cada uno de los 35 salones de la FI-UAQ.

Se pretende comprobar que se disminuye un 90 % el tiempo que toma realizar la asignación de salones en la FI-UAQ respetando las restricciones de horarios, salones y cupos utilizando programación entera. Se diseñará un modelo de programación lineal entera que cumpla lo siguiente:

1. Un curso se imparte necesariamente en un salón.
2. A lo más un curso es permitido por periodo en cada salón.
3. Si un curso tiene más de una sesión por semana, asignar el mismo salón para todas las sesiones. A este tipo de asignación le llamaremos *asignación homogénea*
4. El *tamaño del curso* debe ser menor o igual al *tamaño del salón*.
5. Asignar salones a cursos de acuerdo a las *asignaciones preferidas*.

El tercer punto se utiliza para identificar dos tipos de problemas de asignación de salones el *problema con intervalos* cuando los cursos se imparten una vez al día durante un intervalo de tiempo contínuo. Si los

cursos tienen más de una sesión a la semana y se requiere que se reúnan en el mismo salón toda la semana se les llama *problema sin-intervalos* o *multidía* [Carter & Tovey, 1992]. Carter y Tovey [Carter & Tovey, 1992] demostraron que el *problema de asignación de salones multidía* es del tipo NP-completo. Con el proceso manual se hacía lo mejor posible para realizar una asignación homogénea, en la Figura 1.4 se observa el porcentaje de asignación homogenea obtenida con el proceso manual para el periodo 2018-1 para tres de los campus más importantes de la UAQ. El porcentaje más alto alcanzado con el método manual para la asignación homogénea fue del 98 % de los cursos para el Caso 2. Manualmente se puede realizar una asignación homogenea pero es muy complicado, en los tres casos analizados, uno para cada campus de la UAQ, no se cumplió una asignación homogénea al 100 % en ninguno de los tres con el método manual.

Los requerimientos 4 y 6 se incorporaron en las funciones objetivo y el resto de los puntos son restricciones.

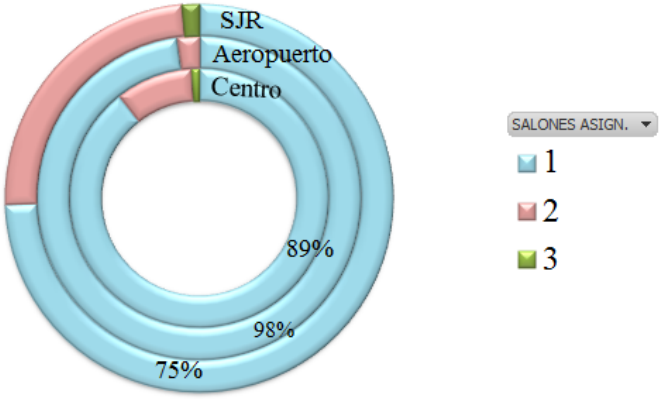


Figura 1.4: Número de salones asignados para las sesiones de los cursos (homogeneidad) en tres de los campus principales de la UAQ.

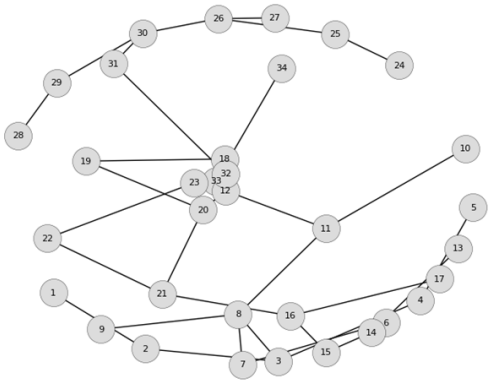
El problema de la FI-UAQ se conforma de los siguientes conjuntos:

- Un conjunto de m cursos (350-450 para este estudio).
- Un conjunto de n salones (35).
- Un conjunto de b banderas o características del salón (variable).
- Periodos p de 30 minutos (30 periodos).
- Días d a la semana (6).

De tal manera que las variables de decisión tendrán cinco subíndices $x_{b,m,n,p,d}$ para incluir los cinco tipos de conjuntos de que definen una variable.

Los m cursos corresponden al total de materias impartidas de los siete programas de estudio ofrecidos en la FI-UAQ (Ingeniería en Automatización, Ingeniería Civil, Licenciatura en Arquitectura, Licenciatura en Animación Digital, Licenciatura en Diseño Industrial, Ingeniería Industrial y Manufactura, y Licenciatura en Matemáticas Aplicadas).

Actualmente la facultad de Ingeniería cuenta con 35 salones del tipo 4, 5 y 6 distribuidos en 5 edificios principales (B, C, E, I y CEDIT). En la Figura 1.5a se muestran las distancias entre salones de acuerdo a las coordenadas obtenidas de una imagen satelital del campus (Figura 1.5b).



(a) Gráfica de las distancias entre salones, los nodos representan los salones y el valor de los arcos las distancias en metros.

(b) Imagen satelital de los cinco edificios de la FI-UAQ campus centro.

Figura 1.5: Infraestructura de salones de clases de la FI-UAQ (2019-1).

La cantidad de banderas b es variable y dependerán de las especificaciones del usuario. Pueden utilizarse para determinar el conjunto de cursos que se desea asignar en un conjunto preferido de salones. Por ejemplo, asignar todos los cursos de Matemáticas en 5 salones del edificio E. Las banderas se utilizan para cumplir con el requisito de *preferencias de asignación*.

En la FI-UAQ los cursos se imparten de 7:00-22:00 hrs de lunes a viernes, y de 7:00-17:00 hrs los sábados. Los cursos pueden tener duraciones variables, de mínimo 1 hora y máximo 5 horas, por esta razón se dividieron los días en periodos de 30 minutos para tener compatibilidad con cualquier duración de curso. En la Figura 1.6 se muestra una representación de la división de un día entre semana en periodos y la manera de calcular la duración de un curso de acuerdo a su periodo inicial p_i y periodo final p_f.

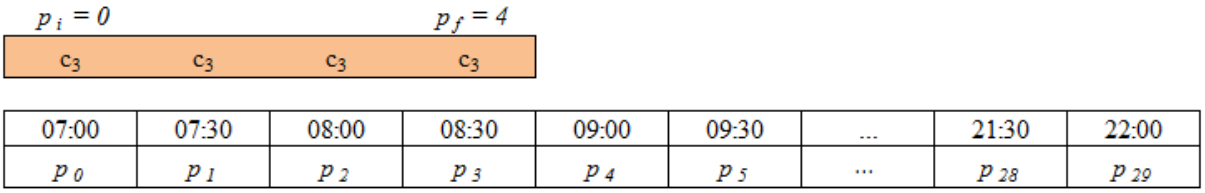


Figura 1.6: El curso c_3 tiene una duración de 4 periodos; $p_f - p_i + 1 = 4$.

1.3. Objetivos

Implementar un modelo de optimización para la asignación de salones en la FI-UAQ que asigne salones de acuerdo a las relaciones de tamaño de curso y tamaño de salón, además de respetar las preferencias de asignación y disminuir el tiempo que toma esta actividad.

1.3.1. Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

- Determinar las restricciones duras y suaves.
- Definir la función objetivo con base a las restricciones.
- Implementar una estrategia de asignación de salones utilizando un modelo de optimización.
- Comparar el tiempo que toma obtener una propuesta de asignación de salones con el modelo de optimización contra el método manual.

En este trabajo de investigación se resuelve el problema de asignación de salones de la FI-UAQ con programación lineal entera. Se diseñó un modelo matemático de acuerdo a las restricciones duras y suaves de la universidad y se resolvió con Gurobi. Se analizaron combinaciones de argumentos que mejor resuelven el problema analizado.

CAPÍTULO 2

Marco teórico

El área de estudio al que pertenece el problema abordado es Investigación de Operaciones, la cual se reconoció con este nombre a inicios de la Segunda Guerra Mundial cuando a un grupo de científicos se les solicitó que hicieran investigación en operaciones (militares). Se requería desarrollar con urgencia una manera científica de abordar problemas como repartir escasos recursos a diversas operaciones militares y a las actividades de estas operaciones de una manera efectiva. Los logros obtenidos en la guerra al aplicar métodos de la Investigación de Operaciones (IO) ocasionaron que surgiera el interés por utilizarlos en la industria post-guerra. Durante esta época se desarrollaron avances importantes, como ejemplo principal el método Simplex desarrollado por George Dantzing en 1947 para resolver problemas de programación lineal. Fue tal el auge de la IO que gran parte de las herramientas estándar como la programación lineal, programación dinámica, teoría de colas, y la teoría del inventario se desarrollaron antes de que terminara la década de los 50's. Otro factor que ocasionó avances importantes fue el desarrollo de la computadora electrónica digital por su habilidad para realizar cálculos aritméticos. Actualmente millones tienen acceso a softwares de IO utilizados para resolver problemas de todos tipos y tamaños. La investigación de operaciones ha tenido un impacto impresionante en mejorar la eficiencia de numerosas organizaciones al rededor del mundo. En el proceso, IO a contribuido de manera significativa en la creciente productividad de la economía de varios países. Europa y Asia tienen federaciones de sociedades IO que coordinan la realización de conferencias internacionales y publicaciones en revistas internacionales. El Instituto de Investigación de Operaciones y las Ciencias de Gestión (INFORMS por sus siglas en inglés) es una sociedad de IO internacional. *Interfaces* es una de sus numerosas revistas, la cual se enfoca en publicar artículos de investigaciones principales y el impacto que tuvieron en las respectivas organizaciones. En [Hillier & Lieberman, 2010] enlistan un gran número de aplicaciones en las que IO ha ayudado a ahorrar millones de dolares.

La IO busca eliminar conflictos de interés entre las áreas de una organización al proponer resultados convenientes para la mayoría. No quiere decir que se tenga que considerar explícitamente todos los aspectos de la organización, sino que los objetivos planteados deben ser consistentes con los objetivos globales de la organización. La IO frecuentemente se enfoca en encontrar la *mejor* solución (conocida como solución *óptima*)para el modelo que representa el problema en consideración. Nos referimos a *una* mejor en lugar de *la mejor* solución porque puede haber múltiples soluciones con el mismo valor. La "búsqueda por la optimalidad" se interpreta en términos de las necesidades practicas para identificar un posible curso de acción. Es por este enfoque práctico que el equipo IO está compuesto por gente con experiencia en la actividad que se busca mejorar, conocimientos de la organización en general y otros expertos en modelado y métodos de resolución.

Una vez que se conoce a detalle el problema y se han definido los objetivos es posible comenzar con el diseño del modelo. Un modelo matemático representa, de manera idealizada, la esencia del problema por medio de símbolos y expresiones matemáticas. Son ideales para extraer la esencia del asunto que se está consultando, mostrar interrelaciones, y para facilitar el análisis. Por lo tanto, si se tienen n decisiones cuantificables se expresan como **variables de decisión** (por ejemplo, $x_1, x_2, ..., x_n$) cuyos valores se determinarán

al resolver el modelo. La medida de rendimiento que se busca maximizar o minimizar se expresa como una función matemática de las variables de decisión (por ejemplo, $P = 3x_1 + 8x_2 + ... + 4x_n$). A esta función se le conoce como **función objetivo**. Cualquier restricción de los valores que se le pueden asignar a las variables de decisión también se expresan matemáticamente, generalmente como desigualdades (por ejemplo, $x_1 + x_2 + ... + x_n \leq 1$). A estas desigualdades se les conoce como **restricciones**. Por último, a las constantes (coeficientes y lado derecho de las desigualdades) en las restricciones y función objetivo son los **parámetros** del modelo. "Una *solución óptima* es una *solución factible* (solución para la que todas las restricciones se satisfacen) que tiene el *valor más favorable*de la función objetivo" [Hillier & Lieberman, 2010]. Si se busca maximizar la función objetivo el valor más favorable es el valor más grande, y por consecuencia si se busca minimizar este será el valor más pequeño. Muchos problemas tendrán una única solución óptima, pero es posible que tengan más de una. Estos casos suceden cuando la recta de la función objetivo intersecta con el área de soluciones factible, ocasionando que todos los puntos del segmento de línea sean óptimos. Cualquier problema con múltiples soluciones óptimas tendrá un número infinito de estas, cada una con el mismo valor óptimo de la función objetivo.

La resolución de dichos problemas comienza por su formulación a partir de observaciones cuidadosas y recopilación de datos relevantes. Después se construye un modelo científico (comunmente matemático) que plasma la esencia del problema real, permite comprender la estructura del problema y ayuda a revelar relaciones importantes de causa-y-efecto. Desarrollar el modelo es esencial para la resolución del problema, ya que es el puente para utilizar las técnicas matemáticas y las computadoras para analizar y solucionar el problema. La precisión debe ser suficiente para que las soluciones obtenidas sean válidas para el problema real. El modelo se considera "validado" una vez que se realizaron varios experimentos y se obtuvieron resultados factibles en repetidas ocasiones.

Un tipo de modelo particularmente importante son los modelos de programación lineal, llamados así porque su función objetivo y restricciones son funciones lineales. La programación lineal ha causado un impacto extraordinario desde 1950 y actualmente es una herramienta estándar utilizada por numerosos países industrializados con la que ahorran miles y millones de dolares en industrias, y en sectores públicos y privados. El beneficio de poder definir un problema como modelo lineal es que se podrá resolver con métodos de solución eficientes, cómo el método símplex. Los modelos de programación lineal para problemas reales pueden llegar a tener millones de restricciones y variables de decisión. Debido a esto sería poco viable intentar escribir la formulación algebráica o rellenar los parámetros de una hoja de cálculo. Es por esto que para los grandes modelos se utiliza un *lenguaje de modelado*. Un lenguaje de modelado matemático es un software que ha sido diseñado específicamente para formular grandes modelos matemáticos eficientemente. Estos lenguajes matemáticos formulan todas las restricciones a partir una sola expresión y un bloque de datos de la base de datos. Algunos de los softwares desarrollados en las últimas décadas son AMPL, MPL, OPL, GAMS y LINGO.

LINGO es un popular lenguaje para modelar algebráicamente y LINDO es un optimizador tradicional. Excel cuenta con plantillas para solucionar modelos básicos. La versión para estudiantes del software para modelar optimización, MPL (Lenguaje de Programación Matemática por sus siglas en inglés). Es un sistema de modelado que se mantiene al día con el estado del arte y que permite formular modelos complicados de optimización de manera clara, concisa y eficiente. Su solucionador principal CPLEX de IBM es el optimizador más utilizado para programación lineal, programación entera mixta y programación cuadrática. Su interfaz y la biblioteca pueden ser llamadas desde cualquier lenguaje como C, C++, Python, etc. Implementa el método símplex, el método de Barrera y programación entera mixta. El motor de programación de restricciones es el CPLEX CP Optimizer elaborada con principios de inteligencia artificial. Algunos de los MPL que tienen versiones para estudiantes son CONOPT (para programación convexa), LGO (para optimización global), LINDO (para programación matemática), CoinMP (para programación lineal y entera), y BendX (para algunos modelos estocásticos).

Estos softwares incluyen técnicas que se aproximan a la solución óptima, pues a diferencia de la programación lineal los problemas de programación entera no cuentan con una técnica de resolución exacta. La programación lineal entera, a diferencia de la programación lineal que tiene soluciones continuas, tiene soluciones únicamente enteras. En caso de permitir una combinación de ambas se conoce como programación

entera mixta. Otra característica que determina la complejidad de los problemas es la cantidad de objetivos planteados; los problemas pueden ser mono-objetivo o multi-objetivo y de acuerdo al tipo de problema la técnica de solución. El problema de asignación analizado en esta investigación es de **programación lineal entera multi-objetivo con variables binarias**.

Los procedimientos para resolver problemas de programación lineal entera se pueden clasificar a groso modo en tres categorías:

1. Técnicas de enumeración, que incluye el procedimiento de ramificación-y-acotamiento;
2. Técnicas de planos de corte;

Con la práctica se van identificando los tipos de problemas para los cuales se puede utilizar cada uno de estos procedimientos, pues su desempeño depende del problema a resolver. Adicional a estas tres técnicas se han desarrollado varios procedimientos que combinan técnicas.

La técnica de ramificación y acotamiento se basa en dividir la región factible en secciones más pequeñas para que el problema sea manejable. Hay diferentes maneras de dividir la región factible dependiendo del algoritmo de ramificación-y-acotamiento que se elija. Como se explica detalladamente en [Hax, Bradley, 1977], el algoritmo básico de ramificación-y-acotamiento subdivide la región factible para establecer fronteras $\underline{z} \leq z^* \leq \bar{z}$. Esto quiere decir que se calcularán varias soluciones enteras en las diferentes subdivisiones de la región factible; algunas menores y otras mayores al óptimo. El objetivo es encontrar la solución entera ($\underline{z} \leq z^*$, al maximizar) que más se aproxime al óptimo.

Una manera de solucionar problemas de programación entera con variables binarias es el procedimiento especial de ramificación y acotamiento de enumeración completa. Se enlistan todas las posibles combinaciones de variables binarias y se selecciona el mejor punto que sea factible. Esta propuesta funciona muy bien para problemas pequeños donde haya pocas combinaciones potenciales de $0-1$. En general, un problema con n variables contiene 2^n combinaciones $0-1$; por lo tanto para valores grandes de n el método de enumeración es inviable. En su lugar se consideran implícitamente las combinaciones de variables binarias ignorando las desigualdades lineales y manteniendo siempre las restricciones de $0-1$. La idea es utilizar un proceso de subdivisiones fijando algunas variables a 0 o 1. El resto de las variables que no se fijan se llaman *variables libres*. Dependiendo de los coeficientes de las variables en la función objetivo se establecen los valores 0 o 1 para buscar maximizar o minimizar. Si se busca maximizar, entonces se les asigna 0 a las variables libres con coeficientes negativos. El proceso inicia *sin* variables fijas, por lo tanto todas las variables son *libres* y fijadas a cero. Si la solución no satisface las desigualdades de las restricciones, se divide el problema fijando una variable a 1 en búsqueda de soluciones factibles. Se continúa subdividiendo únicamente las secciones donde se encuentran mejores soluciones que en la ramificación anterior.

El algoritmo de planos de corte resuelve problemas de programación entera modificando soluciones de programación lineal hasta que se obtiene una solución entera. A diferencia del método de ramificación y acotamiento, no divide la región factible sino que utiliza un único programa lineal afinándolo al agregar nuevas restricciones. Estas restricciones van reduciendo sucesivamente la región factible hasta encontrar una solución entera óptima. La técnica de planos de corte fue el primer algoritmo que se probó convergía a un número finito de pasos para obtener una solución en programación entera. Generalmente no es muy eficiente en comparación con otros procedimientos como el de ramificación-y-acotamiento, pero ha sido importante en la evolución de otros algoritmos, más eficientes, para la programación entera [Hax, Bradley, 1977].

Definición 1. *Un problema de decisión D está en NP si se puede verificar en tiempo polinomial si una solucion dada es solución de una instancia de D.*

Definición 2. *Un problema de decisión D es NP-Duro si cualquier instancia de cualquier problema NP-Duro se puede convertir en una instancia de D en tiempo polinomial.*

Definición 3. *Un problema de decisión D es NP-Completo si se cumplen las siguientes dos condiciones*

- *El problema D está en NP.*
- *El problema D es NP-Duro.*

El problema que se trata en este trabajo es NP-Completo porque se puede ver como una generalización del MAP el cual se sabe que es NP-Completo aún cuando solo sean tres dimensiones [Karp, 1972], por lo tanto el problema de asignación de salones es al menos tan complicado como el MAP.

2.1. Antecedentes

Los problemas de programación de horarios de cursos en las Universidades son particulares porque cada institución tiene restricciones muy específicas. Debido a esto los modelos diseñados no solucionan todos los casos de estudio, es necesario ajustarlos agregando o eliminando restricciones y funciones objetivo para describir el problema analizado. Otra variación que puede tener el problema de horarios es la manera de dividirlo, en ocasiones para disminuir el tamaño del problema se resuelve únicamente la asignación de salones utilizando un horario pre-asignado. A continuación se resumen algunas de las investigaciones del problema de horarios y asignación de salones en universidades en las que se han implementado y/o desarrollado métodos exactos, heurísticas y/o metaheurísticas.

Erben y Keppler [Erben & Keppler, 1995] utilizaron un algoritmo genético para programar los horarios y salones en su universidad. Optaron por no separar la asignación de salones de la programación de horarios debido a que el objetivo principar era utilizar eficientemente los salones y era más posible obtener una buena solución abordando los problemas simultáneamente.

El documento de Abell [Abell, 1965] fue de los primeros registros del uso de sistemas computacionales con el propósito de diseñar horarios de clases. Utilizó el lenguaje COBOL (Common Business Oriented Language) y una computadora IBM 7090 como la que se muestra en la Figura 2.1 para satisfacer los requerimientos de horarios de los estudiantes y distribuir equitativamente la carga horario.

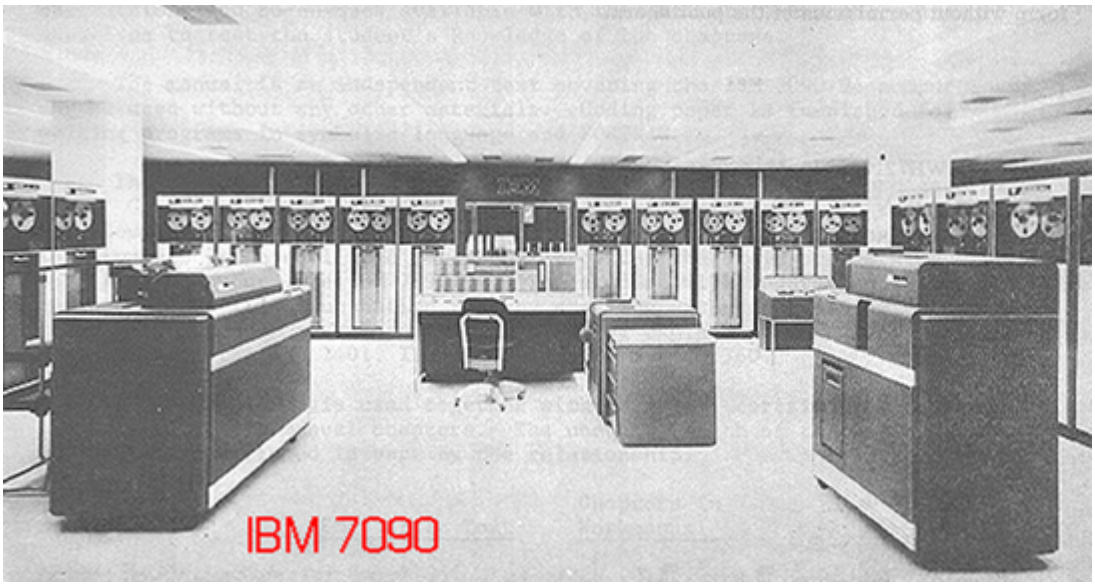


Figura 2.1: Foto de supercomputadora IBM 7090 [Harper, 2009]

Laporte y Desroches [Laporte & Desroches, 1986] diseñaron e implementaron un algoritmo para optimizar los horarios de 2799 estudiantes en una universidad de Canadá con dos campus ubicados a 3 kilómetros de distancia. Utilizaron técnicas computarizadas para proponer horarios y asignación de salones. Los horarios propuestos cumplían el 96.4 % de las restricciones determinadas por los estudiantes. Estas restricciones se generaban con base a los horarios elegidos por los estudiantes, esto genera un problema mayor, ya que los estudiantes con cursos retrasados solicitaban cursos del semestre anterior y a su vez, elegían otros, lo que

generaba un traslape en los horarios, afectando la modelación del programa por el aumento de variables, lo que generaba mayor error a la infactibilidad del problema. A pesar de estas dificultades su algoritmo realizaba 6 iteraciones en 13.05 minutos proponiendo horarios con pequeñas violaciones como 5 salones con un número excedente de alumnos, para estas corridas, se utilizó la computadora de la universidad (CYBER 173).

A finales de la década de los 90's comenzó una tendencia por desarrollar algoritmos para búsqueda de soluciones generales, llamados metaherísticos (recocido simulado, algoritmo genético, búsqueda tabú, etc.) [Burke et al., 1997]. Anteriormente se utilizaron métodos heurísticos simples que resolvían problemas específicos, pero se volvieron obsoletos cuando crecieron los problemas de horarios en las universidades modernas volviéndose más complejos. Se puede consultar la evolución de los métodos propuestos en las presentaciones de las conferencias de Teoría y Práctica en Automatización de Horarios (PATAT por sus siglas en ingles); se han llevado a cabo cada dos años desde 1996 [EVENTMAP, 2016].

Martinez-Alfaro y Flores-Teran dividieron el problema para abordar por separado la programación de horarios de cursos y la asignación de salones. En [Martinez-Alfaro & Flores-Teran, 1998] presentan un sistema de optimización basado en recocido simulado para la asignación de salones en una universidad mexicana. Se plantearon disminuir la distancia recorrida entre los departamentos y los salones asignados, así como disminuir los pisos de diferencia entre un salón y otro, y disminuir la diferencia entre el cupo del salón y tamaño de grupo. Simplificaron el problema separando los cursos con los mismos periodos de clase y resolviendo cada sesión periodo por periodo comenzando por el primer periodo del día. El algoritmo se desarrolló en C y se utilizó mSQl para manejar la base de datos de cursos, salones, e instructores. El tiempo de cómputo promedio fue de 14 horas para el segundo semestre de 1996, buscaron mejorarlo utilizando un procesamiento paralelo para analizar los periodos simultáneamente, si se obtuviera un tiempo de cómputo menor, se podría realizar la asignación de salones cuando se termine el proceso de registro obteniendo una mejor asignación dado que los datos serán reales y no estimados.

Constantino *et al.* [Constantino et al., 2010] utilizó tres algoritmos para solucionar el problema de asignación de salones en una universidad. La asignación se realiza en dos fases:

i) En la primera cada departamento realiza los horarios de sus cursos y no consideran la asignación de salones porque estos se pueden compartir entre departamentos.

ii) En la segunda fase se hace la asignación de salones, de manera centralizada, de acuerdo a los horarios generados en la primera.

El trabajo se enfoca en las metodologías implementadas para la asignación de salones (segunda fase) y comparan los resultados. Complementaron un algoritmo heurístico con programación lineal y lo compararon con los resultados obtenidos con un algoritmo determinista. Los mejores resultados los obtuvieron con la combinación del algoritmo determinista (programación lineal) con el estocástico (búsqueda de variables vecinas). Reportan que en un promedio de 40 minutos redujeron un 50 % las distancias recorridas entre salones en comparación con el método manual.

Tassopoulos y Beligiannis [Tassopoulos & Beligiannis, 2012] mejoraron un algoritmo híbrido que propone una programación de horarios y asignación de salones en 14 minutos. Se enfocaron en diseñar un algoritmo ajustable, por lo tanto le pueden incluir o eliminar las restricciones sin afectarlo. Probaron el algoritmo con datos públicos de instancias y obtuvieron que en promedio superó a las demás técnicas el 66.7 % de los casos, para el estudio utilizaron una computadora de 64-bits y 2.2GHz con Windows 7.

Otro algoritmo híbrido inspirado por la naturaleza y adaptado para resolver el problema de horarios y salones, es el de Skoullins *et al.* [Skoullis et al., 2017], el cual utiliza un algoritmo híbrido basado en optimización de enjambre de gatos (CSO, por sus siglas en inglés). Reportan que superó en rendimiento y en tiempo de cómputo a los algoritmos genéticos, algoritmos evolutivos, recocido simulado, al híbrido propuesto por sus compatriotas Tassopoulos y Beligiannis [Tassopoulos & Beligiannis, 2012], y el enjambre artificial de peces (AFS, por sus siglas en inglés). Como criterio de evaluación consideraron las restricciones de distribución uniforme de horas de profesores, distribución uniforme de sesiones de curso, y las horas libres entre clases que le quedan a cada profesor.

Hayrapetyan [Hayrapetyan, 2017] publicó un proyecto en el que utiliza algoritmo genéticos para programar horarios utilizando Python como lenguaje de programación, en el cual incluye restricciones duras para determinar el horario de cuatro grupos con diferentes cursos. El programa analiza que los horarios no tengan cruces de profesores, cursos y salones. No tiene el alcance que se requiere para la programación de horarios o asignación de salones de una instancia como las resueltas en los artículos citados anteriormente, pero permite comprender el funcionamiento de los algoritmos heurísticos y la importancia de incluir lógica en la captura de datos. A pesar de que existen numerosos artículos sobre asignación de salones y programación de horarios es difícil encontrar ejemplos de códigos con los que se pueda aprender de la lógica y aplicación de estos complejos algoritmos en la práctica.

Cuadro 2.1: Resumen del estado-del-arte.

Autor	Año	Tipo de problema	Método	Tiempo cómputo	Equipo
Abell	1965	Programación de horarios y asignación de salones	Constraint programming	-	IBM-7090
Laporte y Desroches	1986	Programación de horarios y asignación de salones	Ramificación y acotamiento	13.05 minutos	CYBER 173
Carter, Michael W.	1989	Asignación de salones	Relajación Lagrangiana	15 minutos	IBM 4341
Erben y Keppler	1995	Programación de horarios y asignación de salones	Algoritmo genético, implementado con C y Prolog	8.5 horas	-
Martines-Alfaro y Flores-Terán	1998	Asignación de salones	Recocido simulado programado en C	14 horas	IBM-6000
Abdennadher \textit{et al.}	2000	Asignación de salones	Programación lógica con restricciones y un programa CHR	-	-
Constantino \textit{et al.}	2010	Asignación de salones	Método Húngaro, heurística y metaheurística	35 minutos	PC AMD Athlon 2.4MHz 1GB RAM
Tassopoulos y Beligiannis	2012	Programación de horarios y asignación de salones	Algoritmo híbrido (enjambre de partículas)	14 minutos	64-bits 2.2 GHz
Phillips \textit{et al.}	2015	Asignación de salones	Programación entera con Gurobi 5.1	5.5 segundos	64-bit Debian 7 3.33GHz i5-2500k
Skoullis \textit{et al.}	2017	Programación de horarios y asignación de salones	Algoritmo híbrido (enjambre de gatos), programado en ANSI C	<14 minutos	Pro 64-bit, i7-2600, 3.4 GHz, 8GB RAM
Hayrapetyan	2017	Programación de horarios y asignación de salones	Algoritmo genético, código en Python	<5 minutos	-

CAPÍTULO 3

Metodología

El problemas de asignación de salones busca relacionar el salón s con el curso c en un periodo p del día d. En este trabajo se propuso una asignación de salones utilizando programacion lineal entera y formulándolo como un problema de asignación de 3-dimensiones [Pérez, 2017]. Dado un conjunto de n_1 cursos, n_2 salones y n_3 días y un costo c_{ijk} cuando se asigna el curso c_i al salón s_j en el día d_k con $1 \leq i \leq n_1, 1 \leq j \leq n_2, 1 \leq k \leq n_3$, la formulación binaria del programa lineal entero es:

$$\min \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} c_{ijk} \cdot x_{ijk} \tag{3.1}$$

Sujeto a:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} = 1, \ \forall \ k \tag{3.2}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{n} x_{ijk} = 1, \ \forall \ j \tag{3.3}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{n} x_{ijk} = 1, \ \forall \ i \tag{3.4}$$

donde la variable de decisión $x_{ijk} \in \{0, 1\}, \quad \forall\, i, j, k$ es igual a 1 cuando al curso c se asigna al salón s en el día d, y 0 en otro caso.

En el problema de asignación de salones se busca minimizar el costo c_{ijk} y el problema tiene solución si existe al menos una combinación para la que todos los cursos se asignan a un salón cumpliendo con las restricciones que se especifiquen [Hillier & Lieberman, 2010].

3.1. Especificaciones

El problema de asignación de salones en la FI-UAQ se puede definir como un problema de optimización multidía con preferencias monotónicas de acuerdo a la definición de Carter *et al* [Carter & Tovey, 1992]. Se diseñó un modelo de programación lineal (PL) con una lista de restricciones que determinan especificaciones básicas como asignar un sólo salón por curso y un solo curso por salón. Las funciones objetivo a minimizar son la relación tamaño curso/tamaño salón y la diferencia entre el conjunto de salones preferidos y el salón asignado. En este capítulo se presenta el modelo matemático diseñado para representar el problema de asignación de salones de la FI-UAQ.

El área administrativa de la facultad se encarga de recopilar los horarios en un único archivo de Excel, analizan si existen empalmes en los horarios de los profesores e incongruencias de transporte (cursos en diferentes campus con poco tiempo de separación) y le informan a los respectivos coordinadores para que ajusten los horarios. Una vez que los horarios están libres de errores comienza la etapa de asignación de salones analizada en este trabajo. El proceso general se muestra en la Figura 3.1.

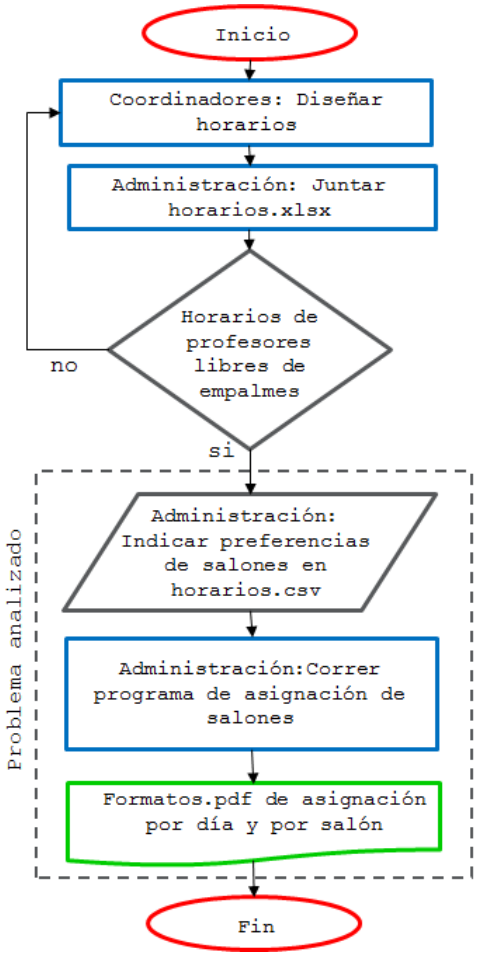


Figura 3.1: Proceso administrativo de recopilación de horarios y asignación de salones.

La principal solicitud del área administrativa para la asignación automática de salones fue respetar el formato de horarios utilizado en el momento (hoja de cálculo de Microsoft Excel), ya que han recibido propuestas para gestionar horarios pero dependen de capturar datos uno por uno, ahorrando tiempo en la actividad principal (gestionar) pero invirtiéndolo en otra. Por lo tanto, en la metodología propuesta se consideró evitar actividades tediosas desde el momento de la lectura de datos hasta la solución del problema de asignación de salones. Esta comienza con la conversión de la hoja de cálculo, en la que se recopilaron los horarios de la facultad, a un archivo separado por comas (fácil de convertir utilizando Microsoft Excel) con el que el programa escrito genera automáticamente las variables de decisión, restricciones y funciones objetivo de acuerdo al modelo diseñado respetando los requisitos de la FI-UAQ, .

Las siguientes definiciones se consideraron para formular el modelo:

- Los cursos tienen duraciones variables entre 1 hr a 5 hrs y pueden ser programados de las 7:00-22:00

de lunes a viernes, y de 7:00-17:00 los sábados. Se definieron periodos de 30 minutos con el subíndice p para identificar los horarios de los cursos.

- Los cursos pueden tener una o más sesiones a la semana, por lo que denotaremos una sesión del curso c en el día d como $S(c,d)$.
- Nos referiremos al conjunto de periodos p contiguos que conforman una sesión $S(c,d)$ como $D(S(c,d))$ con una duración de $p_{f_c} - p_{i_c} + 1$.
- Los cursos tienen una clave única para identificarlo pero en ocasiones más de una clase del mismo curso se abre por semestre debido a la demanda, por lo tanto los cursos se identificaron por la relación: curso (c), periodo (p), día (d).
- La asignación se realizó a nivel de curso y no a nivel de grupo porque los estudiantes pueden elegir cursos de diferentes grupos según sus requerimientos.
- El tamaño de un curso c es el máximo número de alumnos que pueden estár inscritos y se denota como $|c|$.
- La infraestructura the la FI-UAQ tiene 35 salones s que se dividen en tres tipos 4,5 y 6 de acuerdo a su tamaño como se especificó en la Sección 1.1. El tamaño del salón s es el máximo número de alumnos que pueden estar en el salón y se denota como $|s|$.

Primero se diseñó y programó el modelo mono-objetivo con la función objetivo que minimiza la relación entre el tamaño del curso y el cupo del salón, de esta manera el programa asigna el salón s más grande disponible para el curso c en el periodo p el día d. Se consideraron las siguientes restricciones duras:

- Asignar un salón para cada periodo que compone la sesión de un curso.
- Asignar el mismo salón para todas las sesiones que tiene el curso a la semana.
- Asignar a lo más un curso c por salón s en el periodo p del día d.

3.2. Modelo mono-objetivo

Como variables de decisión se definieron:

$$x_{c,s}^{p,d} = \begin{cases} 1 & \text{Si al curso } c \text{ se le asigna el salón } s \\ & \text{en el periodo } p \text{ del día } d \\ 0 & \text{En otro caso.} \end{cases} \tag{3.5}$$

$$y_{c,s}^{d} = \begin{cases} 1 & \text{Si al curso } c \text{ se le asigna el salón } s \\ & \text{en el día } d \\ 0 & \text{En otro caso.} \end{cases} \tag{3.6}$$

El costo w_{cs}^{d} de asignar el curso c al salón s el día d en la función objetivo f_1 dependerá de la relación del tamaño del curso $|c|$ y del tamaño del salón $|s|$.

$$\min\ f_1 = \sum_{c=1}^{C_d} \sum_{s=1}^{S} \sum_{d=1}^{6} w_{cs}^{d} \cdot y_{cs}^{d} \tag{3.7}$$

Sujeto a:

$$\sum_{s=1}^{S} x_{cs}^{pd} = 1, \quad \forall\, c,\ d \in S(c,d),\ p \in D \tag{3.8}$$

$$\sum_{c=1}^{C} x_{cs}^{pd} \leq 1, \quad \forall\, s,\ d \in S(c,d),\ p \in D \tag{3.9}$$

$$\sum_{p \in D(S(c,d))} x_{cs}^{pd} = (p_{f_c} - p_{i_c} + 1) \cdot y_{cs}^{d}, \quad \forall\, c,\ s,\ d \in S(c,d) \tag{3.10}$$

$$\sum_{s=1}^{S} y_{cs}^{d} = 1, \quad \forall\, c,\ d \in S(c,d) \tag{3.11}$$

$$y_{cs}^{d} = y_{cs}^{d'}, \quad \forall\, c, s,\ (d,d') \in 1,...,6 | d < d' \tag{3.12}$$

Donde:

- En la Ecuación 3.7

$$w_{cs}^{d} = \frac{|c|}{|s|} \tag{3.13}$$

- En la Ecuación 3.12 la resta de las variables auxiliares es 0 si se asigna el mismo salón a cada una de las sesiones $S(c,d)$.
- Si un curso tiene más de una sesión $S(c,d)$ a la semana, el mismo salón s se asignará a cada una de ellas. Las restricciones que realizan esta asignación homogénea son las Ecuaciones 3.11,3.10 y la 3.12.

La asignación a la que llamamos "homogénea" se definió por Carter [Carter, 1989] e identificaron como restricción de "eliminación de brincos de salón"; su función es mantener el curso en el mismo salón durante todos los periodos de la sesión, evitando que a mitad de clase se cambie de salón, y también mantiene todas las sesiones de la semana en el mismo salón. A los problemas de asignación de salones que consideran la "eliminación de brincos de salón" Carter y Tovey [Carter & Tovey, 1992] le llamaron *multidía* y demostraron que encontrar una solución factible es un problema NP-completo.

Otra característica definda por Carter y Tovey son las *preferencias monotónicas* [Carter & Tovey, 1992] que indica preferencias de asignación entre conjuntos de cursos y salones. Por ejemplo, especificar que al curso "Diseño asistido por computadora" se le debe asignar un salón del centro de cómputo que cuente con el software adecuado. Las preferencias de asignación también se pueden especificar por conjunto de cursos y salones, por ejemplo señalar que se desea que los cursos de Ingeniería Civíl se asignen en el edificio B.

Cuando se implementó el modelo mono-objetivo para el periodo 2019-1 los cursos programados se asignaron a 31 de los 35 salones con los que cuenta la facultad, pero las preferencias de asignación no eran consideradas por el modelo. Para obtener una asignación similar a la preferida por el área administrativa se realizaron múltiples corridas del programa modificando el archivo de entrada manualmente, lo que tomó 20 hrs. Fue necesario re-evaluar el modelo mono-objetivo para incluir las restricciones suaves que relacionan cursos con salones de acuerdo a preferencias. Para esto incluimos una variable a la que llamamos "bandera" que relaciona conjunto de cursos y salones de acuerdo a las preferencias de asignación, y se definió otro objetivo que minimiza la diferencia entre el salón asignado y el preferido.

3.3. Modelo multiobjetivo

El costo w_{cs}^d de asignar el curso c al salón s el día d en la función objetivo f_2 dependerá de la diferencia entre la bandera preferida $b(c)$ y la bandera asignada $b(s)$.

$$min\ f_2 = \sum_{c=1}^{C_d} \sum_{s=1}^{S} \sum_{d=1}^{6} q_{cs}^d \cdot y_{cs}^d \tag{3.14}$$

Donde, en la Ecuación 3.14:

$$q_{cs}^d = \left| 1 - \frac{b(c)}{b(s)} \right| \tag{3.15}$$

Por lo tanto, el modelo multiobjetivo se conforma de las Ecuaciones 3.5-3.15:

$$\min(f_1, f_2) \tag{3.16}$$

Donde:

$$f_1 = \sum_{c=1}^{C_d} \sum_{s=1}^{S} \sum_{d=1}^{6} w_{cs}^d \cdot y_{cs}^d$$

$$f_2 = \sum_{c=1}^{C_d} \sum_{s=1}^{S} \sum_{d=1}^{6} q_{cs}^d \cdot y_{cs}^d$$

Sujeto a:

$$\sum_{s=1}^{S} x_{cs}^{pd} = 1, \quad \forall\, c,\ d \in S(c,d),\ p \in D$$

$$\sum_{c=1}^{C} x_{cs}^{pd} \leq 1, \quad \forall\, s,\ d \in S(c,d),\ p \in D$$

$$\sum_{p \in D(S(c,d))} x_{cs}^{pd} = (p_{f_c} - p_{i_c} + 1) \cdot y_{cs}^d, \quad \forall\, c,\ s,\ d \in S(c,d)$$

$$\sum_{s=1}^{S} y_{cs}^d = 1, \quad \forall\, c,\ d \in S(c,d)$$

$$y_{cs}^d = y_{cs}^{d'}, \quad \forall\, c, s,\ (d, d') \in 1, ..., 6 | d < d'$$

El modelo multiobjetivo se trató como un modelo mono-objetivo con el método de la suma ponderada:

$$\alpha_1 f_1 + \alpha_2 f_2$$
$$\text{con } \alpha_1 + \alpha_2 = 1$$

Se probaron diez combinaciones de pesos para elegir cuál resolvía mejor la instancia analizada. El detalle de los resultados se presenta en la Sección 4.2.1.

3.4. Implementación

El primer programa del modelo, escrito en Python utilizando la biblioteca de Gurobi, no incluía funciones objetivo. El programa asignaba salones leyendo el archivo de horarios en formato separado por comas y se indicaba una variable de entrada numérica para especificar la cantidad de salones disponibles. Esta primer versión no consideraba tamaños de grupos ni salones, por lo tanto la distinción de tamaños se hizo manualmente separando el conjunto de cursos y salones con tamaños similares y se corría el programa para asignar la relación de salones-cursos. Este proceso semi-automático tomó aproximadamente 10 horas.

La segunda versión del programa incluía la función objetivo que minimiza la relación entre tamaño de curso y salón. El proceso fue mucho más rápido, pues consideraba automáticamente la preocupación principal de asignar cursos a salones asegurando que tuvieran suficientes lugares para todos los alumnos inscritos. El programa leía dos archivos separados por comas: el primero con los horarios de cada curso y una columna que especificaba el tamaño del curso, el segundo era un listado de los salones disponibles y una columna con el tamaño del salón. De esta manera el programa revisaba todas las posibles combinaciones y buscaba minimizar la función objetivo 3.7. Esta versión del programa disminuía el tiempo de asignación un 99 % en comparación con el método manual.

La tercer versión, y última, del programa incluyó la función objetivo que considera las preferencias de asignación. En la FI-UAQ una asignación informal de edificios a programas se ha establecido por años, por ejemplo asignar principalmente cursos de Ingeniería Civil e Ingeniería en Automatización a los edificios B y C. Además de que facilita a profesores y alumnos recordar la zona donde se imparte la clase, ayuda a disminuir el flujo de estudiantes en los pasillos.

Al inicio de la investigación se consideró incluir como función objetivo minimizar el cambio de salones por grupo, o lo que es lo mismo, disminuir las distancias recorridas por grupo como en [Constantino et al., 2010]. Eventualmente este objetivo se descartó porque los alumnos eligen cursos de diferentes grupos de acuerdo a sus necesidades por lo tanto, minimizar el cambio de salones por grupo no aseguraría una disminución del flujo en los pasillos y solamente haría más complejo el problema sin ningún beneficio extra. En cambio se optó por incluir la función objetivo que minimiza la diferencia entre la asignación preferida y la asignación establecida. Por ejemplo, si se busca asignar el tronco común en la menor cantidad de salones del edificio B, se identifican estos cursos y los tres salones con un único número de bandera. La bandera establecida se diferencía únicamente por un dígito del resto de los cursos de esas licenciaturas para que en caso de que haya periodos libres en los tres salones después de la asignación otros cursos de esas licenciaturas puedan ser asignados en ellos por un bajo costo. El resto de las banderas se estableció con una lógica similar donde los cursos que pertenecen al mismo programa de estudios se asignan a un único edificio, como se muestra en el Cuadro 3.1 y en el Cuadro 3.2. Las banderas 4, 6, y 7 son excepciones debido a solicitues especiales de profesores y coordinadores.

	Bandera							
Edificio	1	2	3	4	5	6	7	Total
A							2	2
B	4	3						7
C	6							6
E			11					11
I				1	7	1		9
Total	10	3	11	1	7	1	2	35

Cuadro 3.1: Banderas especificadas para los salones de los edificios en la FI-UAQ.

	Bandera							
Licenciatura	1	2	3	4	5	6	7	Total
Anim. Dig.					8			8
Arq.				10	36	6		50
Auto.	61							62
Civil	114							116
Diseño					25		1	26
Ind.&Man.			45				6	50
Mat.			37					37
Tronco común		31						31
Total	175	31	82	10	69	6	7	380

Cuadro 3.2: Banderas especificadas para los cursos de las licenciaturas en la FI-UAQ.

La especificación de banderas debe hacerse cuidadosamente, analizando cuántas categorías se necesitan para cumplir con las preferencias de asignación y enlistarlas en orden de similitud. Por ejemplo, los salones se pueden abanderar de acuerdo a su ubicación: los salones del mismo edificio o inlcuso en edificios cercanos abanderarlos con el mismo número. Se debe seguir una manera estandarizada para asignar las banderas para obtener una asignación de salones coherente. Si se establecen relaciones de banderas contradictorias, como relacionar un curso con un salón que no es lo suficientemente grande para que todos los estudiantes se sienten tamaño(c) >tamaño(s), el programa de asignación de salones puede corregir estos errores humanos si en el código se especifica que el objetivo "tamaño" debe tener más peso que el objetivo "bandera".

CAPÍTULO 4

Resultados y discusión

4.1. Discusión

El problema de asignación de salones para 380 cursos y 35 salones tiene 42,302 restricciones y 2,473,800 variables de decisión binarias y auxiliares de las cuales 308,140 son variables diferentes de cero. A pesar de la magnitud del problema se resolvió 99.9 % más rápido que con el método manual ahorrando muchas horas de trabajo y cumpliendo con las restricciones de cupo, horario, profesores y herramientas para los 13 programas de estudio de la FI-UAQ. El problema se resolvió con algoritmos de optimización como se propuso inicialmente.

La asignación de salones para el periodo 2019-1 se realizó con una versión del programa que no incluía funciones objetivo porque en el momento no se conocía la manera de programarlas. La Maestra Carmen Sosa Garza, secretaria académica de la FI, nos ayudó revisando si los horarios de todas las licenciaturas de la FI-UAQ tenían errores como cruces en los horarios de los profesores. Después al archivo de horarios se le hicieron algunas modificaciones porque no cumplía con el formato del archivo de entrada, como: espacios en las columnas de horarios, "punto y coma" en lugar de "dos puntos" y el formato de hora. La primer asignación, sin funciones objetivo, tomó aproximadamente 20 horas porque las características que revisan las funciones objetivo se intentaron hacer manualmente dividiendo el archivo por categorías (grupos grandes, pequeños, por licenciaturas) y corriendo el programa varias veces hasta obtener una solución aceptable. Fue un proceso tedioso porque para cumplir con las restricciones de cupo y preferencias de asignación fue necesario modificar los diferentes archivos de entrada que se habían generado para las diferentes categorías y encontrar una combinación de salones para que todos los cursos estuvieran asignados a un salón. La maestra Cecilia Hernández Garciadiego, quien era la encargada administrativa de dicho proceso, nos sugirió varias combinaciones que le funcionaban en el proceso manual y fue así como se obtuvo una propuesta de asignación de salones. El objetivo principal fue realizar combinaciones de manera que la asignación cumpliera con colocar a Arquitectura en el edificio I, a Ingeniería Civil e Ingeniería en Automatización en el edificio B y C, a la Lic. Matemáticas Aplicadas en el edificio E y los grupos de tronco común en la menor cantidad de salones como se muestra en el Cuadro 4.1. El resto de los programas de estudio al ser nuevos y con poca asistencia tienen menor cantidad de cursos y son grupos pequeños, por lo tanto podían ser acomodados en los espacios restantes.

Las primeras pruebas revelaron que el principal factor para la mínima cantidad de salones que se pueden asignar son las coincidencias en horarios. Debido a que la mayoría de los cursos se asignan de 10:30-11:00 de la mañana es complicado cumplir con asignaciones preferidas. La mayoría de los salones disponibles son lo suficientemente grandes para el tamaño de los grupos, por lo tanto este factor no complicó la asignación en esta primera etapa. A pesar de haber tomado más tiempo de lo esperado la versión sin funciones objetivo disminuyó el tiempo original de 80 hrs un 75 % y cumplió con la restricción de asignar todas las sesiones del curso en un mismo salones; el archivo de salida de la asignación por día (Figura 4.1) fue esencial para realizar esta primera asignación de salones ya que permitía identificar fácilmente los espacios libres.

Programa de estudio	Edificio preferido
Licenciatura en Arquitectura	I
Ingeniería Civil	B&C
Ingeniería en Automatización	B&C
Licenciatura en Matemáticas Aplicadas	E
Tronco común	Mínima cantidad de salones
Ingeniería en Diseño Industrial	-
Licenciatura en Animación Digital	-
Ingeniería Industrial y Manufactura	-

Cuadro 4.1: Relaciones preferidas entre programas de estudio y edificios.

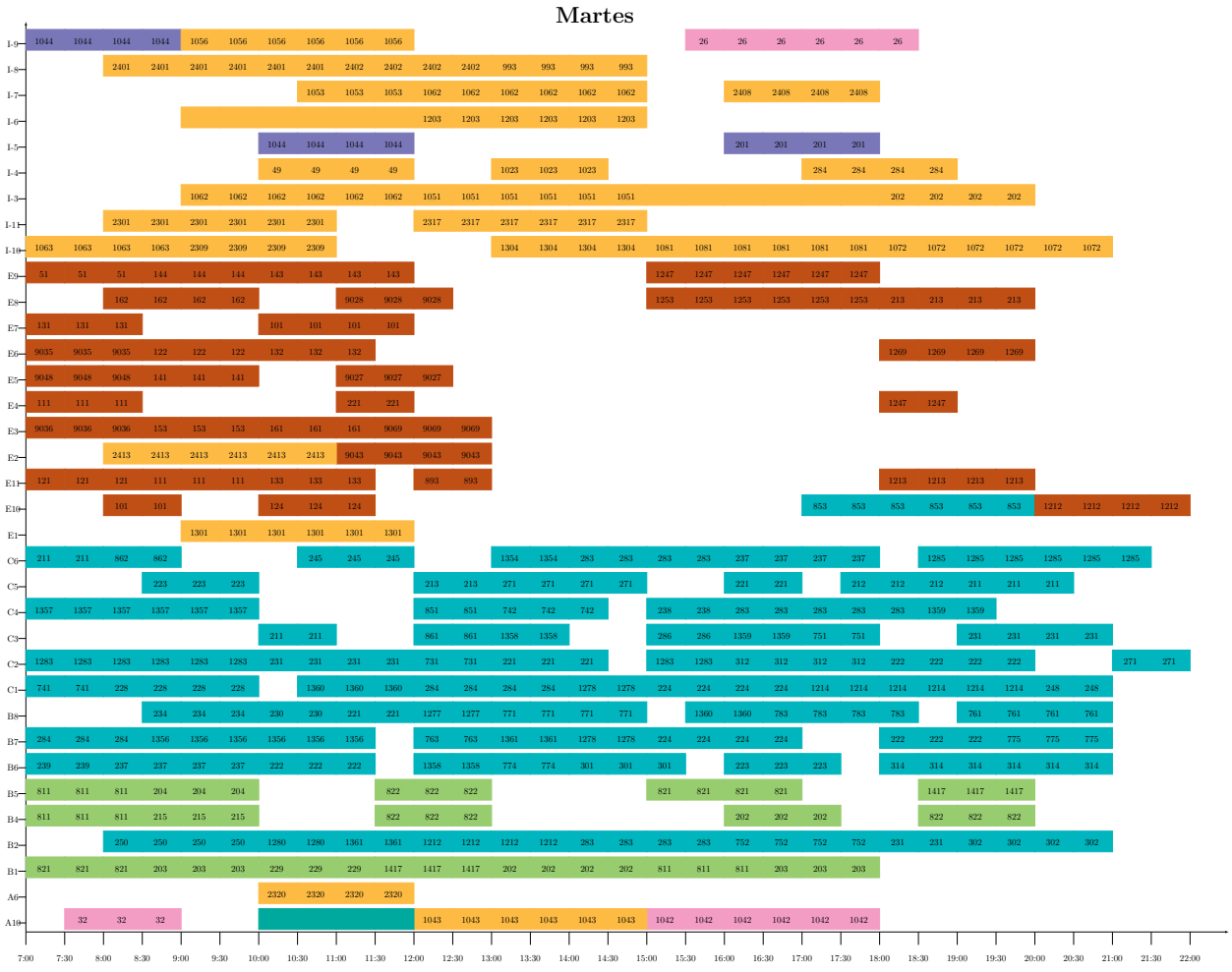


Figura 4.1: Archivo de salida del día Martes para la instancia real del periodo 2019-1.

Las principales modificaciones de la segunda etapa de prueba fueron la programación de las funciones objetivo f_1 y f_2, imprimir el nombre del salón en el archivo separado por comas con los horarios de los

cursos, los archivos de salida de las gráficas de asignación por salones, la modificación en el archivo de salida de las gráficas de asignación por días para que imprimiera los colores de acuerdo a las banderas establecidas e imprimir la clave del curso en lugar de un nombre genérico. Las modificaciones fueron muy significativas porque se corrió el programa en una sola exhibición obteniendo una asignación aceptable. En total se imprimieron 6 formatos de asignación por día y 34 formatos de asignación por salón, ahorrando muchas horas de trabajo extra que no se habían considerado como objetivo inicialmente. En total la corrida tomó 48.41 segundos, que significa una reducción del 99.9 % del tiempo en comparación con el método manual.

4.2. Programa multi-objetivo

4.2.1. Resultados utilizando diferentes combinaciones de pesos ponderados

La primera prueba con el modelo multiobjetivo acomodó 13 cursos (3 %) a salones que violaban la función objetivo f_1, 8 de estos cursos habían sido abanderados incorrectamente (a salones pequeños) y los otros 5 cursos tenían horarios que se empalmaban con varios cursos, quedando libres únicamente salones pequeños en esos periodos. Dado que la FI-UAQ solicitó que la función objetivo f_1 se violara la menor cantidad de veces, se realizaron una serie de pruebas iterativas para analizar cuáles combinaciones de pesos ponderados se ajustaban mejor a los requerimientos de la instancia.

	Peso		Valor			Cuenta			Porcentaje		
Prueba	α_1	α_2	f_1	f_2	Total	f_1	f_2	Total	f_1	f_2	Total
1	0.0	1.0	233.0	8.4	241.4	368	361	729	96.8 %	95.0 %	95.9 %
2	**0.1**	**0.9**	216.8	8.2	225.0	378	362	740	99.5 %	95.3 %	97.4 %
3	**0.2**	**0.8**	216.3	8.3	224.6	379	361	740	99.7 %	95.0 %	97.4 %
4	**0.3**	**0.7**	216.3	8.3	224.6	379	361	740	99.7 %	95.0 %	97.4 %
5	0.4	0.6	215.1	10.1	225.1	379	350	729	99.7 %	92.1 %	95.9 %
6	0.5	0.5	213.5	11.4	225.0	380	350	730	100.0 %	92.1 %	96.1 %
7	0.6	0.4	213.2	11.8	225.0	380	349	729	100.0 %	91.8 %	95.9 %
8	0.7	0.3	212.3	13.6	225.9	380	346	726	100.0 %	91.1 %	95.5 %
9	0.8	0.2	211.3	14.3	225.6	380	342	722	100.0 %	90.0 %	95.0 %
10	0.9	0.1	209.1	29.2	238.3	380	333	713	100.0 %	87.6 %	93.8 %
11	**1**	**0.0**	216.3	8.3	224.6	379	361	740	99.7 %	95.0 %	97.4 %

Cuadro 4.2: Resultados de las pruebas con pesos ponderados.

El programa escrito en Python lee el horario (establecido por los coordinadores) en un formato muy similar al que ya se utilizaba para diseñar los horarios de los cursos, esto ahorra mucho tiempo debido a que no se requiere invertir tiempo en ajustar el formato para que el programa pueda leerlo. En el Cuadro 4.3 se muestra un ejemplo del formato de archivo de entrada para el horario y en el Cuadro 4.4 es el archivo de entrada complementario que indica los salones que se pueden utilizar para la asignación, su tamaño y banderas establecidas de acuerdo a las preferencias de asignación.

El horario para el periodo 2019-1 consistió de 380 cursos que debían ser asignados en 35 salones los cuales se abanderaron como se muestra en el Cuadro 3.1 y el Cuadro 3.2. El porcentaje de objetivo logrado en el Cuadro 4.2 se calculó contando la cantidad de asignaciones con un valor objetivo aceptable.

Se buscó una combinación de pesos ponderados que propusiera una asignación con porcentajes de "objetivo logrado" altos y balanceados, dando prioridad a la función objetivo f_1 ya que si el número de estudiantes en un curso es mayor que el número de sillas en el salón habrá estudiantes sin lugar. La prueba 2, 3, 4 y 11

BANDERA	TAMAÑO	CURSO	PROFESOR	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
1	20	201	685	11:00-12:00		11:00-12:00		11:00-12:00	
1	20	204	11691	16:00-19:00	12:00-14:00		12:00-14:00		
1	20	205	2891	17:30-20:30		17:30-20:30		17:30-20:30	
1	20	206	4775	19:00-20:00	9:00-11:00		9:00-11:00		
1	20	208	13648	20:00-22:00		20:00-22:00		20:00-22:00	
1	20	209	1266	7:00-8:00		7:00-8:00			7:00-8:00
1	20	301	3836	8:00-9:00		8:00-9:00		8:00-9:00	
1	20	303	4478	9:30-11:00		9:30-11:00			9:30-11:00
2	30	200	3353	10:00-11:30		10:00-11:30		10:00-11:30	
2	30	202	4077	15:00-16:30	11:30-13:00		11:30-13:00		
2	30	203	8545	15:00-17:00		15:00-17:00		15:00-17:00	
2	30	207	8530	19:00-20:30		19:00-20:30		19:00-20:30	
2	30	210	10047	7:00-8:30	8:30-10:00		8:30-10:00		
2	30	300	6333	7:00-8:30		7:00-8:30		7:00-8:30	
2	30	302	2891	8:30-10:00	20:00-22:00		20:00-22:00		
2	20	304	4478	9:00-11:00		9:00-11:00		9:00-11:00	9:00-11:00
2	20	305	4478	12:00-14:00		12:00-14:00		12:00-14:00	
2	20	306	4478	11:30-13:00		11:30-13:00		11:30-13:00	

Cuadro 4.3: Ejemplo de archivo de entrada: Cursos

SALÓN	TAMAÑO	BANDERA
B2	35	1
B4	35	2
B5	35	2

Cuadro 4.4: Ejemplo de archivo de entrada: Salones

del Cuadro 4.2 tiene la suma de porcentajes de "objetivos logrados" más alta como se puede observa en la Figura 4.2 que muestra la cantidad de asignaciones con objetivos satisfactorios de los 380 cursos asignados.

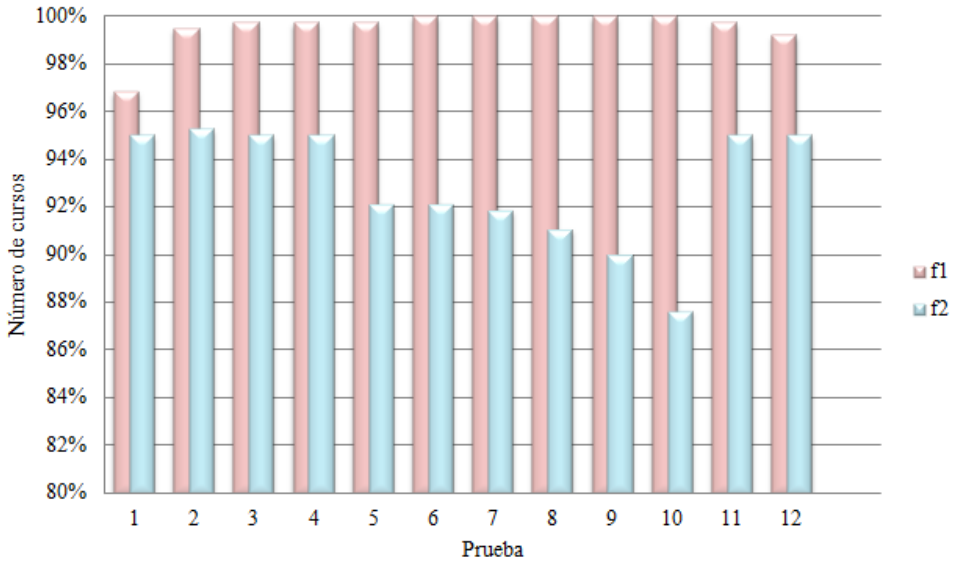


Figura 4.2: Cuenta de los cursos con un valor objetivo satisfatorio para las diferentes combinaciones de pesos ponderados probadas.

El programa de asignación de salones imprime tres documentos de salida:

1. El archivo de entrada del horario de cursos con tres columnas extra, como se muestra en el Cuadro 4.5, en las que se imprime el nombre del salón asignado, el valor de la función objetivo f_1 y el valor de la función objetivo f_2. Estos valores permiten analizar fácilmente cuales asignaciones no se cumplieron y en qué grado.

2. Dos tipos de gráficas, una con la asignación por día como se muestra en la Figura 4.3,

3. y otra con la asignación por salón como se muestra en la Figura 4.4.

Clave_Curso	Clave_Prof	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Salon	f1	f2
200	3353	10:00-11:30		10:00-11:30		10:00-11:30		B5	0.85714286	0
201	685	11:00-12:00		11:00-12:00		11:00-12:00		B2	0.57142857	0
202	4077	15:00-16:30	11:30-13:00		11:30-13:00			B4	0.85714286	0
203	8545	15:00-17:00		15:00-17:00		15:00-17:00		B5	0.85714286	0
204	11691	16:00-19:00	12:00-14:00		12:00-14:00			B2	0.57142857	0
205	2891	17:30-20:30		17:30-20:30		17:30-20:30		B5	0.57142857	0.5
206	4775	19:00-20:00	9:00-11:00		9:00-11:00			B2	0.57142857	0
207	8530	19:00-20:30		19:00-20:30		19:00-20:30		B4	0.85714286	0
208	13648	20:00-22:00		20:00-22:00		20:00-22:00		B2	0.57142857	0
209	1266	7:00-8:00		7:00-8:00		7:00-8:00		B2	0.57142857	0
210	10047	7:00-8:30	8:30-10:00		8:30-10:00			B4	0.85714286	0
300	6333	7:00-8:30		7:00-8:30		7:00-8:30		B5	0.85714286	0
301	3836	8:00-9:00		8:00-9:00		8:00-9:00		B2	0.57142857	0
302	2891	8:30-10:00	20:00-22:00		20:00-22:00			B5	0.85714286	0
303	4478	9:30-11:00		9:30-11:00		9:30-11:00		B2	0.57142857	0
304	4478	9:00-11:00		9:00-11:00		9:00-11:00		B4	0.57142857	0
305	4478	12:00-14:00		12:00-14:00		12:00-14:00		B5	0.57142857	0
306	4478	11:30-13:00		11:30-13:00		11:30-13:00		B4	0.57142857	0

Cuadro 4.5: Ejemplo de archivo de salida con las tres columnas que especifican: salón asignado, valor objetivo f_1 y valor objetivo f_2.

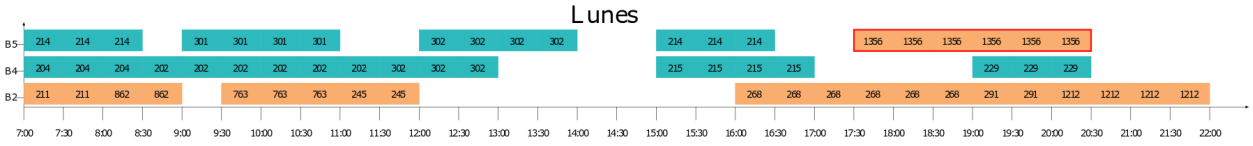


Figura 4.3: Ejemplo de archivo de salida del día Lunes.

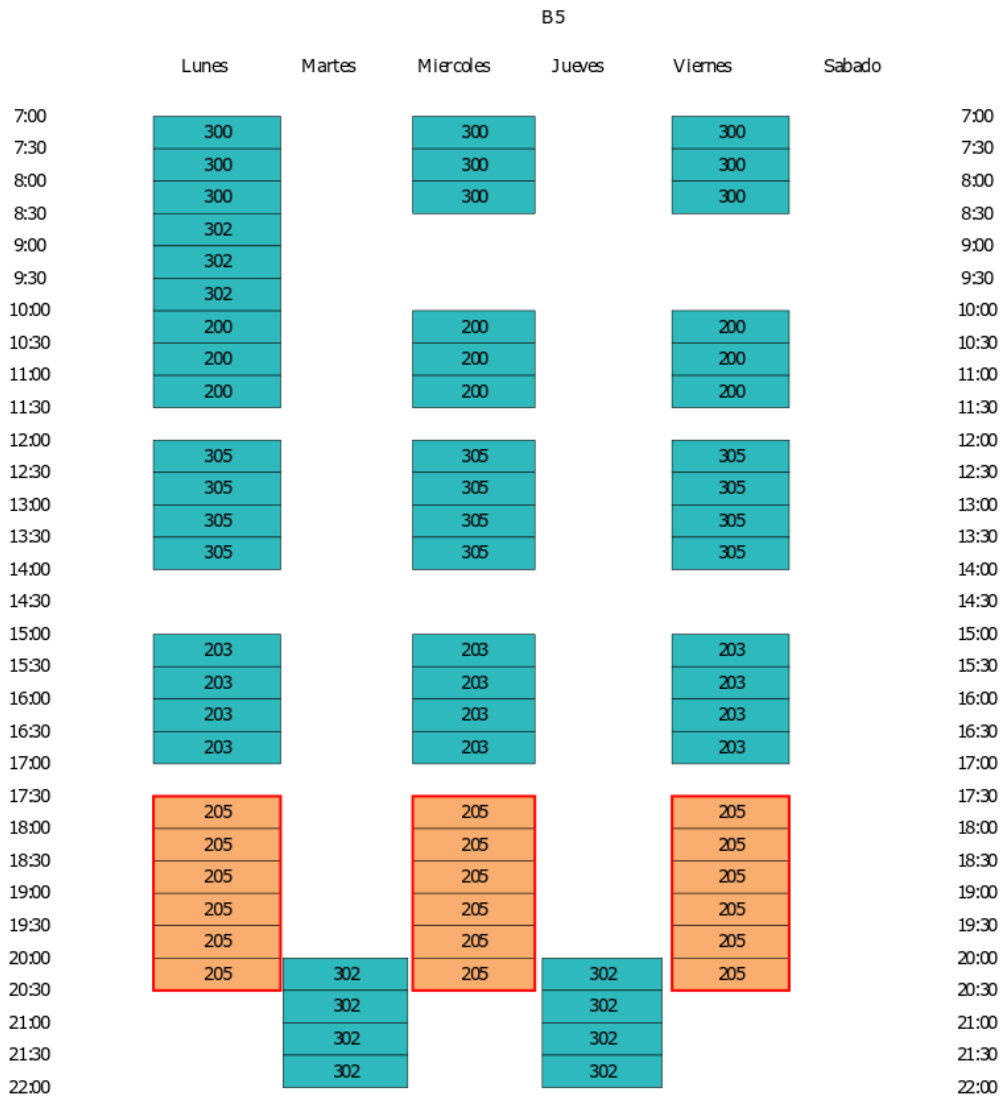


Figura 4.4: Ejemplo de archivo de salida para el salón C2.

Las gráficas fueron muy útiles para identificar periodos *populares*, rangos de tiempo en los que se programan gran cantidad de cursos. Estos periodos *populares* ocasionan que se requiera una mayor cantidad de salones que los 35 disponibles en la FI-UAQ para que una solución factible exista. Se utilizó el formato por días (Figura 4.3) para identificar los periodos *populares*, se informó a los coordinadores involucrados y se solicitó que se realizaran ajustes de horarios dentro de lo posible para obtener soluciones factibles. El formato por salones (Figura 4.4) solía realizarse a mano en un archivo de Excel y pegarse fuera de los salones para identificar periodos en los que los salones estuvieran desocupados en caso de que los alumnos quisieran estudiar o la administración requiriera disponer de espacios libres para actividades no-curriculares. La impresión automática de los 35 archivos ahorra muchas horas de trabajo, fue un objetivo secundario muy satisfactorio que no se habían considerado al inicio de la investigación.

En la Figura 4.3 y la Figura 4.4 los cursos se colorearon de acuerdo a su número de bandera, se puede observar que el curso “1356” se tuvo que asignar a un salón con un número de bandera diferente al preferido porque el salón “B2” ya tenía asignados en esos horios a los cursos “268”, “291”, y “1212”. Así es como

actúa el programa de asignación de salones con la configuración No.5 cuando demasiados cursos con la misma bandera se empalman.

4.3. Importancia e impacto

4.3.1. Impacto social

1. El programa de asignación de salones redujo el tiempo de dicha actividad un 99.9 % por lo que ya no será necesario realizar la asignación de salones durante periodos vacacionales.
2. La manera en que el programa realiza la asignación de salones disminuye el flujo de alumnos en los pasillos, evitando pérdida de tiempo y ruido para otros cursos.
3. El programa realiza una asignación homogénea, este tipo de asignación le facilita a alumnos y profesores recordar en qué salón tienen clase, pues un curso se impartirá en el mismo salón para todas las sesiones de la semana.
4. Las gráficas de asignación por salón permiten visualizar fácilmente los espacios libres en los salones para que los alumnos puedan utilizarlos para estudiar.
5. Al aprovechar los salones adecuados para tomar cursos de teoría y no tener que utilizar otros salones equipados con restiradores, salones pequeños, lejanos o con bancas incómodas que generalmente eran la solución cuando faltaba espacio con el método manual, los alumnos y profesores estarán más agusto durante las clases facilitando el aprendizaje.

4.3.2. Impacto ambiental

1. Se aprovechan al máximo los espacios disponibles evitando construcciones que generan un impacto ambiental.
2. La reducción de tiempo que se requiere para realizar la actividad reduce el uso de la computadora, disminuyendo el consumo de luz.

4.3.3. Impacto económico

1. El archivo de salida de asignación por día permite identificar los horarios más comunes que complican la asignación en una menor cantidad de salones, solicitando cambios de horarios para cursos específicos se evitan futuras inversiones inecesarias en infraestructura.
2. La reducción de 99.9 % de tiempo requerido para realizar la asignación de salones ahorra más de 79 horas de trabajo de dos personas que pueden dedicarse a otras actividades.

4.4. Trabajo futuro

El programa presentado de asignación de salones utiliza un solucionador comercial, por lo tanto se propone escribir un programa que utilice el algoritmo de ramificación-y-acotamiento para solucionar instancias sin requerir solucionadores comerciales. También será necesario diseñar una interfaz amigable para que cualquier usuario pueda realizar la asignación de salones automatizada.

Se recomienda mapear el proceso de la programación de horarios en la FI-UAQ e incluir en el proceso una etapa en la que se retroalimente a los coordinadores sobre los horarios más comunes para fomentar que programen cursos en todos los horarios y asi evitar el congestionamiento de salones, estacionamientos, pasillos, etc.

Finalmente se recomienda desarrollar la lógica que verifique los cruces en los horarios de los profesores. Es común que un profesor que enseña en más de una licenciatura tenga asignado en el mismo horario más de un curso, actualmente los cruces se verifican manualmente antes de realizar la asignación de salones pero aún así, hay casos que quedan sin corregir. Los cruces son la causa principal de las modificaciones de último minuto, por lo que asegurar que los horarios no tengan errores evitará retrasos en la publicación de horarios o confusiones que afecten el inicio de clases.

CAPÍTULO 5

Conclusión

El problema de asignación de salones en la Facultad de Ingeniería UAQ se resolvió utilizando programación lineal entera. Se diseñó un modelo de optimización multiobjetivo, se programó en Python y se resolvió con el solucionador de optimización comercial Gurobi. El modelo multiobjetivo se trató como un modelo mono-objetivo con el método de suma ponderada. Se realizaron 11 pruebas con diferentes combinaciones de pesos para comparar y elegir qué pesos proponen una asignación de salones con el mayor porcentaje de restricciones suaves satisfactorias. La asignación de 380 cursos del semestre 2019-1 se realizó en 39.44 segundos (99.9 % más rápido que con el método manual). Las funciones objetivo del modelo multiobjetivo permiten indicar el tamaño de los grupos y salones, y especificar preferencias de asignación a nivel de curso. Se cumplieron los objetivos planteados inicialmente:

1. Con algoritmos de optimización se redujo el tiempo que toma la asignación de salones, aprovechando recursos humanos.
2. El modelo cumple las restricciones de cupo, programación de horarios y especificaciones para realizar la asignación.
3. Con la metodología presentada es posible programar todos los cursos de los 13 programas de estudio de la FI-UAQ 99.9 % más rápido en comparación con la estrategia manual.
4. Las restricciones duras determinadas fueron la asignación de un único salón por curso durante un intervalo de tiempo, un único curso por salón durante un intervalo de tiempo, y realizar asignaciones homogéneas. Las restricciones suaves fueron asignar de acuerdo al tamaño del grupo y salón, y respetando preferencias de asignación.
5. La estrategia implementada para la asignación de salones fue respetar el formato de horarios utilizado por los administrativos para leer el horario programado por los coordinadores y asingar salones automáticamente a nivel de curso utilizando algoritmos de optimización.

Los resultados obtenidos fueron aprobados por las autoridades de la FI-UAQ y se continuará trabajando los próximos semestres para implementar el proceso automatizado de asignación de salones en la asignación de horarios.

Bibliografía

[Abdennadher et al., 2000] Abdennadher, S., Saft, M., & Will, S. (2000). Classroom Assignment using Constraint Logic Programming. *Proceedings of the Second International Conference and Exhibition on The Practical Application of Constraint Technologies and Logic Programming (PACLP 2000)*.

[Abdullah et al., 2007] Abdullah, S., Burke, E. K., & McCollum, B. (2007). A hybrid evolutionary approach to the University Course Timetabling Problem. *IEEE Congress on Evolutionary Computation*, (pp. 1764–1768).

[Abell, 1965] Abell, V. A. (1965). Purdue academic student scheduling (PASS). Goals, guidelines and operations.

[Burke & Ross, 1995] Burke, E. & Ross, P. (1995). Practice and theory of automated timetabling.

[Burke et al., 1997] Burke, E. K., Jackson, K. S., Kingston, J. H., & Weare, R. F. (1997). Automated timetabling: the state of the art. *The Computer Journal*, 40(9), 565–571.

[Carpaneto & Toth, 1987] Carpaneto, G. & Toth, P. (1987). Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, 18, 137–153.

[Carraresi & Gallo, 1984] Carraresi, P. & Gallo, G. (1984). A multi-level bottleneck assignment approach to the bus drivers ' rostering problem. *European Journal of Operational Research*, 16, 163–173.

[Carter, 1989] Carter, M. W. (1989). A Lagrangian Relaxation Approach To The Classroom Assignment Problem. *INFOR: Information Systems and Operational Research*, 27(2), 230–246.

[Carter & Tovey, 1992] Carter, M. W. & Tovey, C. A. (1992). When Is the Classroom Assignment Problem Hard? *Operations Research*, 40(1-supplement-1), S28–S39.

[Colorni et al., 1998] Colorni, A., Dorigo, M., & Maniezzo, V. (1998). Methaheuristics for high school timetabling. *Computational Optimization and Applications*, 9, 275–298.

[Constantino et al., 2010] Constantino, A. A., Filho, W. M., & Landa-silva, D. (2010). Iterated Heuristic Algorithms for the Classroom Assignment Problem. *International Conference on the Practice and Theory of Automated Timetabling (PATAT 2010)*.

[Erben & Keppler, 1995] Erben, W. & Keppler, J. (1995). A genetic algorithm solving a weekly course-timetabling problem.

[EVENTMAP, 2016] EVENTMAP (2016). PATAT Conferences.

[Harper, 2009] Harper, J. (2009). IBM 7090/94 Architecture.

[Hax, Bradley, 1977] Hax, Bradley, M. (1977). Integer Programming. In *Applied Mathematical Programming* chapter 9, (pp. 273–300). Addison-Wesley.

[Hayrapetyan, 2017] Hayrapetyan, A. (2017). Timetable automation using Genetic Algorithms.

[Hillier & Lieberman, 2010] Hillier, F. S. & Lieberman, G. J. (2010). *Introduction to Operations Research*. McGraw Hill, 9th edition.

[Karp, 1972] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85–103). Springer.

[Laporte & Desroches, 1986] Laporte, G. & Desroches, S. (1986). The problem of assigning students to course sections in a large engineering school. *Pergamon Journals Ltd*, 13(4), 387–394.

[Martinez-Alfaro & Flores-Teran, 1998] Martinez-Alfaro, H. & Flores-Teran, G. (1998). Solving the classroom assignment problem with simulated annealing. *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, 4, 3703–3708.

[McCollum, 2010] McCollum, B. (2010). The second international timetabling competition.

[Optimization, 2018] Optimization, G. (2018). Gurobi optimizer reference manual.

[Oude Vrielink et al., 2016] Oude Vrielink, R. A., Schepers, D., & Jansen, E. A. (2016). Practices in timetabling in higher education institutions: a systematic review. *Proceedings of the 11th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016)*, (pp. 1–10).

[Pérez, 2017] Pérez, S. L. (2017). *Personnel assignment problems through the multidimensional assignment problem*. PhD thesis, Universidad Autónoma Metropolitana.

[Phillips et al., 2015] Phillips, A. E., Waterer, H., Ehrgott, M., & Ryan, D. M. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53, 42–53.

[Skoullis et al., 2017] Skoullis, V. I., Tassopoulos, I. X., & Beligiannis, G. N. (2017). Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm. *Applied Soft Computing Journal*, 52, 277–289.

[Talbi, 2009] Talbi, E.-G. (2009). *Metaheuristics From design to implementation*. John Wiley & Sons, Iinc.

[Tassopoulos & Beligiannis, 2012] Tassopoulos, I. X. & Beligiannis, G. N. (2012). Solving effectively the school timetabling problem using particle swarm optimization. *Expert Systems with Applications*, 39(5), 6029–6040.