Modelo Matemático para la Programación de Horarios, De Cursos Universitarios, Aplicando una Técnica Metaheurística

Por

José Gerardo Córdoba Hernández

Tesis sometida en cumplimiento parcial de los requerimientos para el grado de:

MAESTRÍA EN CIENCIAS

en

MATEMÁTICAS (APLICADAS)

UNIVERSIDAD DE PUERTO RICO

RECINTO UNIVERSITARIO DE MAYAGÜEZ

2018

Aprobada por:

Pedro Vásquez Urbano, D. Sc. Presidente, Comité Graduado	Fecha
Mercedes S. Ferrer Alameda, M. E. Miembro, Comité Graduado	Fecha
Edgardo Lorenzo González, Ph.D Miembro, Comité Graduado	Fecha
Raúl E. Macchiavelli, Ph.D Representante, Estudios Graduados	Fecha
Olgamary Rivera Marrero, Ph. D. Directora del Departamento	Fecha

Resumen

En la presente investigación se desarrolló un modelo matemático que resuelve un problema de asignación de horarios. El problema consiste en asignar eventos (clases), carga académica al profesor y salones adecuados, para atender estudiantes en cada curso. La función objetivo busca maximizar las preferencias por cursos y horario del profesor, disminuyendo la cantidad de secciones superpuestas del mismo curso.

La programación de horarios se clasifica como un problema NP-completo, por lo cual se utilizan algoritmos genéticos para resolver el modelo presentado en esta investigación. La solución del problema se obtuvo usando el software MatLab 2015 y se diseñó una interfaz fácil de utilizar por el usuario. Los resultados obtenidos muestran que el algoritmo desarrollado permite obtener una solución cercana al óptimo en tiempo razonable, la misma da una buena aproximación con el horario real que se utilizó como prueba.

Abstract

The following research presents a mathematical model which was developed to solve a scheduling assignment problem. The problem consists in assigning events (classes), academic load to the professor, and adequate classrooms to attend students for each course. The objective function seeks to maximize scheduling preferences by course and professor, reducing the amount of overlapping sections of the same course.

The scheduling program is classified as an NP-complete problem, by which genetic algorithms were used to solve the model presented in this research. The solution to the problem was obtained using the MatLab 2015 software and an interface was designed that is easy to use by the user. The obtained results demonstrate that the developed algorithm allows to obtain a solution close to the optimal one at a reasonable time, the solution gives a good approximation to the real schedule that was used for testing.

A Dios y mi madre Haydee Hernández Muñoz

AGRADECIMIENTOS

Un agradecimiento especial a la Universidad Nacional de Costa Rica por el apoyo brindado para realizar mis estudios en el exterior.

Doy gracias Dios por haberme permitido llegar hasta acá y proporcionado la salud para lograr mis metas. Agradezco a la Universidad de Puerto Rico, Recinto de Mayagüez quien por medio del Departamento de Ciencias Matemáticas me entregó una formación que me ha dado valor profesional y a sus profesores por compartir sus conocimientos y experiencias en el área de la matemática.

A mi madre, Haydee Hernández por apoyarme y guiarme desde niño a ser un profesional. Muchas gracias a mis amigos, Emilio Calderón Gómez, Daniel Ovalle Peña por toda la ayuda y momentos agradables que me brindaron.

Al Dr. Víctor Ocasio González por su tiempo y aportes. A la Prof. Mercedes S. Ferrer Alameda y Dr. Edgardo Lorenzo, miembros del Comité por su aporte y sugerencias para mejorar la versión final de la tesis.

Índice General

1. Introducción **1**

1.1. Objetivos 3

1.1.1. Objetivo general 3

1.1.2. Objetivos específicos 3

1.2. Limitaciones de las soluciones obtenidas en programación horaria 3

1.3. Organización del documento 4

2. Revisión Bibliográfica **5**

2.1. Clasificación de problemas de programación horaria 5

2.2. Métodos de solución 6

2.2.1. Concepto de optimización 6

2.2.2. Solución al problema de asignación horaria 7

2.3. Algoritmos genéticos 8

2.4. Algunas Soluciones al Problema de Programación Horaria 9

2.4.1. Programación lineal 9

2.4.2. Algoritmos genéticos 15

2.4.3. Búsqueda tabú 18

2.4.4. Colonia de hormigas 20

3. Modelo general y algoritmo **23**

3.1. Descripción del modelo general 23

3.2. Algoritmo 26

3.2.1. Matrices de asignación de salones y profesores 28

3.2.2. Matriz Cursos-Eventos 29

3.2.3. Representación de preferencias de profesores 30

3.2.4. Disponibilidad de salones 31

3.2.5. Construcción de la población inicial 32

3.2.6. Operador de recombinación de genes 34

3.2.7. Operador de Mutación 34

3.2.8. Representación final de un horario factible 36

4. Caso de estudio y resultados **38**

4.1. Programas de estudios 39

4.2. Conjuntos de bloques de horarios 39

4.3. Cursos con características especiales 40

4.4. Patrones de horarios 41

4.5. Resultados 42

4.5.1. Datos de entrada 42

4.6. Restricciones y función objetivo en el caso de estudio 44

4.6.1. Puntos importantes del algoritmo 46

4.6.2. Eventos no asignados 48

4.6.3. Análisis de la solución 48

4.7. Costo de ofrecimiento de los cursos 50

5. Conclusiones y trabajos futuros **52**

5.1. Conclusiones 52

5.2. Trabajos futuros 53

A. Interfaz 56

B. Patrones horarios en el caso de estudio 62

C. División de cursos en el caso de estudio 64

D. Matriz Pre_curso y Pre_bloque en el caso de estudio 66

E. Conversión crédito-evento en el caso de estudio 71

Índice de Tablas

4.1. Programas ofrecidos por el Departamento de Ciencias Matemáticas 39

4.2. Distribución y numeración de bloques horarios 40

4.3. Cursos con características especiales 40

4.4. Distribución de salones según su capacidad 42

4.5. Características del equipo 46

4.6. Pesos constantes y valor de función objetivo 47

4.7. Muestra de la solución, coincidencia 1 49

4.8. Muestra de la solución, coincidencia 2 49

4.9. Valor de la función objetivo costo 51

4.10. Muestra de la solución, costo mínimo 51

B1. Patrones horarios para eventos cuadro de horas 62

B2. Patrones de horarios para eventos de cuatro horas 62

B3. Patrones de horarios para eventos de cinco horas 63

E1. Tabla de conversión de créditos 71

Índice de Figuras

2.1. Operador recombinación de genes en el algoritmo genético 9

2.2. Operador de mutación en el algoritmo genético 9

2.3. Esquema Teórica-Práctica-Laboratorio 10

3.1. Estructura general del algoritmo 27

3.2. Matriz de salones 28

3.3. Matriz de profesores 29

3.4. Matiz Cursos-Eventos 29

3.5. Matriz CC 30

3.6. Representación de preferencias de profesores 31

3.7. Disponibilidad de salones 32

3.8. Pseudocódigo de la población inicial 33

3.9. Operador de cruce del modelo general 34

3.10. Pseudocódigo operador de recombinación de genes 35

3.11. Operador de mutación del modelo general 35

3.12. Pseudocódigo operador de mutación 36

3.13. Pseudocódigo del algoritmo 36

3.14. Representación final de un horario factible 37

Capítulo 1

Introducción

El problema de asignación horaria es una realidad que las instituciones educativas enfrentan al inicio de cada periodo lectivo. Dicho problema, está directamente relacionado al número de salones disponibles, profesores y cursos a ser ofrecidos, que lo puede convertir en un problema difícil de resolver.

Gran parte de las instituciones de educación superior resuelven este problema manualmente, este proceso puede tomar varios días o semanas para completar la asignación de horario a los profesores. La preparación de un horario en forma manual limita el análisis a un pequeño conjunto de las posibles soluciones, además puede ser necesario que se asigne una o más personas para realizar el trabajo de modo que pueda satisfacer las condiciones mínimas de la institución. Sin embargo, la responsabilidad de estas personas va más allá de solo asignar un salón donde se pueda ofrecer el curso. Se debe considerar que el salón tenga la capacidad para recibir la cantidad de estudiantes matriculados en el curso, que cuente con el equipo tecnológico y asignar a un profesor capacitado para ofrecer la clase, entre otros aspectos. En resumen, la asignación de horarios en las universidades, escuelas y colegios radica en programar las clases que se ofrecerán en un período académico determinado.

En la programación de horarios surgen algunas preguntas como: ¿cuál es el método óptimo para resolver el problema de programación horaria? ¿el método elegido dará una solución factible en un tiempo razonable? La teoría disponible menciona que no existe un consenso de respuesta ante estas preguntas. La mayoría de los investigadores ubican el problema de programación horaria entre aquellos de mayor complejidad en la programación entera, NP-completos,[1] debido a su tratamiento matemático y la comple-

[1]Son problemas intrínsecamente intratables desde el punto de vista computacional, si algún algoritmo resuelve dicho problema requerirá una cantidad de tiempo exponencial.

jidad del problema (Bejarano, 2010). Varios autores han tratado el problema de asignación horaria pero no existe consenso alguno sobre un modelo único o más eficiente para hallar una solución ya que puede variar de acuerdo con las restricciones y objetivos de cada institución (evitar cruces entre cursos del mismo programa, tener salones disponibles, evitar doble asignación de docentes, carga asignada al profesor, etc.).

Dada la complejidad del problema se opta por aplicar métodos heurísticos para resolver este tipo de problemas. La desventaja de los métodos heurísticos es que no garantizan encontrar soluciones óptimas o cercanas a las óptimas, sino que obtienen soluciones aceptables y hacen uso de recursos computacionales aceptables. Los métodos exactos, por ser de búsqueda exhaustiva, consumen muchos recursos computacionales cuando se aplican a problemas de este tipo.

En esta investigación se utilizarán algoritmos genéticos, que según la bibliografía, producen resultados aceptables en tiempos razonables. El algoritmo e interfaz se construyeron utilizando el programa MatLab 2015.

1.1. Objetivos

1.1.1. Objetivo general

- Proponer un modelo matemático para resolver el problema de programación horaria en una universidad.

1.1.2. Objetivos específicos

- Diseñar y formular un modelo matemático para obtener la solución del problema de programación horaria que cumpla con los requisitos y necesidades de la universidad.
- Definir las restricciones asociadas al modelo.
- Comparar los resultados del modelo propuesto con la programación real de cualquier semestre de la universidad.
- Evaluar la efectividad de los resultados obtenidos.
- Diseñar una interfaz que sea fácil de utilizar para el usuario.

1.2. Limitaciones de las soluciones obtenidas en programación horaria

Debido al tipo de problema, es imposible obtener una solución global al asunto de programación horaria en las instituciones de educación. Puesto que cada institución cuenta con características especiales en el momento de asignar cursos. Por ejemplo, las instituciones de educación cuentan con una infraestructura restringida, lo cual causa limitación de salones en cuanto a la cantidad y capacidad de atender estudiantes. Por esta razón en la presente investigación se busca plantear un modelo general que dé como solución un horario final o un punto de partida mediante los algoritmos genéticos. Es decir, debido a las características del problema es casi imposible llegar a una formulación matemática que se pueda aplicar a todas las instituciones de educación superior.

1.3. Organización del documento

El documento está organizado de la siguiente manera:

1. En el capítulo 2 se presentan la revisión bibliográfica. Donde se abordan los principales fundamentos teóricos para la programación horaria y se muestran trabajos que dan solución al problema de asignación horaria mediante diferentes métodos.

2. En el capítulo 3 se describe detalladamente el modelo general y las bases principales del algoritmo que darán solución al problema planteado.

3. En el capítulo 4 se describe el caso de estudio y los resultados obtenidos.

4. Finalmente en el capítulo 5 se presentan las conclusiones y recomendaciones para trabajos futuros.

Capítulo 2

Revisión Bibliográfica

Un problema importante que abordan las instituciones de educación superior es la preparación de horarios, para lograrlo muchas de ellas buscan automatizar esta actividad mediante una programación (algoritmo) matemática que facilite la preparación de estos. Para desarrollar un algoritmo que obtenga una solución aceptable al problema de programación horaria.

2.1. Clasificación de problemas de programación horaria

El problema de programación de horarios o calendarización consiste en el uso de recursos que deben ser asignados en intervalos de tiempos idóneos, teniendo presente las restricciones (condiciones) establecidas por la institución. El problema de programación de horario se aplica en diferentes áreas de trabajo, por tal razón se considera que su estudio es de gran relevancia para una institución educativa, aerolíneas, empresas, etc. Los tipos de programación más comunes según Guerra, Parto y Salas (2013) son:

- "Transport Timetabling": está relacionado con la asignación de rutas de los conductores de buses de transporte público o privado, trenes y aviones.
- "Sports Timetabling": establece el calendario de los equipos que participan en diversas áreas de deportes, que tienen como característica diferentes tipos de enfrentamientos entre equipos, ya sea de uno contra uno, ida y vuelta.
- "Employee Timetabling and Rostering": establece los turnos particularmente en el sector de la salud, puesto que, enfermeras y médicos deben cumplir diferentes turnos de trabajo, se deben equilibrar las

cargas de trabajo.

- "Educational Timetabling": en el ámbito educativo, los principales problemas son los de programación de horarios tanto en colegios (School Timetabling) como en universidades (University or Course Timetabling).

Esta investigación busca plantear un modelo general para dar solución a la programación de horarios en las instituciones educativas (Educational Timetabling). En el capítulo 3 se explica el modelo general que dará solución a este tipo de problemas.

2.2. Métodos de solución

En este capítulo se explican y mencionan los conceptos y técnicas más relevantes para entender y dar solución a este tipo de problemas, más específicamente a los de asignación de horarios académicos.

2.2.1. Concepto de optimización

Generalmente, se puede decir que optimizar es dar solución a un grupo determinado de alternativas donde se busca elegir la mejor solución dentro de una familia de soluciones válidas o factibles. Además de buscar la mejor solución, la optimización también tiene como objetivo encontrar dicha solución en el menor tiempo posible. Es aquí donde la teoría de la complejidad computacional es de gran importancia, porque estudia los recursos que son necesarios para que el algoritmo alcance una solución al problema.

Para Ahumada (2014), los recursos más importantes que son objeto de estudio son: el tiempo y el espacio. El tiempo se traduce en la cantidad de pasos de ejecución de un algoritmo para resolver el problema y el espacio se puede definir como la cantidad de memoria necesaria que utiliza la computadora para ejecutar el algoritmo.

Por su complejidad este tipo de problema se pueden clasificar en tres tipos principales: problemas P, problemas NP[1], problemas NP-completo (Cormen, Leiserson, Rivest y Stein 2009).

- **Problemas P**: es el conjunto de problemas de decisión cuya solución se puede obtener en tiempo polinomial.

[1]Las iniciales "NP" es el acrónimo en inglés de Polinómico No determinístico (Non-Deterministic Polynomial-time).

- **Problemas NP**: es el conjunto de problemas que se pueden resolver en un tiempo polinomial sobre una máquina de Turin[2] y la solución se verifica en tiempo polinomial (Cormen et al., 2009). El tiempo de procesamiento depende de la cantidad de datos de entrada (Guerra et al., 2013).

- **Problemas NP-completos**: los autores Montoya, Aponte y Rosas (2010) describen este problema como aquellos problemas NP para los que no es posible encontrar soluciones óptimas para instancias de gran tamaño en un tiempo de cálculo razonable. Son los problemas más difíciles de resolver en el conjunto NP y muy probablemente no formen parte de la clase de complejidad P. Según Bejarano (2010), si algún algoritmo resuelve correctamente un problema NP-completo, requerirá en el peor de los casos, una cantidad de tiempo exponencial para resolverlo.

Cooper y Kingston (1995) demuestran que el problema de programación horaria es un problema NP-completo. En la siguiente sección se explicarán algunas técnicas aplicadas para dar solución al problema de programación horario.

2.2.2. Solución al problema de asignación horaria

Las técnicas aplicadas para resolver el problema de programación horaria se clasifican en dos grandes grupos: técnicas tradicionales y no tradicionales. A continuación, se presentan estos dos enfoques de solución:

- **Técnicas tradicionales**: son métodos que recorren todo el espacio de búsqueda (región factible), por tanto, se dice que encuentran todas las soluciones factibles del problema, se les considera como algoritmos completos (Guerra et al., 2013). Podemos mencionar algunos de los algoritmos que pertenecen a esta categoría en el área de programación de horarios académicos: programación entera, programación lineal, "backtracking", entre otras.

- **Técnicas no tradicionales**: son métodos que no encuentran todas las posibles soluciones al problema, solo acotan o reducen el espacio de búsqueda, por lo tanto, se dice que son métodos incompletos (Mejía, 2010). Podemos mencionar algunos de los algoritmos que pertenecen a esta categoría en el área de programación de horarios académicos: Colonia de hormigas, redes neuronales (Neural Networks), recocido simulado (Simulated Annealing), búsqueda tabú (Tabu Search), algoritmos genéticos (Evolutionary Algorithms), entre otros.

[2]Es modelo abstracto que formaliza la idea intuitiva de algoritmo. Es un modelo matemático para representar a una maquina teórica. A pesar de su simplicidad tiene el mismo poder computacional de una computadora de propósito general.

En esta investigación se trabaja con los algoritmos genéticos, los cuales se describen en la siguiente sección.

2.3. Algoritmos genéticos

Para Suárez (2012) los algoritmos genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Se basan en el proceso genético de los organismos vivos, en el que, a lo largo de las generaciones, las poblaciones evolucionan en la naturaleza, de acuerdo con los principios de selección natural y la supervivencia de los individuos más fuertes.

La codificación básica para representar los algoritmos genéticos en forma de individuos se hace con unos y ceros, pero no es la única forma de hacerlo. Es decir, la representación se debe adecuar a las condiciones del problema, por ejemplo: pueden representarse como un conjunto de genes los cuales agrupados forman una ristra de valores llamadas cromosomas.

Los dos elementos principales en los algoritmos genéticos son: la representación de las posibles soluciones (cromosomas) y la función objetivo, la cual mide si la solución es aceptable. Además, los algoritmos genéticos siguen la siguiente estructura general de programación (Pitol, s.f).

1. Define la población inicial (posibles soluciones del problema).
2. Evalúa la población inicial mediante la función objetivo.
3. *Selección de padres*: permite seleccionar los mejores cromosomas para la reproducción. Las técnicas más comunes para seleccionar individuos son selección por ruleta: consiste en asignar a cada individuo un porcentaje proporcional de la ruleta dependiendo del valor de su función objetivo y selección por torneo: se comparan dos individuos mediante la función objetivo y el más apto es seleccionado (Ahumada, 2014).
4. *Recombinación de genes*: seleccionan a dos o más individuos (llamados padres), se toman las características de los mismo y se generan los nuevos individuos (hijos) (Figura 2.1).
5. *Mutación*: es una variación realizada a un individuo, el objetivo de este operador es brindar diversidad en la población (Figura 2.2).
6. Guardar la mejor solución obtenida respeto al valor de la función objetivo.

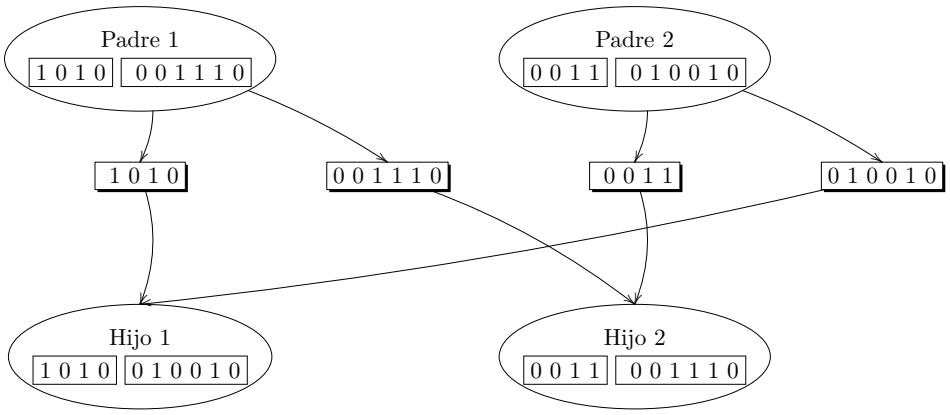


Figura 2.1: Operador recombinación de genes en el algoritmo genético

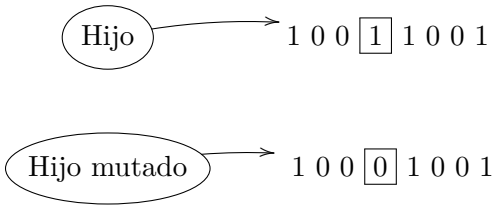


Figura 2.2: Operador de mutación en el algoritmo genético

7. Mientras no se cumpla la condición establecida para detener el algoritmo, vuelva a (3).

Finalmente, en el paso (7) se deben imponer condiciones de terminación del algoritmo genético, algunas de ellas son: un período de tiempo determinado, un número determinado de generaciones, un valor aceptable para la función objetivo (Pitol, s.f). Las soluciones metaheurísticas son interactivas, facilitando al usuario el aportar conocimientos al modelo (Suárez, 2011, p. 51).

2.4. Algunas Soluciones al Problema de Programación Horaria

En esta sección se describen algunos trabajos que tienen por objetivo resolver el problema de programación de horarios académicos usando diferentes métodos para determinar su solución.

2.4.1. Programación lineal

Méndez, Miranda y Zabala (2016) plantean diseñar horarios que contemplen simultáneamente los recursos disponibles y las preferencias de los estudiantes hacia las materias que desean matricular. El estudio se realizó en una universidad privada de Argentina.

Para cada programa la universidad clasifica las materias en obligatorias y electivas. Además, existen tres tipos de clases: teórica, práctica y laboratorio, las cuales se pueden combinar en tres posibles esquemas: teórica, teórica-práctica o teórica-práctica-laboratorio (Figura 2.3). Cada una tiene un patrón de horario definido por una frecuencia semanal y una cantidad de bloques de horarios continuos por vez.

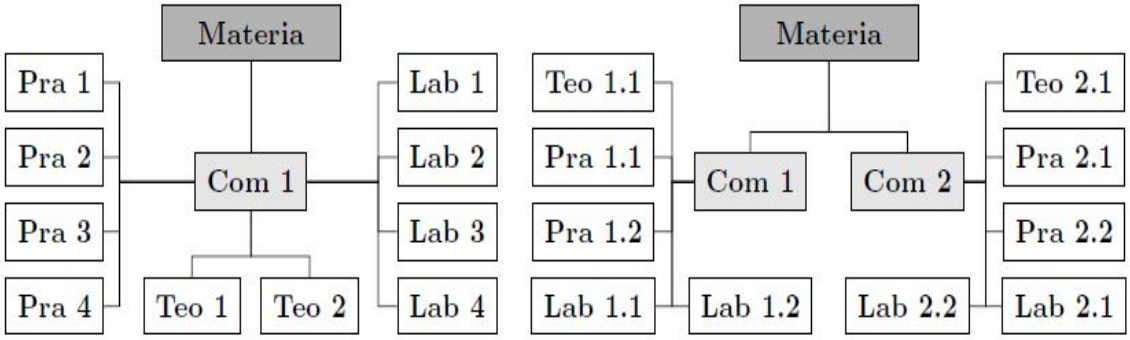


Figura 2.3: Ejemplo de esquema

Para resolver el problema, Méndez et al. (2016) formularon un modelo de programación lineal entera, con las siguientes variables de decisión binarias:

- x_{ac} toma valor 1 si el estudiante a es asignado a la clase c. Esta variable está definida sólo para aquellas clases c que correspondan a materias que se encuentran en la lista de preferencias del alumno a.

- xs_{acs} toma valor 1 si la clase c asignada al estudiante a es programada en el bloque de horario s. Esta variable está definida sólo para aquellas clases c que correspondan a materias que se encuentran en la lista de preferencias del esutdiante a y para bloques de horarios s disponibles del docente a cargo de la clase c.

- y_{cvs} toma valor 1 si el evento v de la clase c es programado en el bloque de horario s. Esta variable está definida sólo para los bloques de horarios s disponibles del docente a cargo de la clase c.

- yr_{crs} toma valor 1 si la clase c es asignada al aula r en el bloque de horario s. Esta variable está definida sólo para los bloques de horarios s disponibles del docente a cargo de la clase c y para las aulas r que tengan el tipo (las aulas están clasificadas según sus características en tipos, por ejemplo, disponibilidad de equipo multimedia, instrumentos de laboratorio. etc.) requerido por la clase.

- w_c toma valor 1 si la clase c es programada en el semestre.

- z_{fd} toma valor 1 si el docente f tiene asignada una clase el día d. Esta variable está definida sólo para los días d en que haya bloques de horarios disponibles para el docente f.

Con el objetivo de obtener un horario factible establecieron las siguientes restricciones:

1. Asignación estudiante-clases, consideran toda materia tiene clases teóricas por lo cual un estudiante está inscripto en una materia cuando está asignado a una de sus clases teóricas.
2. Factibilidad temporal, está restricción modela la asignación de bloques de horarios a clases impidiendo superposición horaria.
3. Factibilidad de aulas, no deben exceder la capacidad y no tener superposición horaria.
4. Factibilidad del horario para alumnos, garantiza que las materias programadas en bloques de horario no estén superpuestas, logrando que el estudiante puede asistir a todas las clases.
5. Factibilidad docente, las clases asignadas a los docentes no deben solaparse.

La función objetivo propuesta por Méndez et al. (2016) se enfoca en maximizar la satisfacción global de los estudiantes. Para cada estudiante $a \in \mathcal{A}$ consideran una suma ponderada entre la preferencia por la materia $m \in M^a$ y su desempeño académico definido por la universidad. La función objetivo está definida como:

$$\text{máx } z = \sum_{a \in A} \sum_{\substack{c \in T^m \\ m \in M^a}} (\alpha r k^a + \beta w^{am}) x_{ac}$$

$\mathcal{M}$: Conjunto de materias.
$\mathcal{A}$: Conjunto de estudiantes.
$M^a \subseteq \mathcal{M}$: Conjunto de materias elegidas por el estudiante $a \in \mathcal{A}$.
rk^a: Desempeño académico del estudiante $a \in \mathcal{A}$. El desempeño académico de cada alumno es un valor descendente entre 10 y 1.
T^m: Clases teóricas de la materia $m \in \mathcal{M}$.
w^{am}: Valor de preferencia para la materia $m \in M^a$ otorgado por el estudiante $a \in \mathcal{A}$

Se intentó resolver el problema mediante el software *CPLEX*, pero este presentó serias dificultades. En consecuencia, desarrollaron un proceso de dos fases. En la primera fase, consideran una relajación del modelo que incluye un subconjunto de estudiantes y relaja algunas restricciones. En una segunda etapa,

a partir de la asignación horaria obtenida en la primera etapa, se considera el modelo original fijando las variables correspondientes de asignación de bloque de horario a las clases y se busca una asignación de materias a estudiantes que maximice la preferencia global.

Los autores Marín y Maya (2016) presentan un problema en instituciones de educación primaria y secundaria. Consideran un modelo sin distinción entre los elementos del profesor y materia, así todo el modelo se basó en el establecimiento de las condiciones para cada materia, las cuales se relacionan directamente con su respectivo profesor.

El centro educativo debe tomar la decisión de qué materia dictar en cierto grupo. Marín y Maya (2016) definen las variables de decisión

$$A_{kl}^{ij} = \begin{cases} 1 & \text{El día } i \text{ a la hora } j \text{ se asigna la materia } l \text{ en el grupo } k \\ 0 & \text{El día } i \text{ a la hora } j \text{ NO se asigna la materia } l \text{ en el grupo } k \end{cases}$$

$$B_{kl}^{ij} = \begin{cases} 1 & \text{El día } i \text{ a la hora } j \text{ el grupo } k \text{ inicia un bloque de la materia } l \\ 0 & \text{El día } i \text{ a la hora } j \text{ el grupo } k \text{ no inicia un bloque de la materia } l \end{cases}$$

Basado en las variables de decisión y las restricciones impuestas por el centro educativo, se plantea la siguiente función objetivo, la cual maximiza la cantidad de bloques que se puedan formar durante la semana.

$$\text{máx} \sum_{i \in D} \sum_{j \in H} \sum_{k \in G} \sum_{l \in M} B_{kl}^{ij}$$

donde H es el conjunto de las horas de la clase en cada día, D representa el conjunto de días de clase, G los grupos existentes en la institución y M las materias que deben ser dictadas. Se puede observar que este modelo es más sencillo que el planteado por Méndez et al. (2016) debido al tipo de institución en la cual se formuló el problema.

Este modelo sencillo se resolvió mediante el entorno *MOSEL* como lenguaje de modelación y programación; y cada instancia se resolvió con el Software FICO Xpress 7.8. Luego los valores de cada variable generados por el optimizador fueron procesados en el Software Microsoft Excel 2010, en el cual se utilizaron Macros para transformar los valores binarios de cada variable en los Días, Grupo, Hora y Materia correspondiente, obteniendo el horario final de cada una de las instancias.

Esquivel (2014) propone resolver el problema de programación horaria mediante un modelo de programación lineal. Uno de los objetivos principales es minimizar el tiempo de traslado de los profesores que imparten clases en diferentes sedes y además el horario deber ser lo más compacto posible.

Restricciones del modelo

En el modelo de la programación de horarios tiene una condición para los profesores especiales:

1. Las horas cátedra semanal que imparte el profesor especial en jornada extendida es una cantidad par. Lo anterior significa que en uno o varios días tendrán un horario especial, los cuales se distribuirán por cada 2 horas un día con dicho horario, por lo tanto, comienza su labor a la tercera hora y finaliza a la octava.

2. Las horas cátedra semanal que imparte el profesor especial en jornada extendida es una cantidad impar, distribuyéndose del total de días que tendrá dicho horario, en un día su labor empiece a la segunda hora y termine a la séptima, y el resto de los días desde la tercera hora hasta la octava.

Esquivel (2014) orienta las restricciones hacia la construcción del horario, en el entorno del sistema educativo público, teniendo en cuenta el Proyecto Educativo Institucional, los requerimientos y los diferentes traslados a la sede de algunos docentes:

1. Minimizar el traslado de los docentes compartidos entre las sedes.

2. Aquellas asignaturas de intensidad de 2 o más horas semanales, programarla en bloques de 2 horas.

Con respecto a los requerimientos del cuerpo de docentes

1. Programación de cada una de las asignaturas por grupo, en períodos consecutivos.

2. Límite de períodos de tiempo por día a programar de las asignaturas de intensidad horaria semanal de dos o más.

3. Programación de las asignaturas en bloque de dos períodos.

4. Períodos de traslado entre las sedes de los profesores compartidos.

5. Programación de los profesores compartidos en la sede no titular en períodos consecutivos.

6. Programación de los profesores con horario especial.

Con respecto a los requerimientos de la institución educativa

7. Programación en jornada normal.

8. Programación en jornada extendida.

9. Programación en jornada extendida en períodos consecutivos.

10. Programación del cuerpo de profesores.

11. Programación de las asignaturas que se imparten en un solo bloque a la semana.

12. Ocupación del cuerpo de profesores.

Con respecto a los requerimientos Legislativos

13. Límite de períodos de tiempo diarios que imparte el profesor.

14. Número de períodos de tiempo a programar en jornada extendida por profesor y sede.

La función objetivo se planteó en dos partes, en la primera parte se desea minimizar el total de traslados semanales de los profesores compartidos entre las sedes y en la segunda parte penalizar las asignaturas con intensidad horaria semanal de 2 o más horas, que dictan los profesores en su sede titular y se programan un período de tiempo en el día.

$$\text{mín}\left(\sum_{s\in SED}\sum_{p\in PRO[s]}\sum_{d\in D}\sum_{t\in T} BY_{psdt}\cdot p_1 + \sum_{g\in GRU}\sum_{a\in ASI[g]}\sum_{d\in D} BZ_{agd}\cdot p_2\right)$$

SED: Conjunto de sedes.
PRO: Conjunto de profesores.
$PRO[s]$: Conjunto de profesores compartidos en la sede s que no es la sede titular.
$PROCOM$: Conjunto de profesores $p \in PRO$ compartidos.
D: Conjunto de días.
T: Conjunto de períodos de tiempo.
BY_{psdt}: 1 si el profesor $p \in PROCOM$ imparte en la sede diferente a la titular $s \in SED$ el día $d \in D$ y arranca en el período de tiempo $t \in T$, 0 en otro caso.
GRU: Conjunto de grupos.
ASI: Conjunto de asignaturas a impartir.
$ASI[s]$: Conjunto de asignaturas que se imparten en el grupo g.
BZ_{agd}: 1 si la asignatura $a \in ASI$ de intensidad horaria semanal igual o mayor a dos y la imparte el profesor en su sede titular en el grupo $g \in GRU$ es programada el día $d \in D$ solo un período de tiempo, 0

en otro caso.

p_1: Penalización por el número de días a la semana que necesita el profesor compartido para impartir las diferentes asignaturas en la sede no titular.

p_2: Penalización a aquellas asignaturas de intensidad horaria igual o mayor a dos que la imparte el profesor en su sede titular y se programa solo un período de tiempo en el día.

2.4.2. Algoritmos genéticos

Ahumada (2014) resuelve el problema de programación horaria en INACAP, una institución de educación superior chilena, corporación de derecho privado, fundada el 21 de octubre de 1966 y constituida por tres instituciones: Centro de Formación Técnica, Instituto Profesional y Universidad Tecnológica de Chile.

Desea minimizar las "ventanas" para los docentes y "ventanas" en la planificación de cursos de un mismo nivel y carrera; además de asignar en primer lugar a los profesores con mayor prioridad y asignar en primer lugar las salas con mayor prioridad.

Construye una serie de matrices (secciones, salas, docente y horarios) para organizar y representar los datos de entrada. Luego organiza y representa las estructuras de datos para la validación en una matriz de combinación docente-horario: se inicia basándose en la disponibilidad horaria entregada por el docente y validada por el director de carrera; matriz de combinación docentes-secciones: se inicia basándose en la disponibilidad entregada por el docente y validada por el director de carrera, combinación secciones-salas: se inicia basándose en la cantidad de alumnos de una sección y el cupo de una sala además del tipo de sala que requiere una sección, lo cual es planificado por un director de carrera, disponibilidad horaria de salas: se inicia basándose en una planificación ya realizada restringiendo el uso de las salas que ya se encuentran utilizadas. También permite asignaciones parciales de horarios.

Suarez (2012) propone tres diferentes algoritmos genéticos para la generación de un horario:

- NSGA-II (Non-Dominated Sorting Genetic Algorithm-II): es un algoritmo de optimización multiobjetivo con elitismo.
- Algoritmo genético simple: no lo aplica a toda la solución en conjunto, sino que se utiliza para la solución individual de cada grupo, con el fin de que las variaciones sean más particulares.

- Algoritmo de búsqueda aleatoria: una técnica en la que la solución del problema se construye a través de la generación de posiciones indiscriminadas para la ubicación de los diferentes elementos a distribuir.

Justifica la aplicación de tres métodos de la siguiente manera:

> El uso de los diferentes métodos permite comparar la calidad de las soluciones obtenidas, examinando cuál de ellos ofrece un mejor nivel de optimización en sus resultados, de acuerdo con las restricciones impuestas (Suarez, 2012).

En el planteamiento de su problema considera las siguientes restricciones:

1. En un mismo día no se deben dictar más de 2 horas de la misma asignatura.
2. Un profesor no puede dictar 2 materias a la misma hora del día.
3. Dos asignaturas no pueden ser programadas a una misma hora de la jornada en el mismo salón o espacio físico.
4. Esta restricción contempla las asignaturas que de acuerdo con el tipo de actividades desarrolladas y al nivel de atención requerido por el estudiante, tienen una mejor disposición en la primera y segunda hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 3 y 4.
5. Esta restricción contempla las asignaturas que de acuerdo con el tipo de actividades desarrolladas y al nivel de atención requerido por el estudiante, tienen una mejor disposición en la tercera y cuarta hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 1 y 2.
6. Esta restricción contempla las asignaturas que de acuerdo con el tipo de actividades desarrolladas y al nivel de atención requerido por el estudiante, tienen una mejor disposición en la quinta y sexta hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 6, 7 y 8.

Finalmente, define su función objetivo como

$$\sum_{i=1}^{NG}\sum_{j=D}^{5} F_{Rest1}(Materias) + \sum_{j=1}^{5}\sum_{k=1}^{6} F_{Rest2}(Profesores) + F_{Rest2}(Salones)$$

Donde:

i: variable para recorrer la cantidad total de grupos a programar.

j: variable para recorrer los días de la semana.

k: variable para recorrer las horas del día.

NG: cantidad total de grupos a programar.

F_{Rest1}: 1 Si algún código se repite más de 2 veces en la columna j o si al repetirse algún código 2 veces en la columna j, estos no están consecutivos, 0 en otro caso.

F_{Rest2}: 1 Si se repite algún elemento en el vector de profundidad determinado por (j, k), 0 en otro caso.

Rodríguez (2012) trabaja en la programación horaria del Departamento de Ciencias Matemáticas, Universidad de Puerto Rico, Recinto de Mayagüez. Representa un horario factible mediante el uso de tres matrices: S representa la asignación de un salón a una clase, P representa la asignación del profesor a la clase y $HProf$ es una matriz que guarda la información sobre la carga, cursos, horario y salones que le han sido asignados a cada profesor y las valoraciones de cada uno de los elementos asignados. Cabe resaltar que las matrices S y P son matrices superpuestas, es decir, si $S_{ij} \neq 0$, entonces $P_{ij} \neq 0$ y si $S_{ij} = 0$, entonces $P_{ij} = 0$.

Luego de generar su primer horario factible (población inicial), aplica tres operadores genéticos con el fin de obtener nuevas soluciones que cumplan todas sus restricciones. Dichos operadores utilizarán información externa al horario para mejorar la solución, por ejemplo: la preferencia del profesor por un curso, horario y salón. A continuación, se describen los tres operadores genéticos que son aplicados a las matrices S y P:

1. Recombinación de genes: consiste simplemente en seleccionar dos filas de la matriz S (P) e intercambiarlas. Ese operador únicamente actúa sobre la asignación de un profesor a un curso.

2. Recombinación y mutación de genes: consiste en intercambiar la información entre dos entradas de la matriz S (P) de una misma fila. El operador actúa principalmente sobre la asignación de un profesor a un bloque de horario, sin embargo, afecta la asignación del salón con el fin de lograr el intercambio de horario sin necesidad de perder la factibilidad.

3. Mutación: Consiste en asignar al profesor un salón de su preferencia, siempre y cuando no afecte la factibilidad de la solución.

De acuerdo con el planteamiento de su problema, consideró las siguientes restricciones:

1. Un profesor sólo puede estar asignado, a lo más, a un evento por bloque.

2. Dado un salón y un bloque de horario, a lo más, puede haber un evento programado.

3. Un evento de tres horas se programará, a lo más, en un bloque de horario.

4. Un evento de cuatro o cinco horas se programará, a lo más, en dos bloques de horario.

5. No debe haber cursos asignados durante el receso académico.

6. Todo profesor debe tener su carga completa.

7. Debe cumplirse de manera forzosa toda preasignación.

8. Un evento sólo puede ser programado en salones que cumplan con sus requerimientos.

9. No se programarán clases después de las 5:00 pm.

El algoritmo fue probado utilizando un horario realizado manualmente por el Departamento de Ciencias Matemáticas, se obtuvo una coincidencia del 70 % en cuanto a la asignación de las clases. En este algoritmo no considera la cantidad de créditos máxima y mínima, que pueden ser asignados a cada profesor, por lo que este aspecto y otros se deben mejorar con la programación manual.

2.4.3. Búsqueda tabú

Costabel (2005) toma como punto de partida el trabajo de grado sobre Tabú Search "Asignación de Salones y Horarios", elaborado Cuevas, Panzera y Ugalde (2003).

Define la variable de decisión X_{csh}: 1 si la clase c es programada en el salón s y la hora h, $c \in C, s \in S, h \in H$, 0 en otro caso. Luego C: Conjunto de clases, H: Conjunto de horas, S: Conjunto de salones.

Para evaluar una solución obtenida, la función objetivo tiene en cuenta si se cumplen o no las restricciones y las preferencias definidas en la formalización del Problema de Asignación de Salones y Horarios a Asignaturas (PASHA). Así las restricciones impuestas por Costabel (2005) son:

R1 Cada clase tiene una duración determinada.

R2 Una clase tiene un único salón.

R3 Un horario y salón se programan a una única clase.

R4 Las clases se dictan dentro de los horarios válidos.

R5 Todas las clases son programadas.

R6 El horario de cada clase es contiguo.

R7 Cada clase se dicta en el mismo día.

R8 Las clases de un grupo deben estar en el turno especificado.

R9 Las clases de un grupo deben dictarse en diferentes días.

para las preferencias:

P1 Superposición: las clases de una asignatura básica u obligatoria con un solo grupo no se superponen.

P2 Preferencia de horario: todas las clases se programan dentro del horario de preferencia especificado.

P3 Capacidad del salón: la capacidad del salón es suficiente para la clase.

P4 Recursos necesarios: el salón de una clase cuenta con los recursos necesario para su ofrecimiento.

P5 Contigüidad horaria: los teóricos y prácticos de una asignatura se dictan en horarios contiguos.

P6 Desperdicio de salón: no se utilizan salones de gran capacidad para clases con pocos alumnos.

P7 Espaciamiento en días: las clases de un grupo teórico se distribuyen en forma uniforme en la semana. Para evaluar el espaciamiento entre las clases se mide la distancia en días entre la primera y la última en la semana. Dependiendo de la cantidad de clases por semana existe un mínimo exigible para esa distancia:

- 1 clase por semana: no aplicable.
- 2 clases por semana: distancia ≥ 2 (o sea al menos un día libre entre ambas clases).
- 3 clases por semana: distancia ≥ 3.
- 4 clases por semana: distancia ≥ 4.
- 5 clases por semana: distancia ≥ 5.
- 6 clases por semana: no aplicable.

P8 Similitud de horario: las clases de un grupo teórico comienzan en “horario similar” durante la semana. Se considera “horario similar” para las clases de un grupo teórico si el horario de cada una de ellas coincide en por lo menos media hora. No se aplica si la Preferencia de Horario para cada una de esas clases es disjunta.

Define la función objetivo como:

$$f = \sum_{i=1}^{9} pesoR_i \cdot costoR_i + \sum_{i=1}^{8} pesoP_i \cdot costoP_i$$

La investigación de Costabel (2005) logró mejorar el algoritmo del cálculo de la función objetivo para hacerlo más eficiente en tiempo de ejecución, con respecto al trabajo realizado en el proyecto de grado sobre Tabú Search en el Problema de Asignación de Salones y Horarios, por Cuevas, Panzera y Ugalde (2003).

2.4.4. Colonia de hormigas

El algoritmo de colonia de hormigas trata de imitar el comportamiento de las hormigas a la hora de buscar alimento, el cual, se hace mediante una comunicación indirecta entre los individuos a partir de rastros de feromonas.

Gore (2017) genera una programación horaria utilizando el algoritmo colonia de hormigas. Define el problema como $TP = (T, L, R, S, C)$, donde T son los intervalos de tiempo para la programación, L son los recursos limitados de la universidad, R es el conjunto de salones, debido a que algunas salones no pueden estar disponibles durante un intervalo de tiempo, define el conjunto de intervalos donde el salón esté disponible como: $T_r \subseteq T$, E es el conjunto de eventos que deben ser programados, S conjunto de estudiantes que tomarán algún evento, C conjunto de restricciones. Asume que todos los eventos son múltiplos de un valor fijo de tiempo al cual llama "time–quantum", así define intervalo de tiempo como: uno o más "time–quantum" consecutivos en el horario. El problema principal al aplicar el algoritmo colonia de hormigas a un problema es encontrar una representación adecuada que pueda ser utilizada por las hormigas artificiales para construir la solución, en consecuencia Gore (2017) define los salones como el conjunto de lugares de trabajo, entonces para cada salón $r \in R$ el número de lugares de trabajo es definido como $size_r \in N$.

Un nodo de estudiante solo puede estar conectado a un "dock", si cumple las siguientes restricciones:

1. Un "dock" está conectado al menos a uno de los eventos a los que el estudiante necesita asistir.
2. Un estudiante no tiene pre-asignaciones en "time–quantum" representado por "dock" y el dur_{es} "time–quantum" consecutivos. El evento e_s es el evento más corto inscrito por el estudiante que se puede mantener en el "dock" mencionado anteriormente, y su duración se denota por dur_{es}.

Para elegir el camino que toma la hormiga k, que se encuentra en el nodo i y pase al nodo j, lo calcula utilizando la regla de proporcionalidad aleatoria.

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{ij}^{\alpha}][\eta_{ij}]^{\beta}}, j \in \mathcal{N}_i^k \tag{2.4.1}$$

Donde τ_{ij} es el rastro de la feromona que conecta el nodo i con el nodo j, η_{ij} valor heurístico del nodo (es importante tener en cuenta que el valor heurístico se modifica dinámicamente a lo largo de la ejecución del algoritmo), α y β son los parámetros que determinan la influencia relativa del rastro de feromonas y la información heurística y $\mathcal{N}_i^k$ es el vecindario factible de hormiga k cuando está en el nodo i.

La actualización de la feromona se da al final de cada iteración del algoritmo por la mejor hormiga. La probabilidad de que la mejor hormiga permita actualizar el rastro de la feromona es de 5 %, a diferencia de otros trabajos realizados con el algoritmo colonia de hormigas, Gore (2017) no mantiene el valor de la feromona. Luego, define el nivel de influencia entre dos eventos, la cual está modelada por la función de influencia $f : E \times E \to \{0, 1\}$. Cuando $f_{a,b} = 0$ significa que el evento a no tiene influencia sobre el evento b, sin embargo si $f_{a,b} = 1$ significa que el evento a está programado sobre el conjunto de "dock" adecuados para el evento b. Formalmente la función de influencia se define como:

$$f_{a,b} = \frac{|chosenD_a \cap D_b|}{|D_b|} \tag{2.4.2}$$

Donde $chosenD_a \subseteq D_a$ es un conjunto de "docks" donde el evento a está programado en una solución propuesta y $D_b \subseteq D$ es el conjunto de "docks" adecuados para el evento b.

Utiliza como variable a optimizar el número de obligaciones no programadas. Define una obligación como la asignación de un estudiante determinado a uno de los ejercicios de laboratorio que necesita para asistir. La función de calidad de la solución Q_e para el evento e y la sugerencia de la solución Sug:

$$Q_e = \frac{assignedObligations(e, Sug)}{numberOfObligations(e)} \cdot spaceEfficiency(e, Sug)^3 \tag{2.4.3}$$

donde

$$spaceEfficiency(e, Sug) = \left[\frac{assignedOblogations(e, Sug)}{reservedSeats(e, Sug)}\right] \tag{2.4.4}$$

La ganancia de la feromona para cada evento $\Delta\tau_e$ es dada por:

$$\Delta\tau_e = \left(\frac{\sum_{e_i \in E} (f_{e,e_i} \cdot Q_{e_i})}{|E|} \right)^4, e \in E \tag{2.4.5}$$

La ganancia de feromonas $\Delta\tau_e$ se deposita en los bordes del recorrido que conecta el evento e con el "dock" d y en los bordes que conectan a un estudiante con cualquiera de los "dock" en los que está programado el evento.

Capítulo 3

Modelo general y algoritmo

3.1. Descripción del modelo general

En esta investigación se diseñó un modelo general el cual se resolvió usando una metaheurística (algoritmos genéticos) con el fin de dar solución al problema de programación horaria.

El método de los algoritmos genéticos conjuga rapidez y eficiencia en su trabajo, produciendo resultados aceptables; se considera una técnica muy conveniente para resolver problemas de asignación horaria (Guerra et al., 2013).

El objetivo principal en la elaboración de un horario académico es asignar a cada evento un profesor disponible y un salón con los requisitos necesarios donde pueda impartir el evento.

Para comprender el modelo es importante tener claro los siguientes términos:

- Bloque: se refiere a un elemento del conjunto de períodos de tiempo en los cuales se puede programar un curso de tres horas.
- Curso: cursos que siguen un programa, especialmente dentro de un plan de estudios. Ejemplo: secuencia de cursos en inglés, español, matemáticas, etc.
- Evento: se refiere a una sección de un curso específico. Ejemplo: MATE 1-001, MATE 1-002, MATE 1-003; son tres eventos distintos, pero todos del mismo curso.
- Pre-asignación: es un evento que el director decide asignarle a un profesor específico.

Con el fin de dar solución al problema de programación de horarios académicos se definen los siguientes conjuntos y variables que se utilizarán en el modelo.

Conjunto de cursos $C = \{C_1, C_2, \cdots, C_n\}$.
Conjunto de eventos del curso $C_j = \{e_{1j}, e_{2j}, \cdots, e_{m_j j}\}$, con $1 \leq j \leq n$ y $m_j = |C_j|$ cardinalidad del conjunto C_j.
El evento i del curso j, se denota como e_{ij}, con $1 \leq i \leq m_j$ y $1 \leq j \leq n$.

Conjunto de eventos preasignados $E_p \subseteq E$.
Conjunto de eventos de tres horas o menos $E_3 \subseteq E$.
Conjunto de eventos de cuatro horas o más $E_4 \subseteq E$.
Conjunto de profesores $P = \{p_1, p_2, \cdots, p_l\}$.
Conjunto de salones $S = \{s_1, s_2, \cdots, s_r\}$.
C_{ms}: Capacidad mínima de estudiantes en el salón s.
C_{Ms}: Capacidad máxima de estudiantes en el salón s.
Conjunto de bloques horarios $B = \{b_1, b_2, \cdots, b_t\}$.
C_{ij}: Cantidad de créditos asignados al evento e_{ij}.
CR_{mp}: Mínimo de créditos a ser asignados al profesor p.
CR_{ap}: Créditos asignados por el algoritmo al profesor p.
CR_{Mp}: Máximo de créditos a ser asignados al profesor p.
CE_{ij}: Cantidad de estudiantes en el evento e_{ij}.
CC_j: Total de estudiantes matriculados en el curso j.
w_i: Pesos que se pueden ajustar para dar mayor o menor importancia a sus cantidades asociadas. Donde $i \in \{1, 2, 3\}$.
$Pre\_curso_{pj}, Pre\_bloque_{pb}$: Preferencia del profesor p por el curso j y bloque b, respectivamente.

Variables de decisión

$$x_{e_{ij}psb} = \begin{cases} 1 & \text{si el evento } e_{ij} \text{ está asignado al profesor } p \text{ en el salón } s \text{ durante el bloque } b \\ 0 & \text{en otro caso} \end{cases}$$

El modelo general que representa la solución del problema de programación de horarios académicos está dado por:

Restricciones

1. Un profesor sólo puede estar asignado, a lo más, a un evento por bloque.

$$\sum_{j=1}^{|C|}\sum_{i=1}^{|C_j|}\sum_{s\in S} x_{e_{ij}psb} \leq 1, \forall p, b \tag{3.1.1}$$

2. Dado un salón y un bloque de horario, a lo más, puede haber un curso programado.

$$\sum_{j=1}^{|C|}\sum_{i=1}^{|C_j|}\sum_{p\in P} x_{e_{ij}psb} \leq 1, \forall s, b \tag{3.1.2}$$

3. Debe cumplirse toda pre-asignación.

$$\sum_{e_{ij}\in E_p}\sum_{p\in P}\sum_{s\in S}\sum_{b\in B} x_{e_{ij}psb} = |E_p| \tag{3.1.3}$$

4. Un evento menor o igual de tres horas se programará, a lo más, en un bloque de horario.

$$\sum_{p\in P}\sum_{s\in S}\sum_{b\in B} x_{e_{ij}psb} \leq 1, \forall e_{ij} \in E_3 \tag{3.1.4}$$

5. Un evento de cuatro horas o más se programará, a lo más, en dos o tres bloques de horario.

$$\sum_{p\in P}\sum_{s\in S}\sum_{b\in B} x_{e_{ij}psb} \leq 3, \forall e_{ij} \in E_4 \tag{3.1.5}$$

6. La cantidad de créditos asignados al profesor p debe estar acotado inferior y superiormente, por la cantidad mínima y máxima de créditos solicitados.

$$CR_{pm} \leq \sum_{j=1}^{|C|}\sum_{i=1}^{|C_j|}\sum_{s\in S}\sum_{b\in B} x_{e_{ij}psb} \cdot C_{ij} \leq CR_{pM}, \forall p \tag{3.1.6}$$

7. La suma de estudiantes por eventos del mismo curso debe ser igual al total de estudiantes del curso.

$$\sum_{i=1}^{|C_j|} CE_{ij} = CC_j, \forall j \tag{3.1.7}$$

8. La capacidad del salón s debe ser razonable para el evento asignado.

$$C_{ms} \leq CE_{ij} x_{e_{ij}psb} \leq C_{Ms}, \forall i, j, p, s, b \tag{3.1.8}$$

9. Si un curso tiene más de un evento a programar, se evitará asignar eventos del mismo curso en un mismo bloque horario.

$$\sum_{i=1}^{|C_j|} \sum_{p \in P} \sum_{s \in S} x_{e_{ij}psb} \leq 1, \forall b, j \tag{3.1.9}$$

Con el propósito de que cada profesor tenga la misma posibilidad de satisfacer sus preferencias se establecen las siguientes restricciones:

$$\sum_{j=1}^{|C|} Pre\_curso_{pj} = 1, \forall p \tag{3.1.10}$$

$$\sum_{b \in B} Pre\_bloque_{pb} = 1, \forall p \tag{3.1.11}$$

La función objetivo está dada por:

$$\text{máx}\left(PrefP - ESP_{jb}\right), \forall j, b \tag{3.1.12}$$

donde

$PrefP = \sum_{j=1}^{|C|} \sum_{i=1}^{|C_j|} \sum_{p \in P} \sum_{s \in S} \sum_{b \in B} (w_1 Pre\_curso_{pj} + w_2 Pre\_bloque_{pb}) x_{e_{ij}psb}$ los pesos w_1 y w_2 se pueden modificar para dar mayor o menor importancia a las preferencias de los profesores.

$ESP_{jb} = w_3 \sum_{i=1}^{|C_j|} \sum_{p \in P} \sum_{s \in S} x_{e_{ij}psb}$ representa la cantidad total de eventos superpuestos del curso j en el bloque b.

La función objetivo 3.1.12 maximiza la preferencia de los profesores respecto a los cursos y bloques y es penalizada cada vez que exista un evento superpuesto del mismo curso.

3.2. Algoritmo

Silva (2017) define algoritmo como: la secuencia finita de operaciones que se realizan sin ambigüedades y que produce en un tiempo finito una solución a un problema matemático.

Para la elaboración de nuestra metaheurística nos basamos en algunas estrategias planteadas por Rodríguez (2012), como por ejemplo el utilizar un único padre para generar un único hijo y el operador de recombinación de genes, el cual concluimos que era efectivo para la evolución de nuestra solución. Algunas diferencias importantes entre el presente trabajo y el realizado por Rodríguez (2012) son: se agregaron más restricciones al modelo matemático, se tomaron en cuenta parámetros de gran relevancia para generar una solución más acorde a la vida real, es decir, se considera los créditos asignados al profesor (carga mínima

y máxima de créditos) y eventos, la capacidad y disponibilidad de salones y para finalizar se elaboró una interfaz amigable.

El algoritmo desarrollado en esta investigación está compuesto por tres etapas. La primera etapa está vinculada a la interacción del usuario con la interfaz, la cual consiste en el ingreso de la información, la segunda etapa consiste en la construcción de la solución y la última etapa es la presentación de resultados (Figura 3.1).

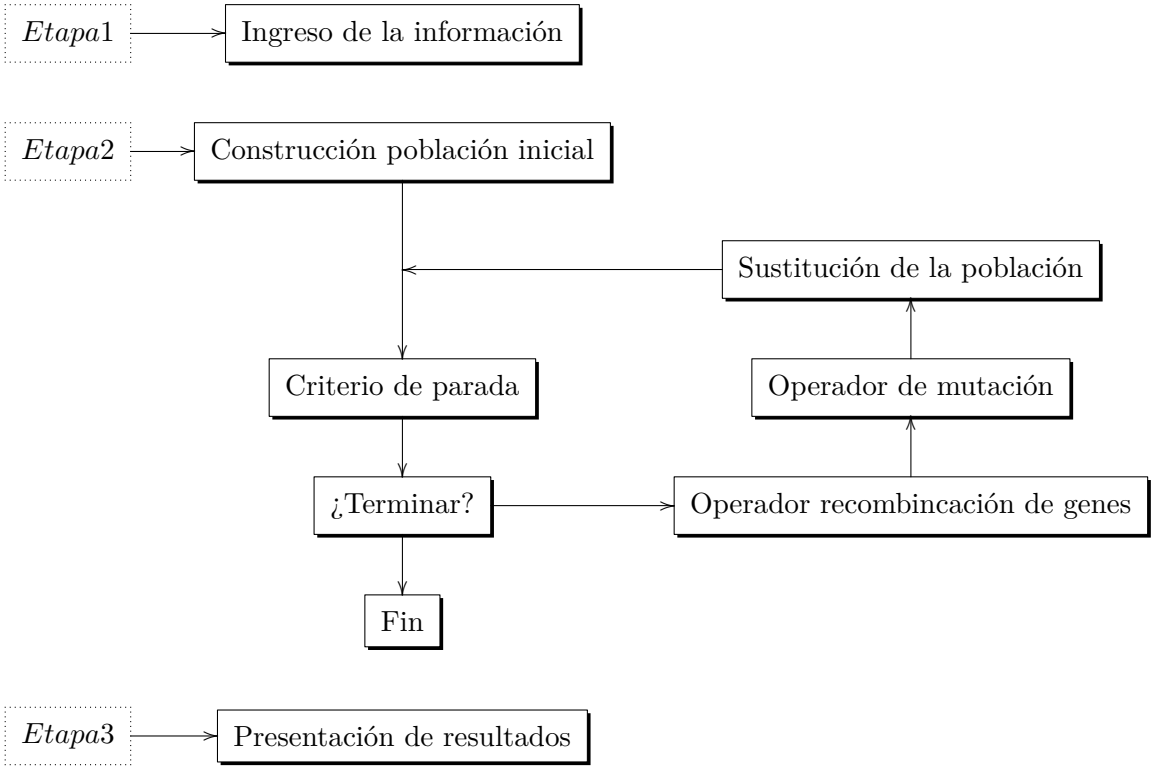


Figura 3.1: Estructura general del algoritmo

Como se mencionó anteriormente la técnica que se aplicó para hallar una solución factible al problema de programación horaria es conocida con el nombre de algoritmos genéticos. Con el fin de aplicar dicha técnica es necesario definir un conjunto de elementos tales como población inicial, individuo, operadores genéticos, entre otros.

A continuación, se describen los elementos fundamentales del algoritmo para la construcción de la población inicial y aplicación de este.

3.2.1. Matrices de asignación de salones y profesores

Para hallar una solución al problema de asignación horaria será necesario utilizar matrices que contengan la información necesaria. Las matrices $PP_{ET\times 6b_t}$ y $SS_{ET\times(6b_t+2)}$, contienen información de la asignación del horario, donde $ET = |E|$ es la cantidad de eventos[1] y b_t es la cantidad de bloques horarios. Las filas de las matrices SS y PP representan la ubicación del evento e_{ij}; por ejemplo, la fila 23 de la matriz SS (o PP) puede representar al evento e_{48}, el cual corresponde al evento 4 del curso 8. Los bloques horarios están representados por seis columnas, cada una de las celdas de matriz SS representa media hora, así si el bloque b_i horario corresponde a los días lunes, miércoles y viernes (LWV), las primeras dos celdas ($6b_i - 5$ y $6b_i - 4$) corresponden al día lunes, las siguientes dos celdas ($6b_i - 3$ y $6b_i - 2$) al día miércoles y las últimas dos celdas ($6b_i - 1$ y $6b_i$) al día viernes. Esta identificación de columnas sigue hasta el último bloque horario de los días LWV. Si el bloque horario b_i corresponde a los días martes y jueves (MJ), las primeras tres celdas corresponden al día martes ($6b_i - 5$, $6b_i - 4$ y $6b_i - 3$) y las últimas tres celdas ($6b_i - 2$, $6b_i - 1$ y $6b_i$) corresponden al día jueves, similarmente para la matriz PP. Finalmente, las columnas $6b_t + 1$ y $6b_t + 2$ representan la cantidad de estudiantes y créditos del evento, respectivamente.

La entrada de la matriz SS representa la asignación de cada evento en un bloque horario y un salón junto a la cantidad de estudiantes y créditos asignados al evento. Si la entrada de la matriz $SS_{k\times 6b_*} = s$ ($b \leq b_t$ y $s \neq 0$), significa que el evento e_{ij} ubicado en la fila k está asignado en el bloque b_* en el salón s, si $SS_{k\times(6b_t+1)} = CE_{ij}$ ($CE_{ij} \neq 0$) el evento e_{ij} ubicado en la fila k tiene una capacidad de CE_{ij} estudiantes y $SS_{k\times(6b_t+2)} = C_{ij}$ ($C_{ij} \neq 0$) el evento e_{ij} ubicado en la fila k tiene C_{ij} créditos asignados. De manera similar la matriz PP establece el profesor en el bloque horario y evento.

$$
SS = \begin{array}{c} \\ e_{11} \\ e_{12} \\ e_{22} \\ \vdots \\ e_{ETC_n} \end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} 6b_1-5 & 6b_1-4 & 6b_1-3 & 6b_1-2 & 6b_1-1 & 6b_1 & 6b_2-5 & \cdots & 6b_t & CE & C \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 0 & \cdots & 0 & 30 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & * & 0 & 0 \\
4 & 4 & 4 & 4 & 4 & 4 & * & \cdots & * & 94 & 7 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & * & * \\
* & * & * & * & * & * & * & \cdots & * & * & *
\end{pmatrix}
\end{array}
$$

Figura 3.2: Matriz de salones

Por ejemplo, en la Figura 3.2, el evento e_{11} (evento 1 del curso 1, ubicado en la fila 1) tiene una capacidad de 30 estudiante y 3 créditos que fue asignado durante el bloque horario b_1 en el salón 1, en

[1]Nota: los eventos están ordenados por curso y de manera descendente según su capacidad de estudiantes.

cambio el evento e_{22} (evento 2 del curso 2, ubicado en la fila 3) tiene una capacidad de 40 estudiantes y 4 créditos que fue asignado durante el bloque horario b_1 en el salón 4.

$$PP = \begin{array}{c} \\ e_{11} \\ e_{12} \\ e_{22} \\ \vdots \\ e_{ETC_n} \end{array} \begin{array}{c} \begin{array}{ccccccccc} 6b_1-5 & 6b_1-4 & 6b_1-3 & 6b_1-2 & 6b_1-1 & 6b_1 & 6b_2-5 & \cdots & 6b_t \end{array} \\ \begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & \cdots & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & * & * & * & * & \cdots & * \end{pmatrix} \end{array}$$

Figura 3.3: Matriz de profesores

Por ejemplo, en la Figura 3.3, el evento e_{11} (evento 1 del curso 1, ubicado en la fila 1) se le asignó el profesor 3 en el bloque horario b_1 y el evento e_{22} (evento 2 del curso 2, ubicado en la fila 3) se le asignó el profesor 5 en el bloque horario b_1.

Observe que si la entrada $SS_{k\times 6w} \neq 0$, donde $w \in \{6b_*-5, 6b_*-4, 6b_*-3, 6b_*-2, 6b_*-1, 6b_*\}$ entonces $PP_{k\times 6w} \neq 0$ y si $SS_{k\times 6w} = 0$ entonces $PP_{k\times 6w} = 0$. Así podemos ver la asignación de un horario factible como las matrices SS y PP superpuestas a partir de la columna 1 hasta la columna $6b_t$.

3.2.2. Matriz Cursos-Eventos

La forma en que se ingresan los datos iniciales es de gran importancia para los algoritmos genéticos, es por este motivo que se utilizarán las matrices superpuestas A y B. Donde las filas de la matriz A y B representan los cursos que se deben programar. Las columnas de la matriz A representan la división del curso en eventos, según la capacidad de estudiantes en cada evento y las columnas de la matriz B indican esa capacidad máxima de estudiantes en cada evento (Figura 3.4).

$$A = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 4 & 1 & 2 & 1 \\ 3 & 0 & 0 & 0 \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix} \end{array} \qquad B = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 30 & 80 & 60 & 100 \\ 50 & 0 & 0 & 0 \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix} \end{array}$$

Figura 3.4: Matiz Cursos-Eventos

Por ejemplo, el curso C_1 tiene ocho eventos distribuidos de la siguiente manera: cuatro eventos con capacidad de treinta estudiantes, un evento con capacidad de ochenta estudiantes, dos eventos con capacidad de sesenta estudiantes y un evento con capacidad de cien estudiantes. En cuanto al curso C_2 solo tiene tres eventos con capacidad de cincuenta estudiantes.

Con el propósito de manejar una data más ordenada y de fácil acceso, el algoritmo se encargará de ordenar las filas de las matrices SS y PP de forma ascendente según la cantidad de horas contacto del curso, luego los eventos de cada curso los ordenará de manera descendente, según la capacidad de estudiantes en cada evento. A continuación, se describen otras matrices importantes que servirán para manejar los datos con mayor facilidad.

Primero, se define $CC_{C_n \times 4}$, que es una matriz que indica el orden de los eventos del curso C_j de forma descendente, según la capacidad máxima de estudiantes por evento en la matriz SS (PP). La columna uno indica la fila de inicio en matriz SS (PP) de los eventos del curso j, la columna dos indica la última fila de la matriz SS (PP) donde se encuentran los eventos del curso j, la columna tres indica la cantidad de eventos del curso C_j y la columna cuatro indica las horas contacto profesor-estudiante del curso. En la Figura 3.5 se presenta un ejemplo de la matriz CC, la información proporcionada se lee de la siguiente manera: el curso C_1 de tres horas cuenta con ocho eventos que se encuentran ubicados de la fila 1 a 8 de la matriz SS (PP), el evento del curso C_2 de tres horas, está ubicado en la fila 9 de la matriz SS (PP), así sucesivamente.

$$CC = \begin{pmatrix} 1 & 8 & 8 & 3 \\ 9 & 9 & 1 & 3 \\ 10 & 15 & 6 & 3 \\ 16 & 20 & 5 & 4 \end{pmatrix}$$

Figura 3.5: Matriz CC

3.2.3. Representación de preferencias de profesores

Para representar la preferencias de los profesores será necesario el uso de dos matrices, Pre_curso de tamaño $p_l \times (C_n + m)$. Si la entrada de la matriz $Pre\_curso_{ij} = k_1$ $(k_1 \neq 0)$, significa que el profesor i tiene un preferencia k por el curso j y las columnas $C_n + 1$ a $C_n + m$ indican la preferencia por la cantidad de estudiantes en el evento, donde 0: indica el deseo de evento pequeño, 1: indica el deseo por megasecciones, es decir, eventos que necesiten salones con una capacidad mayor a 93 estudiantes, 2: indica

el deseo por eventos grandes y 3: en otro caso, es decir existe una relación entre la preferencia del curso y la cantidad de estudiantes en los eventos. Finalmente, las columnas $C_n + m + 1$ y $C_n + m + 2$ indican la cantidad mínima y máxima de créditos que pueden ser asignados al profesor. La cantidad de créditos está relacionada con la petición del profesor por clases grandes, megasecciones o preferencia por créditos adicionales (compensaciones).

La matriz *Pre_bloque* de tamaño $p_l \times b_t$. Si $Pre\_bloque_{ij} = k_2$ ($k_2 \in \{0, 1\}$), si $k_2 = 1$ significa que el profesor i tiene preferencia por el bloque j, en otro caso $k_2 = 0$. (Figura 3.6)

$$Pre\_curso = \begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_l \end{array} \begin{array}{c} \begin{array}{cccccccccc} C_1 & C_2 & \cdots & C_n & C_n+1 & \cdots & C_n+m & C_n+m+1 & C_n+m+2 \end{array} \\ \begin{pmatrix} 0.7 & 0 & \cdots & 0.3 & 2 & \cdots & 0 & 12 & 12 \\ * & * & \cdots & 0 & * & \cdots & * & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & * & \cdots & * & * & \cdots & * & * & * \end{pmatrix} \end{array}$$

$$Pre\_bloque = \begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_l \end{array} \begin{array}{c} \begin{array}{cccc} b_1 & b_2 & \cdots & b_t \end{array} \\ \begin{pmatrix} 1 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix} \end{array}$$

Figura 3.6: Representación de preferencias de profesores

El profesor p_1 tiene una preferencia de 0.7, 0 y 0.3, por el curso C_1, C_2 y C_n, respectivamente. Para los cursos C_1 y C_n desea eventos grandes y pequeños, respectivamente. Además, tiene una carga mínima y máxima igual a 12 créditos. También, el profesor p_1 tiene una preferencia por los bloques horario b_1 y b_2, pero no tiene ninguna preferencia por el bloque horario b_t.

3.2.4. Disponibilidad de salones

Se conoce la disponibilidad horaria de cada salón. La información se presenta en la matriz *Dis_salon* de tamaño $s_r \times 6b_t$ donde las filas representan los salones y las columnas los bloques horarios de tres horas. Los bloques horarios están representados por seis columnas, cada una de las celdas de matriz *Dis_salon* representa media hora, así si el bloque horario b_i corresponde a los días lunes, miércoles y viernes (LWV), las primeras dos celdas ($6b_i - 5$ y $6b_i - 4$) corresponden al día lunes, las siguientes dos celdas ($6b_i - 3$ y $6b_i - 2$) al día miércoles y las últimas dos celdas ($6b_i - 1$ y $6b_i$) al día viernes, esta identificación de

columnas se sigue hasta el último bloque horario de los días LWV. Si el bloque horario b_i corresponde a los días martes y jueves (MJ), las primeras tres celdas corresponden al día martes ($6b_i - 5$, $6b_i - 4$ y $6b_i - 3$) y las últimas tres celdas ($6b_i - 2$, $6b_i - 1$ y $6b_i$) corresponden al día jueves. Si el salón se encuentra disponible para dictar clases la entrada de la matriz *Dis_salon* es igual a cero, de lo contrario la entrada será igual a 1.

$$
Dis\_salon = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_r \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 6b_1-5 & 6b_1-4 & 6b_1-3 & 6b_1-2 & 6b_1-1 & 6b_1 & 6b_2-5 & \cdots & 6b_t \end{array} \\
\left(\begin{array}{ccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & * & \cdots & * \\
0 & 0 & 0 & 0 & 0 & 0 & * & \cdots & * \\
* & * & * & * & * & * & * & \cdots & * \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 1 & 1 & 0 & 0 & * & \cdots & *
\end{array}\right)
\end{array}
$$

Figura 3.7: Disponibilidad de salones

A manera de ejemplo presentamos la Figura 3.7, supongamos que el bloque b_1 está relacionado con el bloque de horario de los días LWV de 730 a 830. De esta manera podemos interpretar la información que nos proporciona la matriz *Dis_salon* de la siguiente manera: el salón s_1 no está disponible en el bloque de horario b_1, al contrario, s_2 se puede utilizar para dictar clases en todo el bloque b_1. Finalmente, el salón s_r está disponible los lunes y viernes durante el bloque b_1.

3.2.5. Construcción de la población inicial

Según Suarez (2011) los métodos "metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos, combinando diferentes conceptos de diversos campos como la genética, la biología, la inteligencia artificial, las matemáticas, la física y la neurología, entre otras"(p. 49). En nuestra investigación el marco general es la metaheurística de algoritmos genético. Una diferencia importante a la estructura general de un algoritmo genético con la presente investigación es que utilizamos un único padre para generar un único hijo, donde las matrices PP y SS representan el padre (posibles solución del problema) y las filas de estas matrices representan los genes.

Una vez se ingresados los datos iniciales el siguiente paso es la construcción de la solución inicial la cual debe cumplir con las restricciones planteadas en la Sección 3.1 pág 25. Debido a la variabilidad de relación evento-crédito primero se asignaron los eventos a profesores que tengan igual carga mínima y máxima. Como segunda fase se cumplirá con la carga mínima de créditos de cada profesor. En la tercera fase se

asignan compensaciones (créditos extras) a los profesores con tal preferencia de trabajo.

1. Se selecciona el conjunto de profesores cuya carga mínima es igual a la carga máxima, llámese a este conjunto $P1$
2. Mientras $P1 \neq \emptyset$ selecciona aleatoriamente un profesor $p \in P1$
3. Se asignan eventos dentro de los cursos preferencia del profesor p
4. Termina asignación de profesores en el conjunto $P1$
5. Mientras $P \setminus P1 \neq \emptyset$ selecciona aleatoriamente un profesor $p \in P \setminus P1$
6. Termina asignación de profesores en el conjunto $P \setminus P1 \neq \emptyset$
7. Si hay eventos no asignados
8. Asignar compensaciones a profesores $p \in P \setminus P1$
9. Retorna SS_0 y PP_0, $Asig\_p_0$

Figura 3.8: Pseudocódigo de la población inicial

El objetivo de las técnicas metaheurísticas es obtener nuevas soluciones factibles a bajo costo computacional, razón por la cual el algoritmo implementará dos operadores genéticos con el fin de mejorar la solución inicial, operador recombinación de genes y operador de mutación. Estos operadores tomarán en cuenta la información relacionada con la preferencia de los profesores por los cursos y bloques de horarios, lo cual permitirá tener una evolución consecutiva de la población.

Luego de construir la población inicial y solución final, el algoritmo tendrá dos fases adicionales; la primera fase será útil en el caso que la demanda de eventos sea superior a la oferta. Es en este punto donde los profesores regulares han completado su carga académica, razón por la cual no podrían dictar más eventos. Descrita la situación anterior el algoritmo se encargará de asignar los eventos restantes a un profesor *fantasma*,[2] con el fin de asignar todos los eventos de la programación horaria. La segunda fase del algoritmo consiste en asignar los laboratorios o talleres a un salón y bloque horario. La asignación de laboratorios y eventos sin asignar se realiza al terminar la asignación a los profesores regulares, ya que los encargados de dictar talleres o eventos sin asignar son los estudiantes graduados o profesores con contratos temporeros, los cuales cuentan con un horario más flexible. A continuación, se describen los operadores de recombinación de genes y mutación.

[2]Generalmente las instituciones de educación superior cuentan con profesores por contrato o estudiantes graduados para cumplir con la demanda que no pueden cubrir los profesores regulares.

3.2.6. Operador de recombinación de genes

Consiste en seleccionar dos filas de la matriz PP y SS (matrices superpuestas) e intercambiar la información guardada sobre la asignación del evento, así se obtiene una nueva solución del problema. Ciertamente son muchas las posibles combinaciones que podemos realizar, pero solo algunas representarán una mejoría a solución. Por este motivo, procedemos a aplicar el operador cruce de la siguiente manera: se escoge una fila (un gen, cada gen representa un evento) aleatoriamente y se analiza la información que nos provee acerca del profesor asignado a dicho evento. Luego se utilizan las preferencias del profesor para escoger la otra fila (gen) y proceder a realizar el intercambio de las filas. Note que este operador interviene sobre dos filas (genes) de las matrices PP y SS (Figura 3.9).

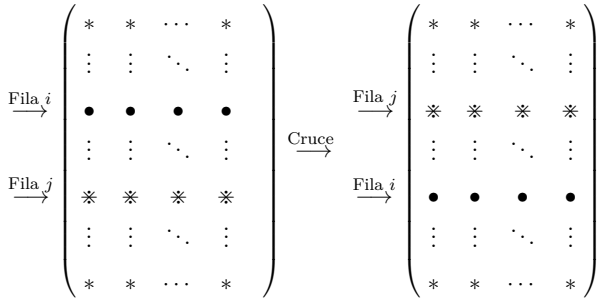


Figura 3.9: Operador de cruce del modelo general

Este operador interviene principalmente sobre la asignación de un profesor a un evento de su preferencia, sin embargo también afectar la carga mínima y máxima de créditos asignados. Para evitar soluciones no factibles causadas por el intercambio de filas, este operador utilizará la información aportada por la institución en cuanto a la cantidad mínima y máxima de créditos que pueden ser asignados al profesor. La construcción de la solución está basada en el Pseudocódigo presentado en la Figura 3.10.

3.2.7. Operador de Mutación

El operador de mutación escogido consiste en seleccionar un evento aleatoriamente, se analiza la información de la matriz PP para determinar el profesor asignado al evento y tratar de asignarlo a un bloque de horario de preferencia del profesor. Al igual que en el operador de recombinación de genes no tiene sentido intercambiar los bloques de horarios sin considerar si esto mejora o afecta la calidad de la solución obtenida. Por lo tanto, para proceder al intercambio de los bloques se utilizará la información del profesor respecto a la preferencia de bloques de horarios. Note que el operador de mutación solo está modificando

1. Se selecciona aleatoriamente un evento $e_{i?}$ y se analiza la información que nos provee acerca del profesor p_1 a este evento.

2. Se busca el curso al pertenece el evento seleccionado, $e_{ij} \in C_j$

3. Selecciona aleatoriamente un curso C_k dentro de las preferencias del profesor p_1 y selecciona un evento $e_{lk} \in C_k$

4. Determine el profesor asignado al evento e_{lk}, si el evento no tiene profesor asignado ir al paso 9

5. Si el evento e_{lk} tiene profesor p_2 asignado

6. Si la preferencia p_1 por el curso C_k es mayor que la preferencia de p_2 entonces intercambie las filas de las matrices PP_0 y SS_0.

7. Si $CR_{mp_1} \leq CR_{ap_1} \leq CR_{Mp_1}$ y $CR_{mp_2} \leq CR_{ap_2} \leq CR_{Mp_2}$ entonces actualice información de $Asig\_p_1$. Si no, regrese a la asignación original.

8. Si la preferencia p_1 por el curso C_k es mayor que la preferencia por el curso C_j entonces intercambie las filas de las matrices PP_0 y SS_0.

9. Si $CR_{mp_1} \leq CR_{ap_1} \leq CR_{Mp_1}$ entonces actualice información de $Asig\_p_1$. Si no, regrese a la asignación original.

10. Retorna $Asig\_p_1$, SS_1 y SS_1

Figura 3.10: Pseudocódigo operador de recombinación de genes

una fila (gen) (Figura 3.11).

Este operador interviene principalmente sobre la asignación de un profesor a un bloque o bloques de horarios de su preferencia y disponibilidad de salones adecuados para dictar el evento en el bloque(s) de horario. Al igual, que el caso anterior este operador podría generar soluciones no factibles, por lo cual el operador utilizará la información relacionada con la disponibilidad y la capacidad máxima de estudiantes por salones. Además, verificar que exista un patrón de horario disponible para asignar el evento.

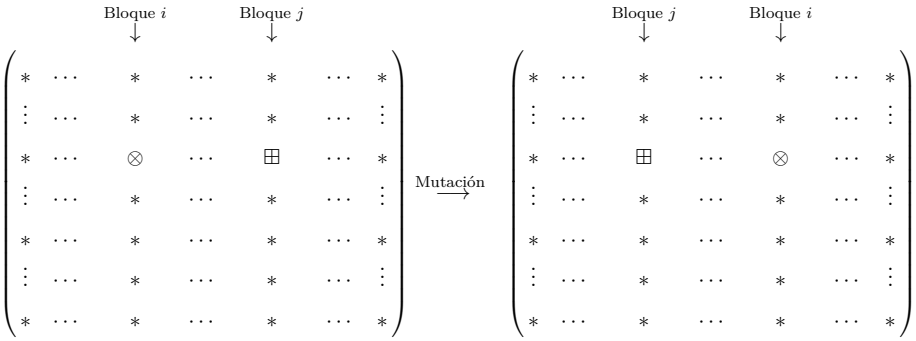


Figura 3.11: Operador de mutación del modelo general

1. Se selecciona aleatoriamente un evento e_{ij} (fila)

2. Determinar el profesor y bloque donde está asignado al evento e_{ij}

3. Si el bloque o bloques están en el conjunto de preferencias del profesor, no aplica operador de mutación

4. En caso contrario, buscar disponibilidad bloque o bloques de horario en el conjunto de preferencias del profesor y salón adecuado para el evento e_{ij}

5. Si existe bloque (s) y salón disponible entonces actualiza SS_1 y PP_1, $Asig\_p_1$

6. Retorna SS_1 y PP_1, $Asig\_p_1$

Figura 3.12: Pseudocódigo operador de mutación

La construcción de la solución está basada en el Pseudocódigo presentado en la Figura 3.13.

1. Ingreso de matrices Pre_bloque, Pre_curso, A y B.

2. Genera la matriz CC y ordena matrices A y B.

3. Genera matrices básicas SS_0, PP_0 y $Asig\_p_0$ (matrices de ceros).

4. Genera primera solución factible, 'horario0' $SS_0 = SS_1$ y $PP_0 = PP_1$, $Asig\_p_0 = Asig\_p_1$

5. Condición de parada.

6. Operador de recombinación de genes

7. Operador mutación

8. Genera horario1

9. Si horario1 es mejor que 'horario0' entonces horario0=horario1 y $SS_0 = SS_1$ y $PP_0 = PP_1$, $Asig\_p_0 = Asig\_p_1$.

10. Retornar 'horario1' y SS_1 y PP_1, $Asig\_p_1$

Figura 3.13: Pseudocódigo del algoritmo

3.2.8. Representación final de un horario factible

Con el fin de ordenar y obtener la información de asignación horaria con mayor facilidad, se introduce una nueva matriz a la cual llamaremos $Asig\_p$ de tamaño $(ET + 1) \times 6(p_l + 1)$. La matriz guarda la información de los eventos, bloques horarios, patrón de horario y salones asignados a cada profesor. Las filas representan los eventos y la entrada $Asig\_p_{(ET+1)(6p_i-5)}$ con $i \in \{1, \ldots, p_l+1\}_{\mathbb{N}}$ representa la cantidad de créditos asignados por el algoritmo. Las columnas $6p_i - 5$ a $6p_i - 3$ contienen los bloques horarios donde se asignó el evento, $6p_i - 2$ el patrón de horario y $6p_i - 1$ a $6p_i$ los salones donde se asignó el evento al

profesor p_i. Note que la asignación de un salón cuenta con dos columnas, debido a que algunos eventos pueden necesitar dos salones diferentes para ser dictados, por ejemplo: lunes, miércoles en salón regular y día viernes en salón de laboratorio.

$$
Asig\_p = \begin{array}{c} \\ e_{11} \\ e_{22} \\ e_{**} \\ \vdots \\ e_{ETC_n} \\ e_{(ET+1)C_n} \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 6p_1-5 & 6p_1-4 & 6p_1-3 & 6p_1-2 & 6p_1-1 & 6p_1 & 6p_2-5 & \cdots & 6p_l \end{array} \\
\begin{pmatrix}
1 & 2 & 0 & 3 & 2 & 2 & 0 & \cdots & 0 \\
3 & 0 & 0 & 2 & 1 & 1 & 0 & \cdots & 0 \\
* & * & * & * & * & * & * & \cdots & * \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
* & * & * & * & * & * & * & \cdots & * \\
12 & 0 & 0 & 0 & 0 & 0 & * & \cdots & *
\end{pmatrix}
\end{array}
$$

Figura 3.14: Representación final de un horario factible

En la Figura (3.14) el profesor p_1 se le asignó el evento e_{11} (evento 1 del curso 1) en los bloques de horarios 1 y 2 en el patrón horario 3 en el salón 2. Además, se le asignó el evento e_{22} en el bloque horario 3 en el patrón 2 en el salón 1. Finalmente, se le asignaron 12 créditos al profesor p_1.

La información obtenida en la programación horaria es complicada de analizar rápidamente por cualquier usuario. Por tal razón se elaboró una interfaz gráfica utilizando Matlab Guide 2015 que ayudará a manejar la información de una manera más sencilla (ver Apéndice B).

Capítulo 4

Caso de estudio y resultados

Actualmente el Departamento de Ciencias Matemáticas de la Universidad de Puerto Rico, Recinto de Mayagüez, realiza la programación horaria de su ofrecimiento de forma manual, tarea que normalmente requiere de varios días de trabajo, puesto que durante el proceso de programación horaria puede ocurrir que se programen una gran cantidad de eventos superpuestos, profesores que no cumplen su carga mínima, entre otras repercusiones que se pueden generar con un horario planificado manualmente. Por lo tanto el encargado de construir el horario debe revisar constantemente el cumplimiento de estas restricciones para construir la versión final del horario.

El modelo general desarrollado en el presente trabajo trata de responder a esta necesidad, generando una asignación lo más cercana a una solución ideal. Su eficiencia será comparada con los datos de la programación horaria realizada manualmente en el primer semestre del año académico 2018-2019 en el Departamento de Ciencias Matemáticas. El objetivo es proporcionar una solución matemática rápida y eficaz, donde el horario generado por el algoritmo se acerque a la realidad sin descartar la posibilidad de realizar cambios manuales.

A continuación, se describe el caso de estudio y luego se muestran los resultados obtenidos por el algoritmo.

4.1. Programas de estudios

El Departamento de Ciencias Matemáticas dicta una gran diversidad de cursos subgraduados para sus programas académicos, como para para otros programas del Recinto Universitario de Mayagüez. Ofrece tres programas subgraduados, cinco programas de maestría y un programa doctoral conjunto con el Departamento de Ingeniería Eléctrica y Computadoras (Tabla 4.1).

Programas de bachillerato	**Programas de maestría**	**Programas doctorales**
Matemática Pura	Matemática Pura	Ciencias e Ingeniería de la Información (CISE)
Ciencias de la Computación	Matemáticas Aplicadas	
Educación Matemática	Estadística	
	Computación Científica	
	Educación Matemática	

Tabla 4.1: Programas ofrecidos por el Departamento de Ciencias Matemáticas

4.2. Conjuntos de bloques de horarios

En el Departamento de Ciencias Matemáticas, se dictan las clases de lunes a viernes, distribuyendo la jornada de la siguiente manera: lunes (L), miércoles (W) y viernes(V), las clases inician desde las 7:30 hasta las 18:30; los martes y jueves hay un receso académico entre las 10:30 y las 12:30 llamado hora universal, así las clases deben iniciar desde las 7:30 hasta las 10:30, luego desde las 12:30 hasta 18:30. Una vez mencionados los bloques horarios, podemos estudiar las diferentes maneras de programar un evento, las cuales llamaremos patrones de horarios. La idea de definir los bloques horarios es garantizar que los eventos inicien y finalicen a la misma hora, los días donde éste fue asignado. Es conveniente definir los siguientes subconjuntos de bloques de horario: el subconjunto B_{LWV} corresponde a los bloques horarios de los lunes, miércoles y vienes, B_{MJ} corresponde a los bloques horarios de los martes y jueves, dichos bloques horarios se consideran de tres horas (Tabla 4.2).

Lunes-Miércoles-Viernes	Bloque
7:30 a 8:30	Bloque 1
8:30 a 9:30	Bloque 2
9:30 a 10:30	Bloque 3
10:30 a 11:30	Bloque 4
11:30 a 12:30	Bloque 5
12:30 a 13:30	Bloque 6
13:30 a 14:30	Bloque 7
14:30 a 15:30	Bloque 8
15:30 a 16:30	Bloque 9
16:30 a 17:30	Bloque 10
17:30 a 18:30	Bloque 11

Martes-Jueves	Bloque
7:30 a 9:00	Bloque 12
9:00 a 10:30	Bloque 13
Hora universal	
12:30 a 14:00	Bloque 14
14:00 a 15:30	Bloque 15
15:30 a 17:00	Bloque 16
17:00 a 18:30	Bloque 17

Tabla 4.2: Distribución y numeración de bloques horarios

4.3. Cursos con características especiales

La mayoría de los cursos se ajustan al modelo de bloques horarios de tres horas de la Tabla 4.2; sin embargo, existen un grupo de cursos con características especiales, por ejemplo, la cantidad de horas contacto es mayor, la necesidad de un salón especial o salones distintos para ser dictados (Tabla 4.3).

Curso	Número de Horas	Condiciones del salón
MATE3005 y MATE4145	5	
MATE3031 y MATE3032	4	
COMP3057	4	2h salón regular y 2h laboratorio
COMP3010	4	2h salón regular y 2h laboratorio
COMP3110	4	2h salón regular y 2h laboratorio
ESMA3016	3	2h laboratorio

Tabla 4.3: Cursos con características especiales

4.4. Patrones de horarios

Con el fin de agilizar y facilitar el modelo matemático se definirán patrones de horarios que deben seguir los eventos a programar, donde un curso de tres o menos horas se debe programar en un bloque de horario. Un curso de cuatro o más horas se debe programar en dos bloques de horarios, excepto en los patrones de horarios de LMWV o LWJV que en ciertos casos se utilizarán tres bloques de horarios. Para los eventos de tres, cuatro y cinco horas se cuentan con dos, doce y tres patrones de horario, respectivamente. Debemos resaltar que para los eventos de cuatro horas existen dos patrones que necesitan tres bloques de horarios para ser programados, esto como consecuencia de la hora inicial de los eventos, es decir si un evento es programado de 8:30 a 9: 30 los lunes, miércoles y viernes, entonces el martes (jueves) el evento deberá comenzar a las 8:30 y terminar a las 9:30. Por esta razón se necesitarán tres bloques para asignar los eventos que sigan estos patrones de horario (ver Apéndice B).

La suma total de horas de dos y tres bloques de horarios corresponde a seis y nueve horas respectivamente, por lo tanto, cuando se programen los eventos de cuatro horas en dichos patrones deben sobrar horas (espacios). Dichas horas se utilizarán para la programación de eventos de una y dos horas. Cabe resaltar que en los eventos de dos horas se deben programar las dos horas de forma consecutiva.

Los eventos se programarán conforme a la necesidad de cada profesor. Como se mencionó anteriormente, primero se asignarán los eventos a profesores cuya carga mínima es igual a la carga máxima, ya que es lo más difícil de resolver debido a la variabilidad de relación evento-crédito, luego se continúa con el resto de los profesores. El horario generado por el algoritmo tratará de satisfacer las preferencias de los profesores sobre cursos y bloques de horarios, además de cumplir con la carga mínima de créditos que deben ser asignados.

4.5. Resultados

A continuación, se muestran los resultados obtenidos con el modelo general propuesto en el caso de estudio, descripción de matrices y parámetros utilizados. Una vez explicado el caso de estudio y evaluado las condiciones reales, se procedió a aplicar el modelo propuesto con el fin de obtener la mejor asignación horaria para el Departamento de Ciencias Matemáticas. El horario generado puede ameritar cambios, ya que existen situaciones en la vida real que no se pueden incluir en el modelo matemático o resultaría muy tedioso para el usuario introducir dicha información, la cual puede ser modificada con mayor facilidad luego de obtener el horario generado por nuestro modelo como punto de partida.

4.5.1. Datos de entrada

Para los efectos de prueba se tomaron en consideración los datos reales de primer semestre año académico 2018-2019 del Departamento de Ciencias Matemáticas de la Universidad de Puerto Rico, Recinto de Mayagüez, uno de los departamentos que ofrece la mayor cantidad de secciones de cursos que son requisito para los estudiantes del Recinto.

Los datos están constituidos por un curso dos horas, treinta y siete de tres horas, seis de cuatro horas y un curso de cinco horas, para un total de 45 cursos que se traducen a 132 eventos y 37 profesores (ver Apéndice C) y 20 salones que se detallan en la Tabla 4.4.

Capacidad de estudiantes	**Salones**
115-128	s_1
95-114	s_2 s_3
81-94	s_4
60-80	s_5
46-59	s_6 s_7 s_8
31-45	s_9 s_{10} s_{11}
1-30	s_{12} s_{13} s_{14} s_{15} s_{16} s_{17} s_{18} s_{19} s_{20}

Tabla 4.4: Distribución de salones según su capacidad

Es importante destacar que los cursos MATE3171 y MATE3172, cuentan con una hora extra (labo-

ratorio) en grupos de 15 estudiantes que son dictados por estudiantes graduados, por este motivo, no se consideran como una prioridad de asignación. La hora extra se programará una vez se haya finalizado la programación de todos los cursos que ofrece el departamento.

En nuestro caso de estudio el total de estudiantes en el curso MATE3171 y MATE3172, es igual a 1260 y 407, respectivamente. De esta manera se deben programar 84 secciones de 15 para el curso MATE3171, para el curso MATE3172 se deben programar 25 secciones de 15 y 2 secciones de 16 estudiantes. Debido a la actual falta de salones en el Departamento de Ciencias Matemáticas programarán los laboratorios en secciones (eventos) de treinta estudiantes.

Los eventos que se desean programar tienen una frecuencia de clases semanales que va desde dos a cuatro días. Los profesores tienen una restricción relacionada con la cantidad mínima y máxima de créditos que deben ser asignados, van desde 3 hasta 22 como máximo.

Para la construcción del caso de prueba, se construyó una asignación basada en las preferencias de cursos, bloques de horarios de cada profesor según la asignación del primer semestre 2018-2019. Durante la entrada de datos iniciales al algoritmo, se decidió usar el siguiente sistema de preferencia 1: 1 si el curso debe ser asignado al profesor como una preasignación, entre 0.9 y 0.4 representa una preferencia alta y menor a 0.4 una preferencia baja. Las preferencias de cada profesor, carga mínima-máxima se pueden observar con más detalle en el apéndice D.

Es importante mencionar que el horario del primer semestre 2018-2019 es la versión final realizada por el Departamento de Ciencias Matemáticas, por tal razón se asignaron las preferencias de cursos a cada profesor de acuerdo con su especialidad, con el fin de reducir el número de posibles asignaciones fuera del área de interés del profesor. Para construir la matriz Dis_salon se obtuvo información proporcionada por el departamento de Ciencias Matemáticas en relación con la disponibilidad de cada salón para el primer semestre 2018-2019.

4.6. Restricciones y función objetivo en el caso de estudio

Conociendo los datos de entrada del caso de estudio, a continuación, se definen los conjuntos y restricciones que modelan la programación de horarios en el Departamento de Ciencias Matemáticas.

Conjuntos y variables

Conjunto de cursos $C = \{C_1, C_2, \cdots, C_{45}\}$.

Conjunto de eventos del curso $C_j = \{e_{1j}, e_{2j}, \cdots, e_{m_j j}\}$, con $1 \leq j \leq 45$ y $m_j = |C_j|$ cardinalidad del conjunto C_j.

El evento i del curso j, se denota como e_{ij}, con $1 \leq i \leq m_j$ y $1 \leq j \leq 45$.

Conjunto de eventos de tres horas o menos $E_3 = \{C_1, C_2, \cdots, C_{38}\}$

Conjunto de eventos de cuatro horas o más $E_4 = \{C_{39}, C_{40}, \cdots, C_{45}\}$

Conjunto de profesores $P = \{p_1, p_2, \cdots, p_{37}\}$.

Conjunto de salones $S = \{s_1, s_2, \cdots, s_{20}\}$.

C_{ms}: Capacidad mínima de estudiantes en el salón s.

C_{Ms}: Capacidad máxima de estudiantes en el salón s.

Conjunto de bloques horarios $B = \{b_1, b_2, \cdots, b_{17}\}$.

C_{ij}: Cantidad de créditos asignados al evento e_{ij}.

CR_{mp}: Mínimo de créditos a ser asignados al profesor p.

CR_{ap}: Créditos asignados por el modelo al profesor p.

CR_{Mp}: Máximo de créditos a ser asignados al profesor p.

CE_{ij}: Cantidad de estudiante en el evento e_{ij}.

CC_j: Total de estudiantes matriculados en el curso j.

w_i: Pesos que se pueden ajustar para dar mayor o menor importancia a sus cantidades asociadas. Donde $i \in \{1, 2, 3\}$.

$Pre\_curso_{pj}, Pre\_bloque_{pb}$: Preferencia del profesor p por el curso j y bloque b, respectivamente.

Variable decisión

$$x_{e_{ij}psb} = \begin{cases} 1 & \text{si el evento } e_{ij} \text{ está asignado al profesor } p \text{ en el salón } s \text{ durante el bloque } b \\ 0 & \text{en otro caso} \end{cases}$$

El modelo general que representa la solución del problema de programación de horarios académicos está dado por:

Restricciones

1. Un profesor sólo puede estar asignado, a lo más, a un evento por bloque.

$$\sum_{j=1}^{45}\sum_{i=1}^{m_j}\sum_{s=1}^{20} x_{e_{ij}psb} \leq 1, \forall p, b \tag{4.6.1}$$

2. Dado un salón y un bloque de horario, a lo más, puede haber un curso programado.

$$\sum_{j=1}^{45}\sum_{i=1}^{m_j}\sum_{p=1}^{37} x_{e_{ij}psb} \leq 1, \forall s, b \tag{4.6.2}$$

3. Un evento menor o igual de tres horas se programará, a lo más, en un bloque de horario.

$$\sum_{p=1}^{37}\sum_{s=1}^{20}\sum_{b=1}^{17} x_{e_{ij}psb} \leq 1, \forall e_{ij} \in E_3 \tag{4.6.3}$$

4. Un evento de cuatro horas o más se programará, a lo más, en dos o tres bloques de horario.

$$\sum_{p=1}^{37}\sum_{s=1}^{20}\sum_{b=1}^{17} x_{e_{ij}psb} \leq 3, \forall e_{ij} \in E_4 \tag{4.6.4}$$

5. La cantidad de créditos asignados al profesor p debe estar acotado inferior y superiormente, por la cantidad mínima y máxima de créditos solicitados por el profesor.

$$CR_{pm} \leq \sum_{j=1}^{45}\sum_{i=1}^{m_j}\sum_{s=1}^{20}\sum_{b=1}^{17} x_{e_{ij}psb} \cdot C_{ij} \leq CR_{pM}, \forall p \tag{4.6.5}$$

6. La suma de estudiantes por eventos del mismo curso debe ser igual al total de estudiantes del curso.

$$\sum_{i=1}^{m_j} CE_{ij} = CC_j, \forall j \tag{4.6.6}$$

7. La capacidad del salón s debe ser razonable para el evento asignado.

$$C_{ms} \leq CE_{ij} \leq C_{Ms}, \forall i, j, s \tag{4.6.7}$$

8. Si un curso tiene más de un evento a programar, se evitará asignar eventos del mismo curso en un mismo bloque horario.

$$\sum_{i=1}^{m_j}\sum_{p=1}^{37}\sum_{s=1}^{20} x_{e_{ij}psb} \leq 1, \forall b, j \tag{4.6.8}$$

La función objetivo está dada por:

$$\max\left(PrefP - ESP_{jb}\right), \forall j, b \tag{4.6.9}$$

donde

$PrefP = \sum_{j=1}^{45}\sum_{i=1}^{m_j}\sum_{p=1}^{37}\sum_{s=1}^{20}\sum_{b=1}^{17}(w_1 Pre\_curso_{pj} + w_2 Pre\_bloque_{pb})x_{e_{ijpsb}}$, en el caso de estudio los pesos w_1 y w_2 se trabajarán con valores constantes.

$ESP_{jb} = w_3 \sum_{i=1}^{m_j}\sum_{p=1}^{37}\sum_{s=1}^{20} x_{e_{ijpsb}}$ representa la cantidad total de eventos superpuestos del curso j en el bloque b. En el caso de estudio el peso w_3 se trabajará con un valor constante.

La función objetivo 4.6.9 maximiza la preferencia de los profesores hacia los cursos y bloques. Además, disminuye la cantidad de eventos superpuestos del mismo curso. El número de variables y restricciones presentes en el caso de estudio es 566100 y 1905, respectivamente.

4.6.1. Puntos importantes del algoritmo

Definimos la efectividad del algoritmo comparando la coincidencia de los cursos y horas de preferencia del profesor y comparando los valores de la función objetivo del modelo versus y el resultado final del horario final del Departamento de Ciencias Matemáticas. Además, debe cumplir con la carga mínima de créditos que se deben ser asignados a cada profesor. También se considera la asignación de eventos en salones idóneos[1], es decir, evitar el desperdicio de espacio en los salones. De esta manera vamos a comparar el horario generado por el algoritmo con el horario del primer semestre 2018-2019 producido por el Departamento de Ciencias Matemáticas. Dicha comparación es importante, ya que los métodos heurísticos no cuentan con métodos para medir la calidad de sus soluciones, entonces es posible hacernos una idea de la efectividad de la solución al problema de asignación de horario.

Sistema operativo	Windows 10
Procesador	Intel Core i5 (1.60 GHz)
Memoria	8GB de Ram
Disco duro	256GB de SSD

Tabla 4.5: Características del equipo

[1]De acuerdo con capacidad mínima y máxima de estudiantes, en el salón establecida por el Departamento de Ciencias Matemáticas.

Para medir el tiempo computacional, todas las pruebas fueron realizadas en un mismo equipo, el cual cuenta con las características descritas en la Tabla 4.5. Se implementó el algoritmo, usando como condición de: a) parada el número de iteraciones máximo y b) el número de iteraciones sin que mejore el valor de la función objetivo. Se concluyó que no existe relación con el alto número de iteraciones realizadas y una mejoría en el valor de la función objetivo, por lo tanto, consideramos apropiado utilizar como condición de parada mil iteraciones.

Además, se calculó valor de la función objetivo (VF) asociada al horario final del primer semestre 2018-2019 del Departamento de Ciencias Matemáticas. Los pesos constantes y valor de la función objetivo en los casos de prueba se presentan en la Tabla 4.6. Al hacer una comparación entre el valor de la función objetivo del modelo propuesto y el del horario final del Departamento de Ciencias Matemáticas, se observa que el horario manual da una mejor solución que el propuesto, una razón puede ser que no está considerando pesos variables para la decisión de los profesores en otros aspectos importantes, como especialidad en sus áreas, años de servicio, etc.

Pesos constantes			Valor promedio	
w_1	w_2	w_3	de la función objetivo	VF
1	1	1	128.92	133
0.8	0.2	1	77.06	83.12
0.2	0.8	1	27.87	32.03

Tabla 4.6: Pesos constantes y valor de función objetivo

Para evitar soluciones no factibles durante el proceso del algoritmo se decide aplicar primero el operador recombinación de genes y luego el de mutación.

Debido a la complejidad con la relación evento-cantidad de créditos, el algoritmo primero asigna los eventos a los profesores que tengan como carga mínima igual a la carga máxima de créditos. En el caso que no exista una combinación de eventos que logre asignar la carga mínima igual a la máxima al profesor, se procede a relajar la restricción de carga máxima por encima de 3 créditos en relación con su carga máxima inicial, si aun así no se logra completar la carga el algoritmo le indicará al usuario que debe cambiar la partición de cursos. Luego, el algoritmo procede a completar la carga mínima a los profesores restantes. Finalmente, si existen eventos no asignados se procede conceder compensaciones a los profesores con dicha preferencia y asignar las secciones sin profesor a profesores fantasmas.

4.6.2. Eventos no asignados

Anteriormente se mencionó que los profesores tienen una carga mínima y máxima relacionada con la cantidad de créditos. En este sentido, ningún profesor puede sobrepasar su carga máxima excepto en casos donde la demanda sea alta, así el encargado de elaborar el horario se ve en la obligación de exceder la carga máxima a lo más por tres créditos al profesor, siempre y cuando exista un acuerdo entre ambas partes de exceder su carga máxima y debe ser menor a la cantidad de créditos máxima permitida por la universidad. En particular el caso de estudio cuenta con un total de 132 eventos lo cual se traduce a 536 créditos, según la Tabla E1 (Apéndice E), pero se cuenta con una carga máxima de profesores de 459.5 créditos. Evidentemente, los profesores regulares no podrán cubrir está demanda, por tal razón el algoritmo se verá en la obligación de generar un profesor extra: profesor *fantasma* con el fin de terminar la asignación de eventos-créditos. Es importante destacar que el profesor *fantasma* no tendrá restricción alguna de créditos ni preferencia de curso y bloques de horarios, ya que el objetivo principal es asignar los eventos no asignados a un salón y bloque de horario; generalmente esta sobrecarga la cubren los estudiantes graduados o profesores por contrato que no tienen derecho a escoger su horario. Así se lograrán asignar todos los eventos.

4.6.3. Análisis de la solución

A continuación, se muestran algunos resultados de asignaciones obtenidos por algoritmo genético y la eficiencia de este.

Se compara la solución del horario generado por nuestro algoritmo con el horario del primer semestre 2018-2019, del Departamento de Ciencias Matemáticas, es decir, la coincidencia de los eventos asignados al profesor y hora de asignación. Además, se evitará el desperdicio de espacio. La Tabla 4.7 muestra en negrita la primera coincidencia horaria del algoritmo respecto con el primer criterio de eficiencia.

La coincidencia respecto a los eventos asignados con el horario final del Departamento de Ciencias Matemáticas es de 88.5 % y la relación evento-hora de asignación es de 19 %, ya que el algoritmo no considera la preferencia del profesor de asignarlo a un evento en una hora específica. No existió desperdicio de salón, es decir, no se asignaron clases pequeñas a salones de gran capacidad.

Como segunda coincidencia se considera la preferencia del profesor por cursos, es decir, asignar eventos dentro del área de interés del profesor. y al menos un evento con preferencia alta. En relación con al bloque

Profesor	Evento	Cantidad de estudiantes	Horario	Días	Salón
	MATE3172	94	13:30 a 14:30	**L-1h, W-1h, V-1h**	**s3**
P1	**MATE3005**	59	15:30 a 17:30	**L-2h, W-2h, V-1h**	**s6**
	$CR_{mp} = 12$	$CR_{ap} = 13.5$	$CR_{Mp} = 14$		
	MATE3020	30	**7:30 a 8:30**	**L-1h, W-1h, V-1h**	**s19**
	MATE3172	**94**	**8:30 a 9:30**	**L-1h, W-1h, V-1h**	**s3**
P2	**MATE3031**	**123**	16:00 a 18:00	M-2h, J-2h	**s1**
	$CR_{mp} = 12$	$CR_{ap} = 19$	$CR_{Mp} = 21.5$		
	MATE3021	**111**	9:00 a 10:30	**M-1:30h, J-1:30h**	**s2**
P3	**MATE4009**	**30**	7:30 a 9:00	**M-1:30h, J-1:30h**	**s12**
	$CR_{mp} = 10$	$CR_{ap} = 10$	$CR_{Mp} = 10$		

Tabla 4.7: Muestra de la solución, coincidencia 1

horario de preferencia, se considera que los eventos estén asignados dentro de las horas solicitadas por el profesor. La Tabla 4.8 muestra en negrita la primera coincidencia horaria del algoritmo respecto con el segundo criterio de eficiencia.

Profesor	Evento	Cantidad de estudiantes	Horario	Días	Salón
	MATE3172	**94**	**13:30 a 14:30**	**L-1h, W-1h, V-1h**	**s3**
P1	**MATE3005**	**59**	**15:30 a 17:30**	**L-2h, W-2h, V-1h**	**s6**
	$CR_{mp} = 12$	$CR_{ap} = 13.5$	$CR_{Mp} = 14$		
	MATE3020	**30**	**7:30 a 8:30**	**L-1h, W-1h, V-1h**	**s19**
	MATE3172	**94**	**8:30 a 9:30**	**L-1h, W-1h, V-1h**	**s3**
P2	**MATE3031**	**123**	16:00 a 18:00	M-2h, J-2h	**s1**
	$CR_{mp} = 12$	$CR_{ap} = 19$	$CR_{Mp} = 21.5$		
	MATE3021	**111**	**9:00 a 10:30**	**M-1:30h, J-1:30h**	**s2**
P3	**MATE4009**	**30**	**7:30 a 9:00**	**M-1:30h, J-1:30h**	**s12**
	$CR_{mp} = 10$	$CR_{ap} = 10$	$CR_{Mp} = 10$		

Tabla 4.8: Muestra de la solución, coincidencia 2

La eficiencia respecto a eventos asignados con el área de interés del profesor es de 100 %, cabe resaltar que solo a un profesor de los treinta y siete no se asignó un evento con preferencia alta, definiendo

preferencia alta por encima de 0.6. La coincidencia respecto a los eventos asignados al profesor en su área de interés es de 100 % y respecto a las horas solicitadas es de 90 %.

4.7. Costo de ofrecimiento de los cursos

La función objetivo 4.6.9 maximiza las preferencia del profesor respecto al bloque horario y curso de su preferencia, pero nos preguntamos: ¿qué costo tiene el ofrecimiento de los cursos en el Departamento de Ciencias Matemáticas considerando el salario del profesor y las asignaciones a éste? Para responder a esta pregunta se asumieron los siguientes salarios o costos de compensaciones o contrataciones:

1. Si el profesor tiene carga completa se le pagarán $36000
2. Se pagará $681 por cada crédito de compensación recibido.
3. Se pagará $1300 por cada crédito asignado al profesor *fantasma*.

Bajo los pagos a los profesores mencionados anteriormente y las restricciones de la página 4.6 se planteó la siguiente función objetivo:

$$\text{mín}(681CR_c + 1300CR_f + PrefPcosto) \tag{4.7.1}$$

donde

$$PrefPcosto = \sum_{j=1}^{45}\sum_{i=1}^{m_j}\sum_{p=1}^{37}\sum_{s=1}^{20}\sum_{b=1}^{17} 681(Pre\_curso_{pj} + Pre\_bloque_{pb})x_{e_{ij}psb}.$$

$PrefPcosto$: Penalidad por no asignar un evento a un profesor regular.

CR_c: Total de créditos asignados como compensación adicional a los profesores.

CR_f: Total de créditos asignados a los profesores *fantasmas*.

La función objetivo minimiza el costo de generar un horario académico al Departamento de Ciencias Matemáticas. Luego de obtener el valor de la función objetivo debemos sumarle la constante de $36000|P|$ para obtener el costo total de generar un horario, porque estamos trabajando bajo las restricciones de la pág 4.6, donde todo profesor debe cumplir con su carga mínima de trabajo. Además, a cada profesor se le aumento su carga máxima por tres créditos, con el fin de disminuir la cantidad de créditos totales asignados al profesor *fantasma*.

La Tabla 4.10 muestra un ejemplo de horario obtenido bajo el criterio de minimizar el costo de este. Se observa que al profesor P1 se le asignó un evento adicional en comparación con la Tabla 4.8, por lo tanto

Condición de parada	iteraciones	tiempo(seg)	Mejor valor de la funcion objetivo
a)	1000	150±3	362790
a)	10000	240±3	349440
b)	1000	60±3	349980
b)	5000	91±3	349440

Tabla 4.9: Valor de la función objetivo costo

se necesitan cuatro bloques de horario para programar los tres eventos, pero el profesor P1 solamente tres bloques de horario como preferencia, por tal razón no existe una coincidencia horaria en el curso MATE3005. Al profesor P2 se le asignó el evento MATE3171 con 127 estudiantes. En cuanto al profesor P3 su asignación no cambio.

Profesor	Evento	Cantidad de estudiantes	Horario	Días	Salón
	MATE3172	**94**	**14:30 a 15:30**	**L-1h, W-1h, V-1h**	**s3**
P1	**MATE3005**	**59**	15:30 a 17:30	**L-2h, W-2h, V-1h**	**s6**
	MATE3171	**34**	**13:30 a 14:30**	**L-2h, W-2h, V-1h**	**s11**
	$CR_{mp} = 12$	$CR_{ap} = 16.5$	$CR_{Mp} = 17$		
	MATE3020	**30**	**7:30 a 8:30**	**L-1h, W-1h, V-1h**	**s12**
	MATE3172	**127**	**10:30 a 11:30**	**L-1h, W-1h, V-1h**	**s1**
P2	**MATE3031**	**123**	**8:00 a 10:00**	M-2h, J-2h	**s1**
	$CR_{mp} = 12$	$CR_{ap} = 21$	$CR_{Mp} = 24.5$		
	MATE3021	**111**	**9:00 a 10:30**	**M-1:30h, J-1:30h**	**s2**
P3	**MATE4009**	**30**	**7:30 a 9:00**	**M-1:30h, J-1:30h**	**s12**
	$CR_{mp} = 10$	$CR_{ap} = 10$	$CR_{Mp} = 13$		

Tabla 4.10: Muestra de la solución, costo mínimo

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

La construcción de horarios de clases es un problema que deben solucionar las universidades al inicio de cada periodo lectivo, por tal razón surge el interés de desarrollar un algoritmo que obtenga una solución final o punto de partida para construir una solución final que cumpla con las restricciones impuestas por la institución. Durante la construcción del algoritmo y avance en el presente trabajo se presentan las siguientes conclusiones:

- Se ha elaborado un modelo general basado en los algoritmos genéticos que permite hallar una solución al problema de programación horaria.
- La solución se encontró en un tiempo razonable.
- La solución obtenida por el modelo no es una solución óptima, pero al compararlo con el horario final que obtuvo el Departamento de Ciencias Matemáticas se puede concluir que se cumple con la asignación de cursos de todos los profesores, se tiene una similitud alta en la coincidencia de horas solicitadas por el profesor, pero no se tiene una buena aproximación en la preferencia de horario del profesor por un curso determinado. En resumen, el horario generado por este modelo se puede utilizar como referencia para generar una versión final. Debido a consideraciones externas que se puede tomar, estos se podrían realizar con mayor facilidad manualmente luego de obtener el horario base.
- Los mejores resultados del algoritmo se presentan al considerar las preferencias de eventos dentro

del área de interés del profesor.

- Una ventaja del modelo planteado es que no se limita a un número determinado de salones, profesores o eventos que se deben dictar.
- Se logró construir una interfaz fácil de utilizar en el ambiente Guide de Matlab (ver Apéndice A).

5.2. Trabajos futuros

- Considerar la distancia entre salones asignados a un mismo profesor.
- Desarrollar un algoritmo que realice las particiones de los cursos en eventos que logren completar la carga mínima y asignar las compensaciones solicitadas por los profesores, basado en la cantidad total de estudiantes por curso.
- Considerar la diferencia horarios presentes en la zona central y periférica de la Universidad de Puerto Rico, Recinto de Mayagüez.
- Considerar la variación de los pesos de acuerdo con las peticiones del profesor respecto a su preferencia de cursos y bloques de horarios.
- Resolver el problema de minimizar el costo de construir un horario con una escala salarial de acuerdo con el rango académico del profesor.

Bibliografía

[1] Ahumada, J. (2014). *"Generación de horarios académicos en INACAP utilizando algoritmos genéticos"* (Tesis de maestría). Universidad de Chile, Santiago, Chile.

[2] Bejarano, G. (2010). *Planificación de horarios del personal de cirugía de un hospital del estado aplicando algoritmos genéticos (time tabling problem)* (Tesis de maestría). Pontificia universidad católica del Perú, Lima, Perú.

[3] Cooper, T.B y Kingston, J.H. (1995). *The Complexity of Timetable Construction Problems* (Informe No. 495). Sydney: The University of Sydney.

[4] Cormen, T., Leiserson, C., Rivest, R. y Stein, C. (2009). *Introducion to algorithms.* London, Inglaterra: The MIT Press.

[5] Costabel, S. (2005). *Metaheurísticas aplicadas a un Problema de Asignación de Salones y Horarios a Asignaturas* (Tesis de maestría). Universidad de la República, Montevideo, Uruguay.

[6] Esquivel, L. (2014). *Modelo matemático para la programación de un horario escolar con multilocalización de docentes* (Tesis de maestría). Universidad del Valle, Santiago De Cali, Colombia.

[7] Gore, P., Sonawane, P. y Potdar, S. (2017). Timetable Generation Using Ant Colony Optimization Algorithm. *International Journal of Innovative Research in Computer and Communication Engineering*, *5*, 6033-6039.

[8] Guerra, M., Pardo, E. y Salas, R. (2013). Problema del School Timetabling y algoritmos genéticos: una revisión. *Vínculos, 10*, 259-276.

[9] Marín, J. y Maya, P. (2015). Modelo lineal para la programación de clases en una institución educativa. *Ingeniería y Ciencia, 12*, 47-71.

[10] Mejía, J. y Paternina, C. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Educación en Ingenería, 9*, 140-149.

[11] Méndez, I., Miranda, J. y Zabala, P. (2016). Un nuevo enfoque para la programación horaria en universidades. *Ingeniería de Sistemas, 30*, 91-114.

[12] Montoya, J., Aponte, A. y Rosas, P. (2010). Un procedimiento de búsqueda voraz adaptativo probabilista para un problema monoproducto de localización de instalaciones no capacitado. *Ingeniería y Desarrollo, 28*, 15-32.

[13] Pitol, F. (s.f). *"Uso de algoritmos evolutivos para resolver el problema de asignación de horarios escolares en la facultad de Psicología de la Universidad Veracruzana"* (Tesis de maestría). Universidad Veracruzana, Veracruz, México.

[14] Rodríguez, J. (2012). *Algoritmos Meméticos en la Programación de Horarios de Clases* (Tesis de maestría). Universidad de Puerto Rico, Mayagüez, Puerto Rico.

[15] Silva, J. (2017). *Heurística para la Planificación de Horarios de la Universidad EAFIT* (Tesis de maestría). Universidad EAFIT, Medellín, Colombia.

[16] Suárez, O. (2011). Una aproximación a la heurística y metaheurísticas. *Inge@uan, 1*, 44-51.

[17] Suárez, V. (2012). *Disminución de la mortalidad académica en instituciones de educación básica y media mediante el empleo de técnicas inteligentes en la asignación de horarios* (Tesis de maestría). Universidad Nacional de Colombia, Manizales, Colombia.

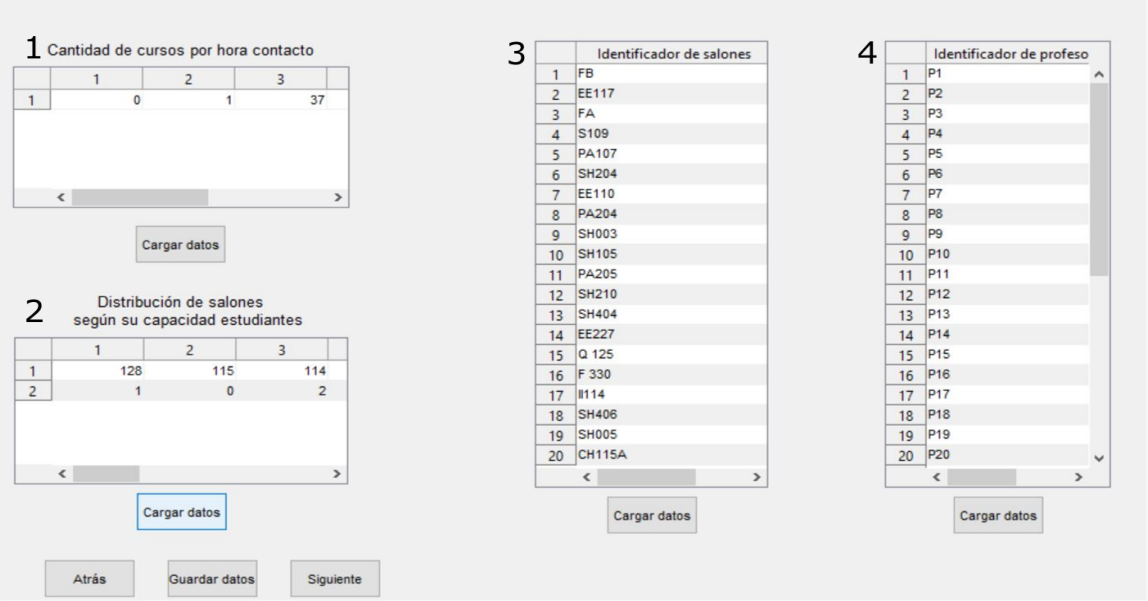
[18] Torres, C. (2013). *Programación de horarios y asignación de aulas de clases universitarias* (Tesis de maestría). Universidad de La Sabana, Chía, Colombia.

Apéndice

A. Interfaz

En este apéndice se presentan las pantallas de la interfaz que se elaboró. De la pantalla 1 a 4 son datos de entradas que se deben cargan desde archivos Excel, portapapeles o escribir directamente en la interfaz y la pantalla 5 genera el horario.

Pantalla 1



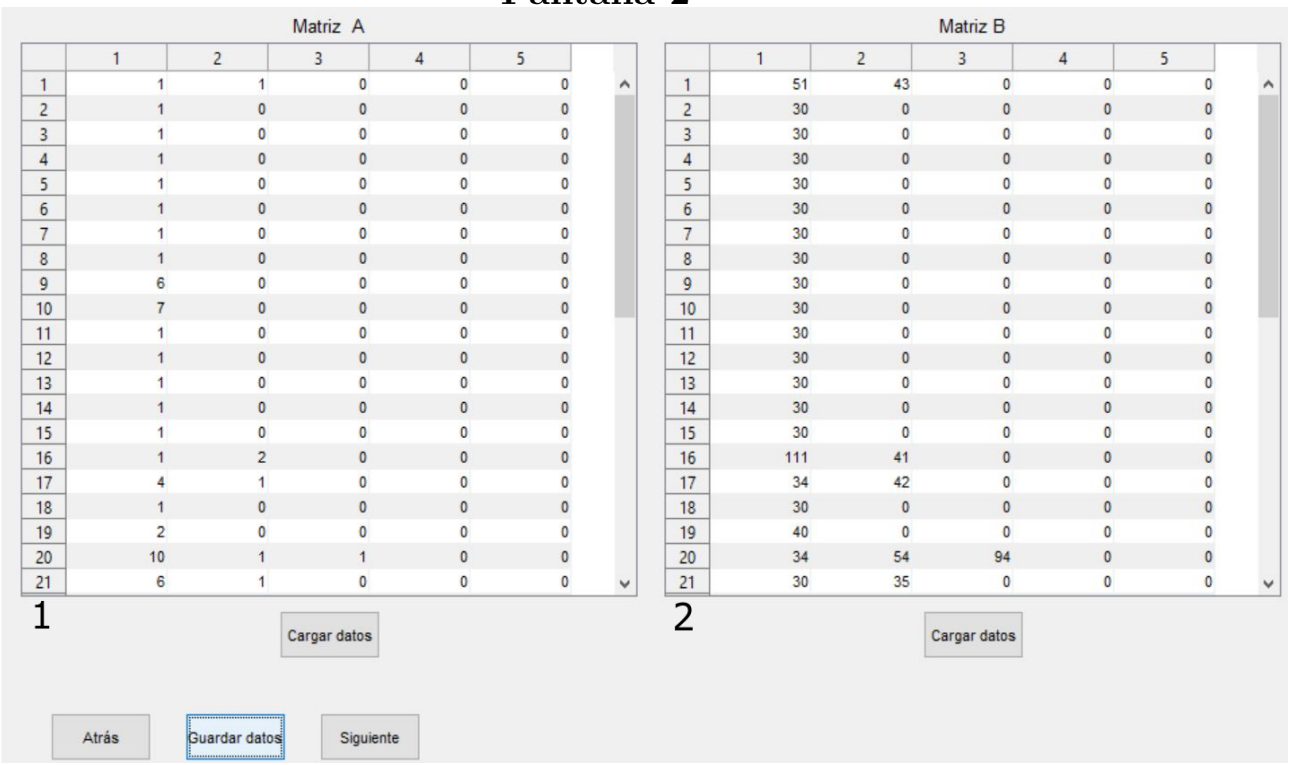
1. Cantidad de cursos por hora: donde la primera fila representa las horas contacto de los curso y la segunda fila la cantidad de cursos a programar según sus horas contacto. El archivo de Exel debe llamarse: Cursos. En el archivo solo debe tener la información de la segunda fila, la primer fila es generada por la interfaz.

2. Distribución de salones según su capacidad: cada dos entradas de la primera fila representa la cantidad mínima y máxima de estudiantes en salón. La entrada de fila 2 y columna $2s_r - 1$ representa la cantidad de salones por su capacidad de atender estudiantes. El archivo de Exel debe llamarse: Salones.

3. Salones: identificador de salones. Se deben ordenar en forma descendente de acuerdo a su capacidad. El archivo de Exel debe llamarse: ID_salones. El total de salones debe coincidir con el total de identificadores de salones.

4. Profesores: identificador de profesores. El archivo de Exel debe llamarse: ID_profesores

Pantalla 2

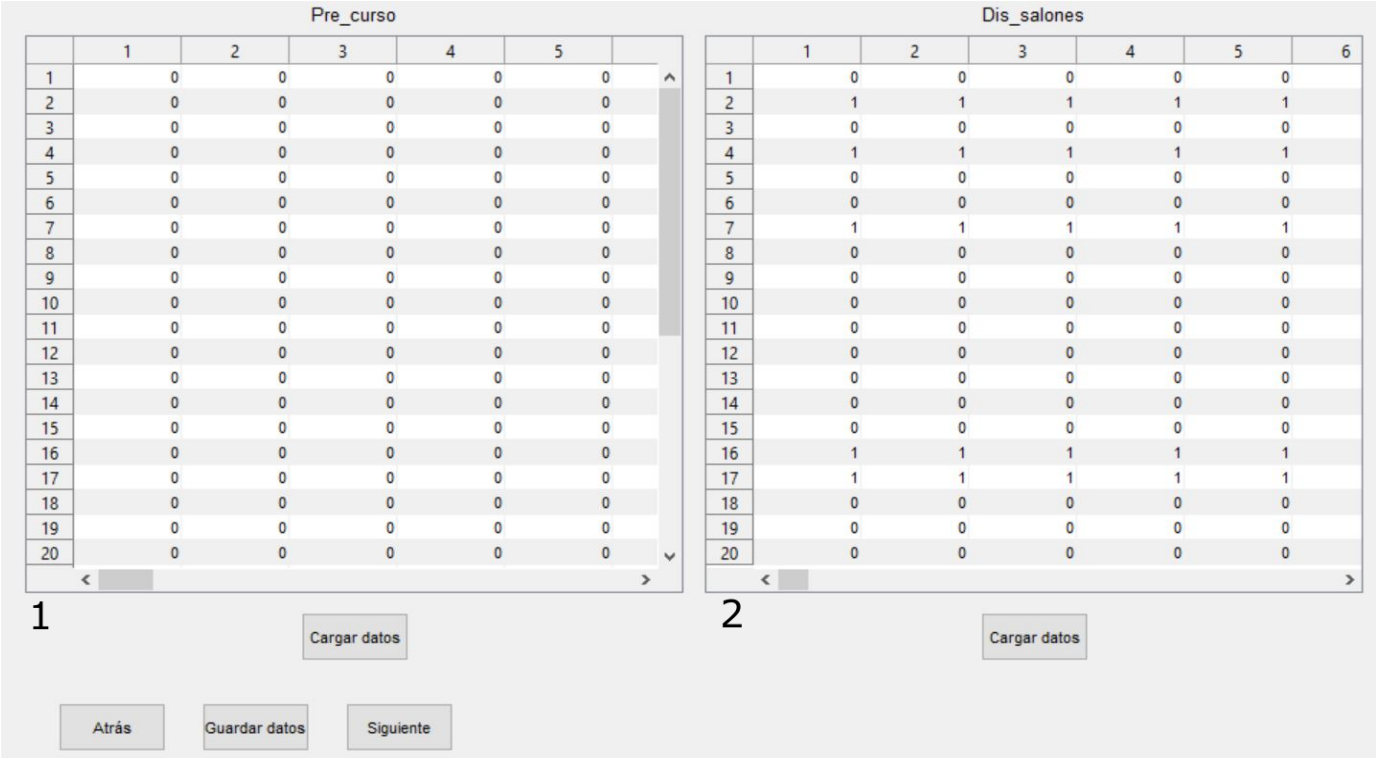


1. Matriz A: partición de cursos en eventos según capacidad de estudiantes. El archivo de Exel debe llamarse: MatrizA.

2. Matriz B: representa la cantidad de estudiantes por evento. El archivo de Exel debe llamarse: MatrizB.

Nota: La matriz de eventos A y B, son matrices superpuestas (ver Sección 3.2.2).

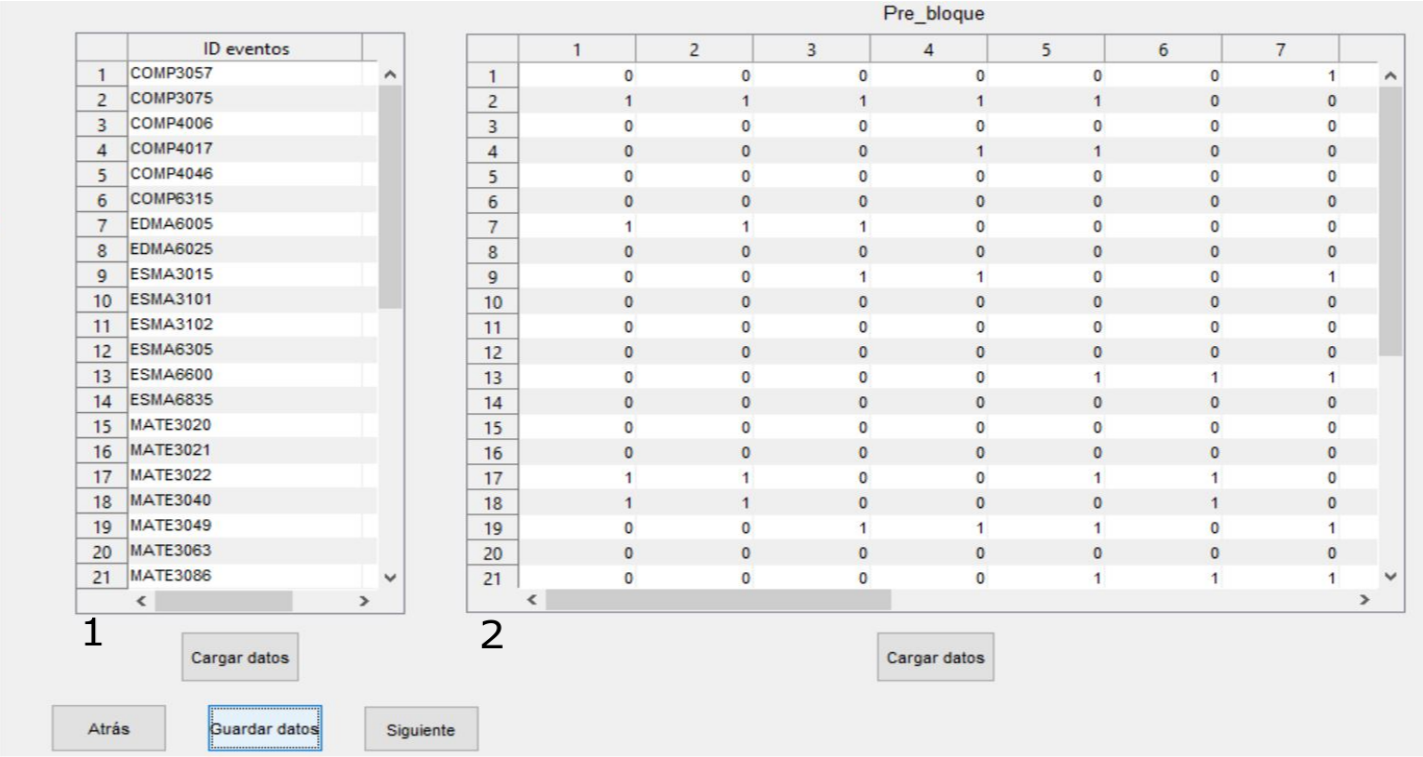
Pantalla 3



1. Pre_curso: se incluye la preferencia del profesor por curso, la relación curso cantidad de estudiantes y la cantidad mínima y máxima de créditos por profesor. El archivo de Exel debe llamarse: Pre_curso.

2. Pre_bloque: indica la disponibilidad por salón. Donde 1 indica que el salón está disponible y 0 en otro caso. El archivo de Exel debe llamarse: Pre_bloque.

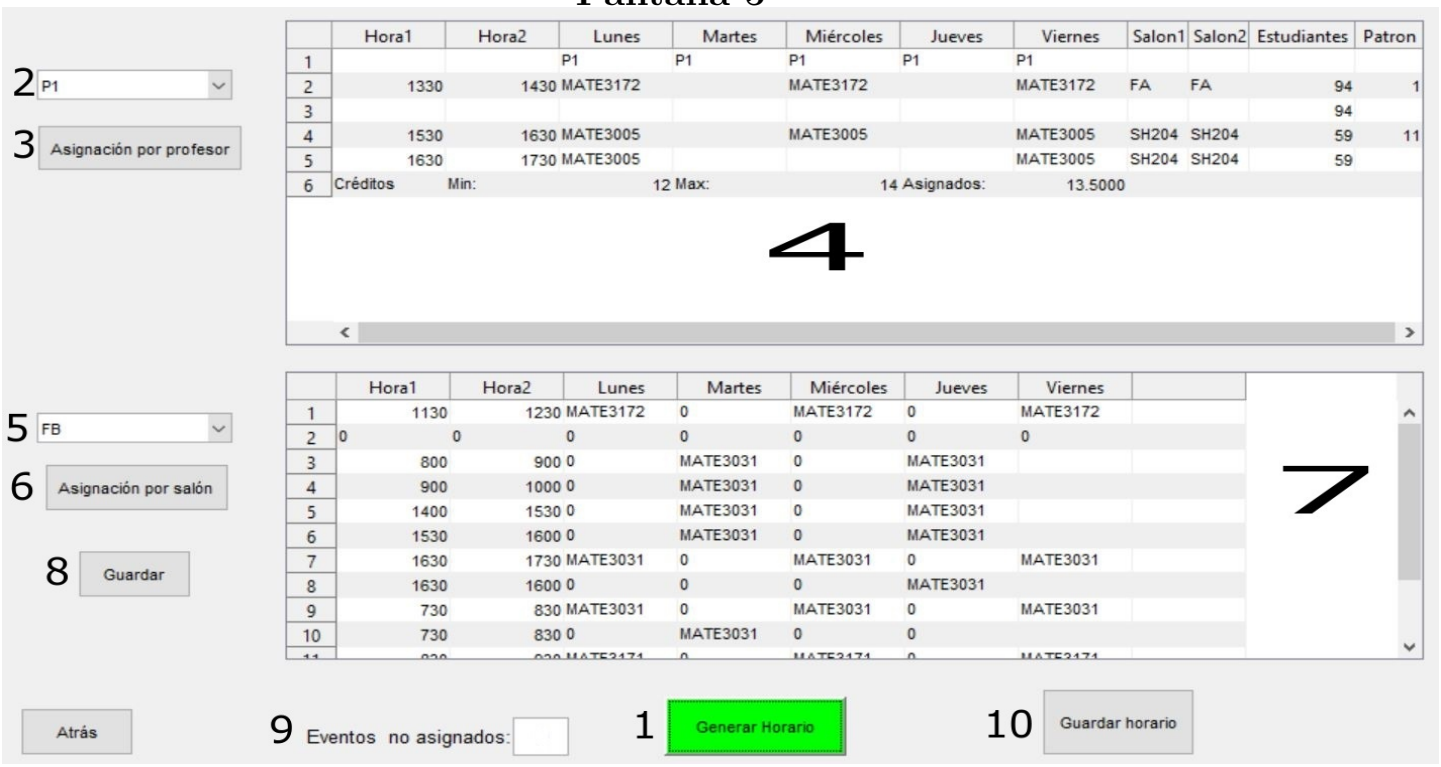
Pantalla 4



1. ID_cursos: identificador de eventos. El archivo de Exel debe llamarse: ID_cursos.

2. Pre_bloque: indica la preferencia del profesor por bloque de horario. El archivo de Exel debe llamarse: Pre_bloque.

Pantalla 5



1. Permite generar el horario.

2. Se despliega la lista de profesores.

3. Luego de seleccionar un profesor en el paso 2 permite ver la asignación de este.

4. La tabla proporciona la hora de inicio (Hora1) y final (Hora2) de los eventos, salón y cantidad de créditos asignados al profesor seleccionado en el paso 2. Ejemplo de lectura, el profesor P1 tiene una carga mínima y máxima de 12 y 14 créditos, respectivamente; el algoritmo le asignó 13.4 créditos. Se le asignó el evento MATE3172 con 94 estudiantes en el salón FA de 13:30 a 14:30, los días lunes, miércoles y jueves (patrón 1), y el evento MATE3005 con 59 estudiantes en el salón SH204, Llos días lunes de 15:30 a 17:30, miércoles de 15:30 a 16:30 y viernes de 15:30 a 17:30 (patrón 1).

5. Se despliega la lista de salones.

6. Luego de seleccionar un salón en el paso 5 permite ver la asignación de este.

7. La tabla muestra la hora de inicio (Hora1) y final (Hora2) de los eventos asignados en el salón seleccionado en el paso 5.

8. Guarda la información de los eventos asignados en el salón seleccionado.

9. Muestra la cantidad de eventos no asignados.

10. Permite guardar el horario generado en un archivo Excel. El archivo de Exel debe llamarse: Horario_final.

B. Patrones horarios en el caso de estudio

Los patrones horarios son las diferentes maneras que el Departamento de Ciencias Matemáticas puede asignar eventos. La combinación entre bloque horario y patrón horario nos da la asignación de un evento.

Patrones horarios para eventos tres de horas						
Patrón 1	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
Patrón 2	MARTES			JUEVES		
	30 min	30 min	30 min	30 min	30 min	30 min

Tabla B1: Patrones horarios para eventos cuadro de horas

Patrones horarios para eventos tres de horas						
Patrón 3	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
	30 min	30 min				
Patrón 4	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
			30 min	30 min		
Patrón 5	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
					30 min	30 min
Patrón 6	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min		
	30 min	30 min	30 min	30 min		
Patrón 7	LUNES		MIÉRCOLES		VIERNES	
			30 min	30 min	30 min	30 min
			30 min	30 min	30 min	30 min
Patrón 8	MARTES			JUEVES		
	30 min	30 min	30 min	30 min	30 min	30 min
	30 min			30 min		

Tabla B2: Patrones de horarios para eventos de cuatro horas

Eventos de cinco horas						
Patrón 16	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
	30 min	30 min			30 min	30 min
Patrón 17	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
	30 min	30 min	30 min	30 min		
Patrón 18	LUNES		MIÉRCOLES		VIERNES	
	30 min	30 min	30 min	30 min	30 min	30 min
			30 min	30 min	30 min	30 min

Tabla B3: Patrones de horarios para eventos de cinco horas

C. División de cursos en el caso de estudio

Las siguiente tabla muestra la información utilizada en la matriz A (partición de cursos por eventos según su capacidad d estudiantes) y matriz B (cantidad de estudiantes por evento)en el caso de estudio. Ejemplo de lectura: el curso COMP3057 tiene dos eventos (secciones), un evento con 51 estudiantes y un con 43, para un total de 94 estudiantes en el curso COMP3057.

						Total
COMP3057	1	1	0	0	0	2
Estudiantes	51	43	0	0	0	94
COMP3075	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
COMP4006	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
COMP4017	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
COMP4046	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
COMP6315	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
EDMA6005	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
EDMA6025	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
ESMA3015	6	0	0	0	0	6
Estudiantes	30	0	0	0	0	30
ESMA3101	7	0	0	0	0	7
Estudiantes	30	0	0	0	0	30
ESMA3102	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
ESMA6305	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30

						Total
ESMA6600	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
ESMA6835	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE3020	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE3021	1	2	0	0	0	3
Estudiantes	111	41	0	0	0	193
MATE3022	4	1	0	0	0	5
Estudiantes	34	42	0	0	0	178
MATE3040	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE3049	2	0	0	0	0	2
Estudiantes	40	0	0	0	0	80
MATE3063	10	1	1	0	0	12
Estudiantes	34	54	94	0	0	488
MATE3086	6	1	0	0	0	7
Estudiantes	30	35	0	0	0	216
MATE3171	9	4	2	1	0	16
Estudiantes	80	124	34	67	0	1351
MATE3172	1	3	0	0	0	4
Estudiantes	127	94	0	0	0	409
MATE3181	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30

Continúa en la siguiente página ...

						Total
MATE4000	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4008	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4009	12	1	0	0	0	13
Estudiantes	30	40	0	0	0	400
MATE4023	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4031	2	1	0	0	0	3
Estudiantes	34	40	0	0	0	108
MATE4051	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4061	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4071	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4120	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE6025	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30

						Total
MATE6101	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE6201	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE6261	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE6675	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE3031	4	1	3	1	0	9
Estudiantes	123	81	33	45	0	717
MATE3032	4	1	1	1	1	8
Estudiantes	33	45	81	93	123	474
MATE3048	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE4145	2	0	0	0	0	2
Estudiantes	30	0	0	0	0	60
COMP3010	4	0	0	0	0	4
Estudiantes	30	0	0	0	0	120
COMP3110	1	0	0	0	0	1
Estudiantes	30	0	0	0	0	30
MATE	1	0	0	0	0	1
Estudiantes	59	0	0	0	0	59

D. Matriz Pre_curso y Pre_bloque en el caso de estudio

Las filas representan los 37 profesores y las columnas la cantidad de cursos y la preferencia relación eventos-estudiantes. En el caso de estudio la matriz es de tamaño 37×54.

$$
\begin{vmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{10}&\frac{4}{5}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&\frac{9}{10}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{9}{10}&\frac{1}{5}&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{9}{10}&\frac{1}{10}&0&0&0&0&0&0&0&0\\
0&0&0&0&0&\frac{1}{100}&0&0&\frac{1}{10}&0&0&\frac{1}{10}&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&\frac{1}{100}&\frac{4}{5}&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{100}&0&0&\frac{1}{100}&0&0&0&0&\frac{1}{100}&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&\frac{1}{100}&0&0&0&0&0&0&\frac{1}{100}&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{100}&0&\frac{17}{20}&0&0&\frac{1}{10}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{10}&\frac{1}{100}&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{10}&0&0&0&\frac{17}{20}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{17}{20}&0&0&\frac{1}{10}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{4}{5}&0&0&\frac{1}{100}\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{9}{10}&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{10}&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\frac{1}{10}&1&0&0
\end{vmatrix}
$$

(matriz bloque de 25×25). Continua matriz Pre_curso...

$$
\begin{vmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{4}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{7}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & \frac{4}{5} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{9}{10} & 0 & \frac{9}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9}{10} & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{4}{5} & \frac{1}{10} & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9}{10} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{vmatrix}
$$

(matriz bloque de fila 26 a 37, columna 26 a 52)

Cantidad mínima y máxima de créditos.

$P1$	12	14
$P2$	12	$\frac{43}{2}$
$P3$	10	10
$P4$	12	14
$P5$	12	13
$P6$	6	10
$P7$	12	14
$P81$	12	17
$P9$	11	12
$P10$	8	10
$P11$	12	21
$P12$	12	16
$P13$	12	12
$P14$	11	12
$P15$	6	11
$P16$	6	9
$P17$	12	14
$P18$	12	14
$P19$	12	12

$P20$	6	6
$P21$	12	12
$P22$	12	15
$P23$	12	22
$P24$	12	16
$P25$	12	12
$P26$	11	11
$P27$	11	11
$P28$	12	12
$P29$	12	12
$P30$	11	11
$P31$	7	7
$P32$	9	12
$P33$	12	12
$P34$	12	12
$P35$	11	11
$P36$	6	6
$P37$	3	3

$P1$	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
$P2$	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0
$P3$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
$P4$	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0
$P5$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
$P6$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
$P7$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$P8$	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
$P9$	0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0
$P10$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
$P11$	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1
$P12$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
$P13$	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
$P14$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
$P15$	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0
$P16$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$P17$	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
$P18$	1	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
$P19$	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
$P20$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
$P21$	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
$P22$	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$P23$	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
$P24$	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
$P25$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

(matriz bloque de 25×17). Continua matriz Pre_bloque...

$P26$	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
$P27$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
$P28$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
$P29$	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$P30$	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
$P31$	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$P32$	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
$P33$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
$P34$	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
$P35$	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
$P36$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
$P37$	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(matriz bloque de fila 26 a 37)

E. Conversión crédito-evento en el caso de estudio

La siguiente tabla muestra el número de horas-crédito adicionales para el profesor para diferentes rangos de estudiantes matriculados y el número de horas contacto por semana asignadas al profesor del curso; por ejemplo, si un profesor está enseñando un curso de conferencia de tres horas contacto por semana con 62 estudiantes recibirá 1.5 créditos adicionales.

1		2		3		4		5		6		Créditos
Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	adicionales
1	44	1	37	1	34	1	33	1	32	1	32	0
45	75	38	52	35	44	34	41	33	38	33	37	0.5
75	104	53	67	45	54	42	48	39	44	38	42	1
105	134	68	82	55	64	49	56	45	50	43	47	1.5
135	164	83	97	65	74	57	63	51	56	48	52	2
0	0	98	112	75	84	64	71	57	62	53	57	2.5
0	0	113	127	85	94	72	78	63	68	58	62	3
0	0	128	142	95	104	79	86	69	74	63	67	3.5
0	0	143	147	105	114	87	93	75	80	68	72	4
0	0	0	0	115	124	94	101	81	86	73	77	4.5
0	0	0	0	125	134	102	108	87	92	78	82	5
0	0	0	0	135	144	109	116	93	98	83	87	5.5
0	0	0	0	145	154	117	123	99	104	88	92	6
0	0	0	0	0	0	124	131	105	110	93	97	6.5
0	0	0	0	0	0	132	138	111	116	98	102	7
0	0	0	0	0	0	139	146	117	122	103	107	7.5
0	0	0	0	0	0	147	153	123	128	108	112	8

Tabla E1: Tabla de conversión de créditos