Revista digital
Matemática, Educación e Internet
https://revistas.tec.ac.cr/index.php/matematica
Vol 26, No 2. Marzo, 2026 – Agosto, 2026

ISSN 1659 -0643
DOI: XXX

Una introducción a la programación de horarios: restricciones, parámetros y funciones objetivo

| An Introduction to Timetabling: Constraints, Parameters and Objective Functions |

| Uma introdução à programação de horários: restrições, parâmetros e funções objetivo |

José Gerardo Córdoba Hernández[1]
jose.cordoba.hernandez@una.cr
Universidad Nacional
Heredia, Costa Rica

Recibido: 5 de junio de 2024 Aceptado: 30 de abril de 2025

Resumen: La programación de horarios en instituciones educativas constituye un problema de optimización combinatoria cuyo objetivo es asignar cursos, docentes, aulas y bloques de tiempo bloques de tiempo de manera coordinada, respetando un conjunto de restricciones. Este artículo presenta una introducción al modelado matemático de dicho problema, diferenciando restricciones estrictas (hard constraints) y flexibles (soft constraints), así como la definición de conjuntos, parámetros y funciones objetivo que permiten estructurar formalmente el modelo. Se presentan tres ejemplos ilustrativos que muestran cómo la variación en los pesos asignados a preferencias y penalizaciones, junto con la reasignación puntual de eventos, impacta directamente el valor de la función objetivo. Los ejemplos evidencian que el equilibrio entre criterios y pequeños ajustes en las asignaciones pueden mejorar de forma significativa la calidad de los horarios generados. Este trabajo constituye una introducción didáctica y un punto de partida para investigaciones futuras orientadas al abordaje del problema de programación de horarios.

Palabras Clave: Optimización combinatoria, asignación de recursos, programación de horarios.

Abstract: Timetabling in educational institutions is a combinatorial optimization problem whose objective is to assign courses, instructors, classrooms, and time slots in a coordinated manner while satisfying a set of constraints. This article introduces the mathematical modeling of such a problem, distinguishing between hard and soft constraints, and formalizing the definition of sets, parameters, and objective functions that structure the model. Three illustrative examples are presented to demonstrate how variations in the weights assigned to preferences and penalties, together with the punctual reassignment of events, directly affect the objective function value. These examples show that balancing criteria and implementing minor adjustments in the allocations can significantly improve the quality of the generated timetables. This work provides a didactic introduction and a starting point for future research aimed at addressing the academic timetabling problem.

Keywords: Combinatorial optimization, resource allocation, Timetabling .

[1]José Gerardo Córdoba Hernández. Profesor de la Escuela de Matemáticas de la Universidad Nacional de Costa Rica. Dirección Postal: Santa Bárbara, Heredia, Costa Rica. Código Postal: 40401 Correo electrónico: jose.cordoba.hernandez@una.cr.

Resumo: A programacão de horários em instituições educativas constitui um problema de otimização combinatória cujo objetivo é atribuir cursos, docentes, salas e blocos de tempo de forma coordenada, respeitando um conjunto de restrições. Este artigo apresenta uma introdução ao modelamento matemático desse problema, diferenciando restrições estritas (hard constraints) e flexíveis (soft constraints), assim como a definição de conjuntos, parâmetros e funções objetivo que permitem estruturar formalmente o modelo. São apresentados três exemplos ilustrativos que mostram como a variação nos pesos atribuídos a preferências e penalizações, junto com a reatribuição pontual de eventos, impacta diretamente o valor da função objetivo. Os exemplos evidenciam que o equilíbrio entre critérios e pequenos ajustes nas atribuições podem melhorar de forma significativa a qualidade dos horários gerados. Este trabalho constitui uma introdução didática e um ponto de partida para futuras pesquisas orientadas ao abordamento do problema de programacão de horários.

Palavras-chave: Otimização combinatória, atribuição de recursos, programação de horários.

1. Introducción

La programación de horarios en instituciones educativas constituye un problema de optimización combinatoria cuyo objetivo es asignar cursos, docentes, aulas y bloques de tiempo bloques de tiempo de manera coordinada, respetando un conjunto de restricciones. No obstante, la mayoría de los trabajos en este campo se dirigen a investigadores con experiencia en modelado matemático y en algoritmos de optimización. En este contexto, el presente artículo propone una introducción accesible a la formulación de problemas de programación de horarios, con énfasis en los primeros pasos, definición de conjuntos, parámetros fundamentales, las restricciones y los ejemplos de funciones objetivo.

El propósito central de este trabajo es ofrecer una base estructurada para el modelado de este tipo de problemas, de modo que los lectores puedan comprender sus principales desafíos y, al mismo tiempo, obtener una primera aproximación a los enfoques disponibles para su resolución en el marco del estado del arte. A través de explicaciones, se busca contribuir a cerrar la brecha existente entre la teoría y la práctica en la optimización de horarios educativos.

La programación de horarios es un problema recurrente en las instituciones educativas y organizaciones empresariales (Hernández et al., 2008). Este desafío se debe a la necesidad de coordinar múltiples restricciones, como la disponibilidad de recursos físicos (aulas), la carga académica del personal docente, cursos ofrecidos, distancia entre aulas y las preferencias de los docentes o estudiantes, todo dentro de un intervalo de tiempo definido (Burke & Petrovic, 2002; Córdoba, 2018; Mejía & Paternina, 2010).

En la mayoría de instituciones educativas la resolución del problema de programación de horarios se realiza de manera manual, el cual es un proceso que puede extenderse durante varios días o incluso semanas. Este método convencional limita significativamente el análisis a un subconjunto de posibles soluciones, restringiendo así la capacidad de encontrar opciones óptimas (Cleger Tamayo et al., 2007; Córdoba, 2018). Además, la asignación manual de horarios usualmente requiere la dedicación de una o varias personas, quienes deben garantizar que se cumplan, al menos, las condiciones mínimas establecidas por la institución. Sin embargo, este método incrementa la carga administrativa y puede dar lugar a ineficiencias en la asignación de recursos académicos (Alghamdi et al., 2020; Bashab et al., 2023). En este sentido, la solución a este problema requiere enfoques que permitan encontrar soluciones factibles en tiempos razonables.

En la programación de horarios, surgen interrogantes críticas, como: ¿qué técnica es más efectiva para abordar este desafío, un método exacto o aproximado? ¿Es factible garantizar que el método seleccionado proporcione una solución viable en un tiempo computacional razonable? Si bien la literatura especializada no ofrece un consenso definitivo sobre estas cuestiones, sin embargo, numerosos estudios han clasificado este problema como NP-completo, debido a la complejidad combinatoria de sus restricciones y al elevado costo computacional asociado a su resolución (Floyd, 1967; López-Cruz, 2015).

Incluso en su forma más sencilla, el problema de programación de horarios se clasifica como NP-completo, lo que significa que evaluar todas las combinaciones posibles para encontrar la solución óptima resulta impracticable en problemas de gran escala (Even et al., 1976). Esta clasificación implica que encontrar soluciones óptimas mediante métodos exactos es computacionalmente inviable para problemas de gran escala. Por esta razón, es necesario explorar métodos alternativos que permitan abordar este desafío de manera eficiente.

Debido a la complejidad inherente del problema, los métodos aproximados, entre ellos heurísticas, metaheurísticas y matheurísticas[1] , constituyen una alternativa eficaz para obtener soluciones factibles y de buena calidad en tiempos computacionalmente razonables (Bashab et al., 2023; González et al., 2020; Saltos & Benavides, 2021). Aunque no garantizan encontrar soluciones óptimas, estos enfoques son capaces de obtener resultados aceptables con un uso razonable de recursos computacionales. En contraste, los métodos exactos, que realizan búsquedas exhaustivas, consumen una cantidad significativa de recursos computacionales, lo que dificulta su aplicación en problemas de gran escala (Bashab et al., 2023; Restrepo & Moreno, 2011).

2. Estado del arte

El desarrollo de un modelo orientado a la obtención de soluciones óptimas en problemas de asignación de recursos requiere un análisis de las investigaciones previas en el área. Dicho análisis tiene como propósito identificar las técnicas más relevantes y apropiadas, disponibles en la literatura para abordar este tipo de problemas.

Entre las técnicas empleadas para resolver problemas de asignación de recursos y, en particular, de programación horaria, se distinguen dos enfoques principales: los métodos exactos y los métodos aproximados. A continuación, se describen algunas características y aplicaciones en la literatura.

Métodos exactos: Estos métodos garantizan la obtención de la solución óptima del problema, ya que exploran de manera sistemática el espacio de búsqueda o aplican algoritmos matemáticos que aseguran optimalidad bajo determinadas condiciones. Se consideran algoritmos completos, aunque su desempeño suele verse limitado ante instancias de gran escala debido al crecimiento exponencial de la complejidad computacional (Guerra Cubillos et al., 2013).

En el contexto de la programación de horarios académicos, destacan enfoques como la programación lineal entera, la programación lineal y los algoritmos de ramificación y acotamiento (Branch and Bound). Por ejemplo, Saltos y Benavides (2021) proponen un modelo de programación lineal entera para la calendarización de cursos universitarios en una institución privada de Ecuador, cuyo objetivo fue optimizar la asignación de aulas y la distribución de materias durante el semestre. La comparación entre la planificación manual y el modelo propuesto evidenció mejoras significativas en los horarios, como: reducción del tiempo de planificación y una distribución más equilibrada de la carga académica.

De manera similar, Canseco et al. (2016) aplicaron un modelo de programación lineal entera resuelto mediante Branch and Bound, implementado en el software de optimización LINGO® 10. El modelo abordó un problema con 735 variables enteras y 471 restricciones, logrando una solución en apenas 4 segundos. De esta manera, se redujo un trabajo que tradicionalmente tomaba semanas a un tiempo computacional mínimo, mejorando sustancialmente la asignación de recursos de la institución.

Métodos aproximados: Estos métodos no garantizan la obtención de la solución óptima global, sino

[1]Es un enfoque que combina técnicas exactas y heurísticas.

que buscan soluciones de alta calidad en tiempos razonables, lo que resulta particularmente útil en problemas de gran escala o alta complejidad combinatoria. Se consideran algoritmos incompletos porque exploran solo una parte del espacio de búsqueda, aunque su flexibilidad y eficiencia práctica los hacen ampliamente utilizados en la programación horaria (Burke & Petrovic, 2002; Mejía & Paternina, 2010).

Dentro de este grupo se incluyen las heurísticas, metaheurísticas y otros enfoques inspirados en procesos naturales o técnicas de inteligencia artificial. Algunos ejemplos aplicados a la programación académica son la colonia de hormigas, las redes neuronales artificiales, el recocido simulado (Simulated Annealing), la búsqueda tabú (Tabu Search) y los algoritmos genéticos.

Bustos et al. (2014) implementaron una metaheurística inspirada en los principios de la evolución biológica, empleando mecanismos como la selección natural y la reproducción, con el fin de resolver el problema de calendarización en una institución universitaria de Chile. El objetivo de la investigación fue reducir los tiempos de planificación de horarios, minimizando al mismo tiempo el incumplimiento de las restricciones establecidas.

De manera similar, el trabajo de Restrepo y Moreno (2011) tuvo como objetivo implementar una metaheurística basada en la Búsqueda Tabú, con el fin de proporcionar una solución parametrizable, eficiente y de rápida ejecución para el problema de calendarización. Esta propuesta estuvo orientada a optimizar la asignación académica, beneficiando tanto la labor docente como la experiencia de los estudiantes en la institución educativa. En el caso de los docentes, se garantizó que las cargas académicas fueran asignadas conforme a su disponibilidad y perfil profesional. En cuanto a los estudiantes, se procuró que los horarios no presentaran traslapes entre cursos y que las clases se organizaran de manera consecutiva.

Algunos de los estudios más recientes que abordan la programación de horarios son: Dünke y Nickel (2023), Rezaeipanah et al. (2020), Siew et al. (2024) y Bashab et al. (2023). El primero, se aplicó en el Instituto Tecnológico de Karlsruhe en Alemania, se enfocó en una matheurístico utilizando un algoritmo genético combinado con programación entera para optimizar la asignación de clases. El segundo trabajo presenta un enfoque híbrido para la programación de horarios universitarios, basado en un algoritmo genético paralelo mejorado, combinado con búsqueda local, cuyo propósito es mejorar la asignación de horarios y aulas para los cursos, garantizando el cumplimiento de las restricciones estrictas[2] (hard constraints) y favoreciendo el cumplimiento de las restricciones flexibles[3] (soft constraints). Por su parte, los dos últimos trabajos corresponden a revisiones de la literatura reciente sobre las metodologías aplicadas para resolver el problema de programación de horarios en el ámbito universitario.

A partir de la revisión de la literatura, la resolución del problema de calendarización de cursos universitarios sigue siendo un desafío vigente, con múltiples estudios que exploran distintas metodologías y enfoques de optimización publicados hasta la fecha. En este trabajo, se adopta un enfoque que proporciona una guía para la formulación de problemas de programación horaria, mediante la presentación de restricciones, parámetros y funciones objetivo que pueden ser considerados en la planificación académica. Esta guía busca ofrecer una perspectiva estructurada que facilite la comprensión de los elementos clave del modelo y sirva como referencia para futuras investigaciones.

3. Elementos clave para la programación horaria

Para modelar y resolver un problema de programación horaria de manera eficiente, primero se debe comprender los elementos que lo componen. Estos elementos se organizan en conjuntos, que repre-

[2] Deben cumplirse obligatoriamente (ej. un curso no puede asignarse a dos aulas al mismo tiempo).

[3] Deseables, pero pueden violarse con penalización en la función objetivo (ej. preferencia de un docente por un aula específica).

sentan los distintos componentes involucrados en la asignación de horarios. Sin embargo, la complejidad del problema está relacionada directamente con la cantidad de elementos y las restricciones que regulan cómo estos conjuntos interactúan entre sí.

En las siguientes secciones, se presentan los conjuntos, parámetros y restricciones más comunes en la programación horaria, así como su importancia en la construcción de un modelo de programación horaria.

3.1. Definición de conjuntos fundamentales

La definición de los conjuntos constituye un paso fundamental en la modelación del problema de programación horaria, ya que permite estructurar de manera ordenada los elementos involucrados en la asignación de recursos. Dichos conjuntos representan los componentes principales del problema como: cursos, docentes, aulas y bloques horarios, y establecen la base para la formulación tanto de las restricciones como de la función objetivo. A continuación, se ilustran algunos conjuntos que pueden emplearse en la formulación de un modelo.

1. $C = \{1, 2, \ldots, j\}$: conjunto de cursos de la institución.

   Ejemplo 1

   1: Matemática General, 2: Cálculo I y 3: Cálculo II.

2. Para cada curso $j \in C$, se define el conjunto de eventos asociados como C_j.

   Ejemplo 2

   C_1: Matemática General, C_2: Cálculo I y C_3: Cálculo II.

3. Cada curso C_j puede dividirse en eventos (secciones, grupos), representados por e_{ij}, donde el subíndice i identifica el evento y j el curso.

   Ejemplo 3

   Cinco eventos del curso C_1: Matemática General se representan como e_{11} (Matemática General-01), e_{21} (Matemática General-02), e_{31} (Matemática General-03), e_{41} (Matemática General-04) y e_{51} (Matemática General-05).
   En este caso, $C_1 = \{e_{11}, e_{21}, e_{31}, e_{41}, e_{51}\}$. De manera general, el conjunto de eventos asociados al curso C_j se expresa como $C_j = \{e_{1j}, e_{2j}, \ldots, e_{mj}\}$.

4. E el conjunto de todos los eventos posibles definidos en la institución. A partir de él es posible definir subconjuntos específicos según sus características. Por ejemplo, los eventos con duración de tres y cuatro horas pueden agruparse en $E_3 \subseteq E$ y $E_4 \subseteq E$, respectivamente.

   También puede considerarse un subconjunto de eventos preasignados, $E_{pre} \subset E$, que incluye aquellos cuya asignación ha sido determinada previamente por la dirección del departamento o la persona encargada de la calendarización. Esta preasignación puede justificarse por la idoneidad del docente para impartir un evento particular o por las características del aula requerida.

5. $P = \{1, 2, \ldots, p\}$: conjunto de docentes disponibles.

Ejemplo 4

1: Docente Carlos, 2: Docente Alberto y 3: Docente José.

6. $A = \{1, 2, \ldots, a\}$: conjunto de aulas de la institución. Es posible definir subconjuntos específicos, como laboratorios ($A_{lab} \subset A$).

Ejemplo 5

1: Aula 218, 2: Aula 219, 3: Aula 220 y 4: Laboratorio 1.

7. $B = \{1, 2, \ldots, b\}$: conjunto de bloques horarios. Cada bloque representa un intervalo de tiempo disponible para asignar un curso. En este trabajo se considerarán bloques horarios de tres horas; no obstante, de acuerdo con la programación de la institución, también es posible definir bloques horarios de una hora, dos horas, u otras duraciones.

Ejemplo 6

1: Lunes 7:00 a. m a 9:00 a.m, 2: Lunes 9:00 a. m a 12:00 m.d y 17: Viernes 7:00 a. m a 9:00 a.m.

Con la definición de los conjuntos fundamentales se ha establecido el marco estructural del problema. El siguiente paso consiste en introducir los parámetros y variables de decisión, los cuales permiten cuantificar la información relevante y expresar las posibles asignaciones de forma matemática. Esta transición es clave para avanzar hacia la formulación de las restricciones y, posteriormente, de la función objetivo.

3.2. Definición de parámetros y variable de decisión

Al igual que los conjuntos, los parámetros y variables constituyen un paso muy importante en la modelación del problema de programación horaria, ya que permite organizar y garantizar su correcta formulación. Los parámetros representan valores fijos que describen las condiciones del sistema, como la capacidad de las aulas, la disponibilidad de los docentes o la carga académica asociada a cada evento. Por otro lado, las variables son elementos dinámicos que determinan la asignación de recursos, indicando, por ejemplo, si un curso es programado en un determinado horario o asignado a un docente específico. En esta sección se presentan algunos parámetros y un ejemplo de variable de decisión que se pueden emplear en la formulación de un problema de calendarización horaria.

A continuación, se presentan algunos parámetros que pueden considerarse en la formulación de un problema de programación horaria. La lista no pretende ser exhaustiva, sino ilustrativa, y tiene como objetivo mostrar ejemplos de cómo se pueden cuantificar distintos aspectos del problema.

1. C_{ij} representa la carga académica asociada al evento $e_{ij} \in E$.

2. CRm_p y CRM_p indican la carga académica mínima y máxima que pueden asignarse al profesor $p \in P$, respectivamente.

3. CRa_p corresponde a la carga académica asignada al profesor $p \in P$ por el algoritmo.

4. Cm_a y CM_a indican la cantidad mínima y máxima de estudiantes permitidos en el aula $a \in A$, respectivamente.

5. CC_j representa el total estimado de estudiantes que se matricularán en el curso $C_j \in C$

6. CE_{ij} corresponde a la cantidad de estudiantes asignados al evento $e_{ij} \in E$ por el algoritmo.

7. T_p denota el límite de distancia o tiempo máximo de traslado permitido para el docente $p \in P$ entre aulas.

8. d_{rk} puede representa la distancia ó el tiempo de traslado entre al aula a_r y el aula a_k.

9. w_i son pesos (coeficientes numéricos) que sirven para asignar importancia relativa a cada criterio dentro del modelo, con $i \in \mathbb{N}$. Estos valores pueden normalizarse, por ejemplo, imponiendo $\sum\limits_{i\in\mathbb{N}} w_i = 1$.
   Ejemplo: en un problema de horarios donde se desea minimizar tanto los conflictos de horarios (f_1) como los tiempos de traslado (f_2), la función objetivo podría expresarse como:

   $$\text{mín}\, Z = w_1 f_1 + w_2 f_2.$$

   Si se eligen $w_1 = 0.7$ y $w_2 = 0.3$, el modelo prioriza la reducción de conflictos por encima de los traslados. En cambio, si se asignan $w_1 = 0.3$ y $w_2 = 0.7$, la prioridad se invierte. De esta forma, los pesos permiten reflejar las preferencias o políticas de la institución al momento de generar los horarios.

A modo de ejemplo, se propone la siguiente variable de decisión, la cual permite representar la asignación de eventos a docentes, aulas y bloques horarios:

$$x_{e_{ij},p,a,b} = \begin{cases} 1 & \text{si el evento } e_{ij} \text{ está asignado al docente } p \in P \text{ en el aula } a \in A \text{ durante el bloque } b \in B \\ 0 & \text{en otro caso} \end{cases}$$

3.3. Restricciones en la programación horaria

Uno de los objetivos en la elaboración de un horario académico es asignar a cada evento un docente y un aula disponible, asegurando que la asignación cumpla con los requisitos establecidos por la institución. Para lograrlo, es necesario definir y gestionar un conjunto de restricciones que regulen la validez y calidad del horario generado. Estas restricciones se dividen en dos categorías: restricciones estrictas (hard constraints), que deben cumplirse obligatoriamente, y restricciones flexibles (soft constraints), cuya violación es posible pero conlleva penalizaciones en la calidad de la solución. A continuación, se ilustran algunos ejemplos de restricciones estrictas y flexibles aplicadas al problema de programación horaria.

Restricciones estrictas:

1. Un docente $p \in P$ no puede impartir más de un evento en un mismo bloque $b \in B$.

2. En cada aula $a \in A$ y bloque horarios $b \in B$ solo puede programarse un evento.

3. Se debe respetar toda preasignación establecida.

4. Un evento con duración menor o igual a tres horas $(E_3 \subseteq E)$ debe programarse en un solo bloque.

5. Un evento de cuatro horas o más $(E_4 \subseteq E)$ puede programarse en dos o tres bloques horarios consecutivos.

6. La carga académica asignada a un docente $p \in P$ debe estar comprendida entre los valores mínimo (CRm_p) y máximo (CRM_p) permitidos.

7. La partición de los estudiantes que se asignarán a los eventos de un mismo curso C_j deberá corresponder al total estimado de estudiantes que se matricularán en dicho curso.

8. La capacidad del aula $a \in A$ debe ser suficiente para el evento asignado.

Restricciones flexibles:

9. Los eventos de un curso C_j no deben coincidir en el mismo bloque horario.

10. Si un docente imparte clases en bloques consecutivos, las aulas asignadas deberán estar cercanas entre sí para y disminuir tiempos de traslado.

11. La carga académica de un docente debe programarse en bloques horarios consecutivos cuando sea posible.

En términos generales, la clasificación entre restricciones estrictas y flexibles depende en gran medida de las políticas y necesidades de cada institución, de modo que una condición considerada preferencia en un contexto puede transformarse en requisito obligatorio en otro. Sin embargo, toda programación horaria debe cumplir con un conjunto de condiciones mínimas, tales como asegurar que un docente no esté asignado a más de un evento en el mismo bloque horario, y que un aula no este asignada simultáneamente por dos eventos distintos.

3.4. Formulación matemática de las restricciones

En esta sección se muestra cómo las restricciones presentadas en la sección (3.3) pueden expresarse mediante sumatorias y notación matemática, utilizando los parámetros y conjuntos definidos previamente. Este paso resulta fundamental, ya que permite formular el problema de manera estructurada y posibilita su resolución a través de algoritmos de optimización.

Restricciones estrictas:

1. Un docente $p \in P$ no puede impartir más de un evento en un mismo bloque $b \in B$.

$$\sum_{e_{ij} \in E} \sum_{a \in A} x_{e_{ij},p,a,b} \leq 1, \quad \forall p \in P, b \in B. \tag{1}$$

   Explicación:
   La sumatoria recorre todos los cursos j, sus respectivos eventos i y los salones $a \in A$, contabilizando cuántos eventos han sido asignados simultáneamente al docente $p \in P$ en el bloque $b \in B$.

   - Si la sumatoria es 0, significa que el docente p no tiene asignado ningún evento en ese bloque.
   - Si la sumatoria es 1, significa que el docente p imparte exactamente un evento en ese bloque, lo cual es factible.
   - Si la sumatoria es mayor que 1, significa que el docente p fue asignado a dos o más eventos en el mismo bloque, lo cual viola la restricción y hace inviable la solución.

2. En cada aula $a \in A$ y bloque $b \in B$ solo puede programarse un evento.

$$\sum_{e_{ij} \in E} \sum_{p \in P} x_{e_{ij},p,a,b} \leq 1, \quad \forall a \in A, b \in B. \tag{2}$$

Explicación:
La sumatoria recorre todos los cursos j, sus respectivos eventos i y los docentes $p \in P$, contabilizando cuántos eventos han sido asignados simultáneamente en el aula $a \in A$ durante el bloque $b \in B$.

- Si la sumatoria es 0, significa que en el aula a no se programó ningún evento en el bloque b.
- Si la sumatoria es 1, significa que exactamente un evento fue programado en el aula a durante el bloque b, lo cual es factible.
- Si la sumatoria es mayor que 1, significa que dos o más eventos fueron programados en el mismo aula a en el bloque b, lo cual viola la restricción y hace inviable la solución.

3. Se debe respetar toda preasignación establecida.

$$\sum_{a\in A}\sum_{b\in B} x_{e_{ij},p,a,b} = 1, \quad \forall e_{ij} \in E_{pre},\ \forall p \in P \text{ preasignado al evento } e_{ij}.$$

Explicación:
La sumatoria recorre todas las aulas $a \in A$ y los bloques $b \in B$, contabilizando las posibles asignaciones de un evento $e_{ij} \in E_{pre}$ que ya tiene un docente $p \in P$ preasignado.

- Si la sumatoria es 0, significaría que el evento preasignado no fue ubicado en ningún aula ni bloque, lo cual viola la restricción.
- Si la sumatoria es 1, significa que el evento e_{ij} fue asignado exactamente al docente y bloque definidos por la preasignación, cumpliendo con la condición requerida.
- Si la sumatoria es mayor que 1, implicaría que el mismo evento fue asignado en más de un aula o bloque, lo cual contradice la unicidad de la preasignación y hace inviable la solución.

4. Un evento con duración menor o igual a tres horas ($E_3 \subseteq E$) debe programarse en un solo bloque.

$$\sum_{p\in P}\sum_{a\in A}\sum_{b\in B} x_{e_{ij},p,a,b} \leq 1, \quad \forall e_{ij} \in E_3. \tag{3}$$

Explicación:
La sumatoria recorre todos los docentes $p \in P$, las aulas $a \in A$ y los bloques $b \in B$, contabilizando cuántas veces un evento $e_{ij} \in E_3$ ha sido asignado.

- Si la sumatoria es 1, significa que el evento e_{ij} fue programado en un único bloque, cumpliendo con la condición de duración.
- Si la sumatoria es mayor que 1, significa que el mismo evento fue asignado en dos o más bloques, lo cual no es permitido ya que su duración cabe en un solo bloque.

De manera análoga, un evento de cuatro horas o más ($E_4 \subseteq E$) puede programarse en dos o tres bloques horarios consecutivos.

$$\sum_{p\in P}\sum_{a\in A}\sum_{b\in B} x_{e_{ij},p,a,b} \leq 3, \quad \forall e_{ij} \in E_4.$$

5. La carga académica asignada a un docente $p \in P$ debe estar comprendida entre los valores mínimo (CRm_p) y máximo (CRM_p) permitidos.

$$CRm_p \leq \sum_{e_{ij}\in E}\sum_{a\in A}\sum_{b\in B} C_{ij} \cdot x_{e_{ij},p,a,b} \leq CRM_p, \quad \forall p \in P. \tag{4}$$

Explicación:

La sumatoria recorre todos los cursos j, sus eventos i, las aulas $a \in A$ y los bloques $b \in B$, acumulando la carga académica C_{ij} correspondiente a los eventos asignados al docente $p \in P$.

- Si la carga académica asignada está entre CRm_p y CRM_p, significa que la dedicación del docente p está dentro del rango permitido, cumpliendo con la condición.
- Si la carga académica asignada al docente p es menor que CRm_p o mayor que CRM_p, significa que su dedicación está fuera del rango permitido, lo cual viola la restricción y hace inviable la solución.

6. La partición de los estudiantes que se asignarán a los eventos de un mismo curso C_j deberá corresponder al total estimado de estudiantes que se matricularán en dicho curso.

$$\sum_{i=1}^{m} CE_{ij} = CC_j, \quad \forall C_j \in C. \tag{5}$$

7. La capacidad del aula $a \in A$ debe ser suficiente para el evento asignado.

$$Cm_a \leq CE_{ij} \cdot x_{e_{ij},p,a,b} \leq CM_A, \quad \forall e_{ij} \in E, a \in A, p \in P, b \in B. \tag{6}$$

Restricciones flexibles:

8. Los eventos de un curso C_j no deben coincidir en el mismo bloque horario.

$$\sum_{i=1}^{m} \sum_{p \in P} \sum_{a \in A} x_{e_{ij},p,a,b} \leq 1, \quad \forall b \in B, C_j \in C. \tag{7}$$

Explicación:
La sumatoria recorre todos los eventos i asociados al curso $C_j \in C$, verificando en qué aula $a \in A$ y con qué docente $p \in P$ se asignan en un bloque $b \in B$. La restricción busca que, de forma deseable, no se asignen dos o más eventos del mismo curso en un mismo bloque horario.

- Si la sumatoria es menor o igual a 1, la condición se cumple y no se aplica penalización.
- Si la sumatoria es mayor que 1, significa que dos o más eventos del curso C_j coinciden en el mismo bloque. En este caso, la solución sigue siendo factible, pero se aplica una penalización en la función objetivo, reduciendo la calidad del horario generado.

9. Si un docente imparte clases en bloques consecutivos, las aulas asignadas deberán estar cercanas entre sí para y disminuir tiempos de traslado.

$$d_{rk} \left[x_{e_{ij},p,a_r,b} + x_{e_{ij},p,a_k,(b+1)} \right] \leq 2T_p, \quad \forall p \in P, b \in B, a_r \neq a_k. \tag{8}$$

Explicación:
La desigualdad considera que si un docente $p \in P$ imparte clases en dos bloques consecutivos b y $b+1$, los salones a_r y a_k asignados deben estar cercanos entre sí.

- Si las aulas asignadas están suficientemente cerca, es decir, d_{rk} no supera el límite establecido, la condición se cumple y no se aplica penalización.
- Si las aulas asignadas están demasiado alejadas, la condición se incumple. En este caso, la solución sigue siendo factible, pero se aplica una penalización en la función objetivo, ya que el docente tendría dificultades para trasladarse entre clases consecutivas.

10. La carga académica de un docente debe programarse en bloques horarios consecutivos cuando sea posible.
Considere la variable binaria $y_{pb} = 1$ si el docente imparte clase en el bloque b y $y_{pb} = 0$ en otro caso.

$$\sum_{b \in B \setminus \{b_{t-1}\}} |y_{pb} - y_{p(b+1)}| \leq \tau_p, \qquad \forall p \in P, \tag{9}$$

donde τ_p denota el límite máximo de espacios libres permitidos entre bloques consecutivos de cada docente $p \in P$.

Explicación:
La sumatoria contabiliza los cambios de estado entre bloques consecutivos del docente p, es decir, las transiciones entre impartir clase y no impartir clase.

- Si la suma de transiciones no supera el valor máximo τ_p, significa que las clases del docente p se han programado de manera compacta, favoreciendo bloques consecutivos y reduciendo espacios libres.
- Si la suma excede τ_p, se interpreta que el docente tiene un horario con demasiados espacios libres.

En resumen, las restricciones estrictas (1-7) delimitan el espacio de soluciones factibles (8-10), mientras que las restricciones flexibles orientan al modelo hacia soluciones de mayor calidad.

3.5. Ejemplo ilustrativo de verificación de restricciones

Con el fin de mostrar de manera práctica cómo se aplican las restricciones en un problema de programación horaria, a continuación se desarrolla un ejemplo ilustrativo. El objetivo no es formular un caso real completo, sino ilustrar cómo una asignación puede verificarse paso a paso para determinar si cumple con las restricciones estrictas y flexibles definidas previamente. Se presentará primero una asignación que resulta factible y, posteriormente, un caso en el que la restricción (7) falla, lo que obliga a realizar un ajuste en la asignación.

- **Definición de conjuntos:**
Supongamos que $E_4 = E_{pre} = \varnothing$ y $E_3 = E$

$$C = \{C_1, C_3\}, \quad E = \{e_{11}, e_{21}, e_{13}, e_{23}\}, \quad P = \{1, 2\}, \quad A = \{1, 2, 3\}, \quad B = \{1, 2\}.$$

CC_j representa el total estimado de estudiantes que se matricularán en el curso $C_j \in C$

CE_{ij} corresponde a la cantidad de estudiantes asignados al evento $e_{ij} \in E$ por el algoritmo.

- **Parámetros y variable de decisión:**
Estimación del total de estudiantes que se matricularán y partición de eventos de acuerdo con el número previsto de estudiantes:

$$CC_1 = 120, \ (CE_{11} = 90, \ CE_{21} = 30); \qquad CC_3 = 60, \ (CE_{13} = 30, \ CE_{23} = 30).$$

Carga académica por evento: $C_{11} = 6, C_{21} = 2, C_{13} = 2, C_{23} = 2$.
Límites de carga académica por docente: $CRm_p = 2, CRM_p = 8$, para $1, 2$.

- **Capacidades de aulas:**
$(Cm_1, CM_1) = (20, 100), (Cm_2, CM_2) = (20, 40), (Cm_3, CM_3) = (20, 40)$.

- **Variable de decisión:**

$$x_{e_{ij},p,a,b} = \begin{cases} 1 & \text{si el evento } e_{ij} \text{ está asignado al docente } p \in P \text{ en el aula } a \in A \text{ durante el bloque } b \in B \\ 0 & \text{en otro caso} \end{cases}$$

  Ejemplo de asignación de horaria (todas las demás $x_{e_{ij},p,a,b} = 0$).

$$x_{e_{11},1,1,1} = 1, \quad x_{e_{13},1,2,2} = 1, \quad x_{e_{21},2,2,1} = 1, \quad x_{e_{23},2,1,2} = 1.$$

  A continuación, se verifica que la asignación propuesta satisface las restricciones definidas en la sección anterior.

- **Restricción de exclusividad de docente por bloque, dada en (1).**

$$\begin{aligned}
(1,1):\ & x_{e_{11},1,1,1} + 0 + x_{e_{11},1,2,1} + \ldots + 0 = x_{e_{11},1,1,1} + 0 + \ldots + 0 = 1 \leq 1 \\
(1,2):\ & x_{e_{13},1,2,2} + 0 + x_{e_{11}112} + \ldots + 0 = 1 \leq 1 \\
(2,1):\ & x_{e_{21},2,2,1} + 0 + x_{e_{23}211} + \ldots + 0 = 1 \leq 1 \\
(2,2):\ & x_{e_{23},2,1,2} + 0 + x_{e_{21},2,2,2} + \ldots + 0 = 1 \leq 1
\end{aligned}$$

- **Restricción de exclusividad de aula por bloque, dada en (2).**

$$\begin{aligned}
(1,1):\ & x_{e_{11},1,1,1} + 0 + x_{e_{21},2,1,1} + \ldots + 0 = 1 \leq 1 \\
(2,1):\ & x_{e_{21},2,2,1} + 0 + x_{e_{11},1,2,1} + \ldots + 0 = 1 \leq 1 \\
(1,2):\ & x_{e_{23},2,1,2} + 0 + x_{e_{13},1,1,2} + \ldots + 0 = 1 \leq 1 \\
(2,2):\ & x_{e_{13},1,2,2} + 0 + x_{e_{23},2,2,2} + \ldots + 0 = 1 \leq 1
\end{aligned}$$

- **Restricción de duración de eventos, dada en (4).**

$$\begin{aligned}
e_{11}:\ & x_{e_{11},1,1,1} + 0 + x_{e_{11},1,2,1} + \ldots + 0 = 1 \leq 1 \\
e_{21}:\ & x_{e_{21},2,2,1} + 0 + x_{e_{21},1,1,1} + \ldots + 0 = 1 \leq 1 \\
e_{13}:\ & x_{e_{13},1,2,2} + 0 + x_{e_{13},1,1,1} + \ldots + 0 = 1 \leq 1 \\
e_{23}:\ & x_{e_{23},2,1,2} + 0 + x_{e_{23},1,2,1} + \ldots + 0 = 1 \leq 1
\end{aligned}$$

- **Restricción de carga académica por docente, dada en (5).**

$$\begin{aligned}
1:\ & C_{11} \cdot x_{e_{11},1,1,1} + C_{13} \cdot x_{e_{13},1,2,2} + \ldots + 0 = C_{11} \cdot 1 + C_{13} \cdot 1 + \ldots + 0 = 6 + 2 = 8 \\
2:\ & C_{21} \cdot x_{e_{21},2,2,1} + C_{23} \cdot x_{e_{23},2,1,2} + \ldots + 0 = C_{21} \cdot 1 + C_{23} \cdot 1 + \ldots + 0 = 2 + 2 = 4
\end{aligned}$$

- **Restricción de partición de estudiantes, dada en (6).**

$$\begin{aligned}
C_1:\ & CE_{11} + CE_{21} = 90 + 30 = 120 = CC_1 \\
C_3:\ & CE_{13} + CE_{23} = 30 + 30 = 60 = CC_3
\end{aligned}$$

- **Restricción de capacidad de aula, dada en (7).**

$$\begin{aligned}
(e_{11} \to 1):\ & 20 \leq CE_{11} \cdot x_{e_{11},1,1,1} = 90 \cdot 1 = 90 \leq 100 \\
(e_{21} \to 2):\ & 20 \leq CE_{21} \cdot x_{e_{21},2,2,1} = 30 \cdot 1 = 30 \leq 40 \\
(e_{13} \to 2):\ & 20 \leq CE_{13} \cdot x_{e_{13},1,2,1} = 30 \cdot 1 = 30 \leq 40 \\
(e_{23} \to 1):\ & 20 \leq CE_{23} \cdot x_{e_{23},2,1,1} = 30 \cdot 1 = 30 \leq 100
\end{aligned}$$

Con el fin de contrastar el ejemplo factible anterior, se presentan a continuación tres escenarios de incumplimiento de restricciones: uno asociado a una restricción estricta y dos vinculados con restricciones flexibles. Estos casos permiten ilustrar cómo, en el primer caso, la solución se vuelve inviable, mientras que en los dos restantes la solución sigue siendo factible aunque con una penalización en la función objetivo.

- **Incumplimiento de restricción estricta, dada en (7).**
  Si (contrariamente a la asignación factible) se asigna e_{11} con $CE_{11} = 90$ al aula 2 (cuya capacidad máxima es $CM_2 = 40$), entonces

  $$Cm_2 \ \leq \ CE_{11} \cdot x_{e_{11},1,2,1} \ \leq \ CM_2 \Rightarrow 20 \leq 90 \cdot 1 \leq 40 \Rightarrow 20 \leq 90 \leq 40,$$

  lo que es falso porque $90 > 40$. Por tanto, la solución es no es factible.

- **Incumplimiento de restricción flexible, dada en (8).**
  Si se prefiere evitar que los eventos de un curso coincidan en un mismo bloque, con la asignación original los dos eventos de C_1 quedaron en 1

  $$\sum_{i=1}^{m}\sum_{p=1}^{2}\sum_{a=1}^{3} x_{e_{ij},p,a,1} = x_{e_{11},1,1,1} + x_{e_{21},2,2,1} + 0 + \ldots + 0 = 2 > 1.$$

  $$\sum_{i=1}^{m}\sum_{p=1}^{2}\sum_{a=1}^{3} x_{e_{ij},p,a,1} = x_{e_{11}112} + x_{e_{21},2,2,2} + 0 + \ldots + 0 = 2 > 1.$$

- **Incumplimiento de restricción flexible, dada en (9).**
  Si se modela como preferencia que un docente no cambie de aula entre bloques consecutivos, puede penalizarse:

  $$d_{12}\left[x_{e_{11},1,1,1} + x_{e_{13},1,2,2}\right] \ \leq \ 2T_1.$$

  Por ejemplo, si $d_{12} = 30$ y $T_1 = 20$, entonces

  $$30\,(1+1) = 60 > 40 = 2T_1,$$

Este incumplimiento de las restricciones flexibles, no afecta la factibilidad del modelo, pero se incorpora como penalización en la función objetivo.

4. Funciones objetivo

En la programación horaria, la elección de la función objetivo desempeña un papel fundamental en la calidad de la solución obtenida. Dependiendo de los objetivos institucionales y operativos, la función objetivo puede priorizar diferentes criterios, como la asignación equitativa de la carga docente, la minimización de espacios libres en los horarios de los docentes, la optimización del uso de las aulas, o la reducción de traslados entre aulas o sedes. Definir una función objetivo adecuada permite dirigir el modelo hacia soluciones que no solo sean factibles, sino también eficientes y alineadas con las necesidades de la institución.

En esta sección, se presentan ejemplos de funciones objetivo que buscan minimizar costos operativos, equilibrar la distribución de carga académica, reducir conflictos de asignación y maximizar la satisfacción de docentes o estudiantes. Cada una de estas funciones responde a distintos enfoques de optimización, lo que permite adaptar el modelo a los requerimientos específicos de cada institución. Los siguientes ejemplos ilustran distintas funciones objetivo utilizadas en la programación horaria:

1. Minimizar el número de cambios de aula para los docentes:

   $$\text{mín} \sum_{p\in P}\sum_{b\in B}\sum_{a_r,a_k\in A, s\neq s'} d_{rk} \cdot x_{e_{ij},p,a_r,b} \cdot x_{e_{ij},p,a_k,(b+1)}.$$

Restricción Asociada:

$$\sum_{a_r,a_k \in A, a_r \neq a_k} d_{rk} \cdot x_{e_{ij},p,a_r,b} \cdot x_{e_{ij},p,a_k,(b+1)} \leq T_p, \quad \forall p \in P.$$

Esta función objetivo busca minimizar la cantidad de cambios de aula que experimentan los docentes en su jornada. Cuando un docente debe trasladarse entre aulas distantes, se incrementa el tiempo de desplazamiento, lo que puede afectar la eficiencia de la enseñanza y la comodidad del docente.

2. Minimizar el uso de aulas para optimizar el espacio:

$$\min \sum_{a \in A} \sum_{b \in B} z_{ab},$$

donde z_{ab} es una variable binaria que indica si el aula $a \in A$ ha sido utilizado en al menos un bloque horario $b \in B$.

Restricción Asociada:

$$z_{ab} \geq x_{e_{ij},p,a,b}, \quad \forall e_{ij} \in E, p \in P, a \in A, b \in B.$$

Esta función objetivo busca minimizar la cantidad de aulas utilizados en la programación horaria con el fin de optimizar el uso del espacio disponible. Reducir la cantidad de aulas utilizados ayuda a mejorar la logística de asignación de aulas, reducir costos operativos y evitar la dispersión innecesaria de los cursos.

3. Minimizar los intervalos libres en la jornada de los docentes:

$$\min \sum_{p \in P} \sum_{b \in B \setminus \{b_{t-1}\}} |y_{pb} - y_{p(b+1)}|,$$

donde $y_{pb} = 1$ si el docente imparte clase en el bloque b y $y_{pb} = 0$ en otro caso.

Restricción Asociada:

$$\sum_{b \in B \setminus \{b_{t-1}\}} |y_{pb} - y_{p(b+1)}| \leq \tau_p$$

Esta función objetivo busca minimizar la cantidad de intervalos libres en el horario de los docentes. Reducir estas interrupciones mejora la eficiencia del horario, optimiza el uso del tiempo de los docentes y reduce tiempos libres en su jornada.

4. Maximizar la preferencias de los docentes:

$$\max \left(PrefP - w_3 \sum_{j \in C} \sum_{b \in B} u_{jb} \right), \tag{10}$$

donde

$$PrefP = \sum_{e_{ij} \in E} \sum_{p \in P} \sum_{a \in A} \sum_{b \in B} (w_1 Pre\_curso_{p,j} + w_2 Pre\_bloque_{p,b})\, x_{e_{ij},p,a,b},$$

y la variable auxiliar $u_{jb} \geq 0$ se define como:

$$u_{jb} \geq S_{jb} - 1, \text{ con } S_{jb} = \sum_{i=1}^{m} \sum_{p \in P} \sum_{a \in A} x_{e_{ij},p,a,b}, \quad \forall C_j \in C, b \in B.$$

Aquí:

- $Pre\_curso_{p,j}$: indica la preferencia del docente $p \in P$ por impartir el curso C_j. Su valor está normalizado en el intervalo $[0, 1]$, donde 0 significa que el docente no desea impartir dicho curso y 1 representa la máxima preferencia.
- $Pre\_bloque_{p,b}$: indica la preferencia del docente $p \in P$ por impartir clases en el bloque $b \in B$. Este parámetro también toma valores en el rango $[0, 1]$. Un valor cercano a 1 indica una alta disponibilidad del docente por impartir clases en ese bloque horario, mientras que valores cercanos a 0 reflejan baja disponibilidad.
- $w_1, w_2, w_3 > 0$: son pesos que permiten ajustar la importancia relativa entre preferencias y penalizaciones. Estos pesos permiten ajustar el modelo según las prioridades de la institución.
- u_{jb}: mide el exceso de eventos del curso $C_j \in C$ en el bloque $b \in B$ por encima del primero permitido.
  a) Si $S_{jb} \leq 1$, entonces $u_{jb} = 0$ (no hay exceso).
  b) Si $S_{jb} > 1$, entonces u_{jb} mide cuántos eventos adicionales se programaron simultáneamente en ese bloque.

La función objetivo maximiza la preferencia de los docentes respecto a los cursos y bloques y es penalizada cada vez que exista un evento superpuesto del mismo curso, es decir, el objetivo es encontrar un equilibrio entre maximizar la satisfacción de los docentes y minimizar los eventos superpuestos.

4.1. Ejemplos ilustrativos en la evaluación de la función objetivo

En esta sección se analizan tres casos a manera de ejemplos: caso 1, alta preferencia docente y penalización media; caso 2, baja preferencia docente y penalización alta; y caso 3, mejora por reasignación de un evento. Asimismo, se destaca cómo los pesos y las superposiciones influyen en el valor final de la función objetivo.

Para ilustrar estos escenarios, en las siguientes subsecciones se emplean las asignaciones definidas en el Ejemplo 3.5 y la función objetivo presentada en la Ecuación 11. La Tabla 1 resume la información relativa a las asignaciones y a las preferencias de los docentes respecto de un curso específico y un bloque horario.

Docente	Evento	$Pre\_curso_{p,j}$	$Pre\_bloque_{p,b}$	Carga C_{ij}	Bloque asignado
1	e_{11}	1.00	1.00	6	1
2	e_{21}	0.70	0.80	2	1
1	e_{13}	0.80	0.90	2	2
2	e_{23}	1.00	1.00	2	2

Tabla 1: Asignación horaria y preferencia de docentes

4.1.1. Caso 1: alta preferencia docente y penalización media

En este ejemplo se asignan los pesos $w_1 = w_2 = 1$ y $w_3 = 0.5$. Dichos valores evidencian que la institución da la máxima importancia a las preferencias docentes, tanto en la asignación de cursos como en la de bloques horarios, mientras que otorga una importancia moderada a la reducción de superposiciones entre los curso C_j.

Cálculo de $PrefP$.

$$\begin{aligned} PrefP_2 &= (Pre\_curso_{1,1} + Pre\_bloque_{1,1})\, x_{e_{11},1,1,1} + (Pre\_curso_{2,1} + Pre\_bloque_{2,2})\, x_{e_{21},2,2,2} \\ &\quad + (Pre\_curso_{1,3} + Pre\_bloque_{1,2})\, x_{e_{13},1,2,2} + (Pre\_curso_{2,3} + Pre\_bloque_{2,2})\, x_{e_{2,3},2,1,2} \\ &= (1+1)\cdot 1 + (0.7+0.8)\cdot 1 + (0.8+0.9)\cdot 1 + (1+1)\cdot 1 \\ &= 7.2 \end{aligned}$$

Cálculo de u_{jb}.

$$S_{1,1} = \sum_{i=1}^{2}\sum_{p=1}^{2}\sum_{a=1}^{3} x_{e_{i1},p,s1} = 2 \Rightarrow u_{1,1} = 1, \quad S_{3,2} = \sum_{i=1}^{2}\sum_{p=1}^{2}\sum_{a=1}^{3} x_{e_{i3},p,s,2} = 2 \Rightarrow u_{3,2} = 1$$

Entonces $\sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 2 \Rightarrow w_3 \sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 0.5 \cdot 2 = 1$

Cálculo del valor de la función objetivo 11.

$$\text{máx}(7.2 - 1) = 6.2$$

El valor de la función objetivo evidencia un aprovechamiento significativo de las preferencias docentes; sin embargo, persiste un costo asociado a la superposición de eventos que disminuye levemente el resultado final. Este ejemplo ilustra cómo el proceso de planificación prioriza la satisfacción del docenteado, aun tolerando ciertos choques horarios.

4.1.2. Caso 2: baja preferencia docente y penalización alta

En este ejemplo se asignan los pesos $w_1 = w_2 = 0.3$ y $w_3 = 0.9$. Dichos valores evidencian que la institución concede una importancia relativamente baja a las preferencias de los docentes, tanto en lo referente a la asignación de cursos como a los bloques horarios, mientras que otorga una relevancia mayor a la reducción de superposiciones entre los cursos C_j.

Cálculo de $PrefP$

$$\begin{aligned} PrefP &= (0.3\cdot 1 + 0.3\cdot 0.8) + (0.3\cdot 0.7 + 0.3\cdot 1) \\ &\quad + (0.3\cdot 0.8 + 0.3\cdot 0.9) + (0.3\cdot 1 + 0.3\cdot 1) \\ &= 2.16 \end{aligned}$$

Cálculo de u_{jb}

Con $w_3 = 0.9$, se obtiene $\sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 2 \Rightarrow w_3 \sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 0.9 \cdot 2 = 1.8$

Cálculo del valor de la función objetivo 11

$$\text{máx}(2.16 - 1.8) = 0.36$$

En contraste con lo observado en el Ejemplo 4.1.1, a reducción de los pesos asignados a las preferencias y el incremento del peso de la penalización conducen a una disminución considerable en el valor de la función objetivo. Este resultado evidencia un escenario extremo, en el que las asignaciones se tornan poco favorables, dado que el modelo otorga mayor prioridad a la eliminación de superposiciones.

4.1.3. Caso 3: mejora por reasignación de un evento

En el Ejemplo 4.1.1, se modifica la asignación del evento e_{21} al bloque 2 y al aula 3, mientras que los demás eventos se mantienen sin cambios.

Al realizar este cambio, la variable de exceso u_{jb} modifica su valor.

$$S_{1,1} = 1, S_{1,2} = 1 \Rightarrow u_{1,1} = u_{1,2} = 0, \qquad S_{3,2} = 2 \Rightarrow u_{3,2} = 1,$$

Entonces $\sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 1 \Rightarrow w_3 \sum_{j=1}^{2}\sum_{b=1}^{2} u_{jb} = 0.5 \cdot 1 = 0.5.$

Cálculo del valor de la función objetivo 11

$$\text{máx}(7.4 - 0.5) = 6.9.$$

En contraste con lo observado en el Ejemplo 4.1.1, al mantener los pesos correspondientes al caso de preferencia alta y penalización media, pero reubicar el evento e_{21} en el bloque 2, la superposición disminuye de dos a uno. Así, la reducción de conflictos genera una mejora inmediata en el valor de la función objetivo.

En conclusión, los ejemplos desarrollados permiten ilustrar de manera didáctica cómo la programación de horarios puede modelarse mediante funciones objetivo que combinan preferencias y penalizaciones. Más allá de los detalles técnicos, el análisis muestra que el equilibrio entre los criterios considerados y pequeños ajustes en las asignaciones pueden mejorar significativamente los resultados.

5. Conclusiones

Este artículo ha presentado un enfoque introductorio a la programación horaria educativa, destacando la importancia de la definición de conjuntos, parámetros, restricciones y funciones objetivo. Se han ilustrado diferentes criterios de optimización y su impacto en la calidad de los horarios generados.

La formulación matemática presentada permite que los problemas de asignación de horarios puedan ser modelados de manera estructurada, facilitando su solución mediante algoritmos de optimización. Además, los ejemplos ilustrativos proporcionan una base comprensible para que nuevos investigadores y profesionales puedan familiarizarse con los conceptos clave en este campo.

Finalmente, este trabajo puede constituir un punto de partida para una implementación básica. Además, la aplicación de un modelo de programación horaria contribuye de manera sustancial a mejorar la equidad en la asignación de recursos y la satisfacción de docentes y estudiantes, de acuerdo con los criterios de relevancia establecidos por la institución.

Agradecimientos: Agradezco al Dr. Byron Jiménez Oviedo por sus valiosos comentarios y sugerencias, así como al arbitraje anónimo por las observaciones que enriquecieron este trabajo.

Contribución de las personas autoras: Este trabajo fue elaborado únicamente por José G. Córdoba Hernández, quien asumió la responsabilidad de todas las etapas de desarrollo del artículo.

Accesibilidad de datos: No aplica.

Referencias

Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A review of optimization algorithms for university timetable scheduling. *Engineering, Technology and Applied Science Research*, *10*(6), 6410-6417. https://doi.org/10.48084/etasr.3832

Bashab, A., Ibrahim, A. O., Hashem, I. A. T., Aggarwal, K., Mukhlif, F., Ghaleb, F. A., & Abdelmaboud, A. (2023). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. *Computers, Materials & Continua*, *74*(3), 6462-6484. https://doi.org/10.32604/cmc.2023.034051

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, *140*(2), 266-280. https://doi.org/10.1016/S0377-2217(02)00069-3

Bustos, B., Barceló, P., & Paredes, R. (2014). *Generación de horarios académicos en INACAP utilizando algoritmos genéticos* [Tesis de maestría, Universidad de Chile].

Canseco, A., Sánchez, D., Zuñiga, C., & Olivares, E. (2016). Aplicación de programación lineal para la asignación de horarios en una institución educativa mexicana. *Revista Ingeniería Industrial*, *15*(2), 135-146. https://revistas.ubiobio.cl/index.php/RI/article/view/2779

Cleger Tamayo, S., Pérez Campaña, M., & Rodríguez Expósito, F. (2007). Alternativa para el proceso de planificación de horarios docentes de una Universidad. *Ciencias Holguín*, *13*(4), 1-8. http://www.redalyc.org/articulo.oa?id=181517998006

Córdoba, J. (2018). *Modelo Matemático para la Programación de Horarios, De Cursos Universitarios, Aplicando una Técnica Metaheurística* [Tesis de maestría, Universidad de Puerto Rico, Recinto de Mayagüez].

Dünke, F., & Nickel, S. (2023). A matheuristic for multi-level multi-criteria university timetabling with student individualization. *Annals of Operations Research*. https://doi.org/10.1007/s10479-023-05325-2

Even, S., Itai, A., & Shamir, A. (1976). On the complexity of time table and multi-commodity flow problems. *SIAM Journal on Computing*, *5*(4), 691-703. https://doi.org/10.1137/0205048

Floyd, R. W. (1967). Nondeterministic Algorithms. *Journal of the ACM*, *14*(4), 636-644. https://doi.org/10.1145/321420.321422

González, L. A., Román, M. A., & Cossio, E. G. (2020). Programación y asignación de horarios: una revisión sistemática de literatura. *Memorias del Congreso Internacional de Investigación Academia Journals*, *12*(6), 440-446. https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/427/1/Programacion%20y%20asignacion%20de%20horarios.pdf

Guerra Cubillos, M. A., Pardo Quiroga, E. H., & Salas Ruiz, R. E. (2013). Problema del School Timetabling y algoritmos genéticos: una revisión. *Revista Vínculos*, *10*(2), 259-276. https://revistas.udistrital.edu.co/index.php/vinculos/article/view/6478/8014

Hernández, R., Miranda, J., & Rey, P. A. (2008). Programación de horarios de clases y asignación de salas para la Facultad de Ingeniería de la Universidad Diego Portales mediante un enfoque de programación entera. *Revista Ingeniería de Sistemas*, *22*, 121-141. https://www.dii.uchile.cl/~ris/RISXXII/horariosUDP_RISVersion%20FINAL.pdf

López-Cruz, O. (2015). Una solución basada en agentes al problema de generación de horarios. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, *2*(3), 73-85. https://ojs.urepublicana.edu.co/index.php/ingenieria/article/view/240/220

Mejía, J. M., & Paternina, C. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Revista Educación en Ingeniería*, *9*, 140-149. https://educacioneningenieria.org/index.php/edi/article/view/15/14

Restrepo, G. E., & Moreno, L. F. (2011). Modelo para la asignación de recursos académicos en instituciones educativas utilizando la técnica metaheurística, búsqueda tabú. *Revista Avances en Sistemas e Informática*, *8*(3), 111-124. https://www.redalyc.org/articulo.oa?id=133122679014

Rezaeipanah, A., Matoori, S. S., & Ahmadi, G. (2020). A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. *Applied Intelligence*, *50*(12), 4482-4502. https://doi.org/10.1007/s10489-020-01833-x

Saltos, R., & Benavides, L. (2021). Formulación e implementación de un modelo de programación entera para la creación de horarios de clases: Un caso de estudio en Ecuador. *Revista digital Matemática, Educación e Internet*, *22*(1). https://doi.org/10.18845/rdmei.v22i1.5734

Siew, E. S. K., Sze, S. N., Goh, S. L., Kendall, G., Sabar, N. R., & Abdullah, S. (2024). A survey of solution methodologies for exam timetabling problems. *IEEE Access*, *12*, 41479-41498. https://doi.org/10.1109/ACCESS.2024.3378054