

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
VALPARAÍSO - CHILE



“UN ALGORITMO PARA LA PLANIFICACIÓN DE HORARIOS DE  
AYUDANTÍA UNIVERSITARIA”

CAMILA ANDREA ARANCIBIA FAÚNDEZ

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERA CIVIL EN INFORMÁTICA

Profesor Guía: María Cristina Riff Rojas  
Profesor Correferente: Pedro Godoy

Marzo - 2026



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción):  Memoria o trabajo de título  Tesis de Postgrado

Título del trabajo: Un algoritmo para la planificación de horarios de ayudantía universitaria.

Nombre del candidato(a): Camila Andrea Arancibia Faúndez

Carrera / Grado: Ingeniería Civil Informática

Campus: Casa Central Departamento: Informática

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, María Cristina Riff Rojas, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

- 6 meses  12 meses  2 años  3 años  5 años  10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

---

---

---

### 4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 19/03/2026

Firma: 

Estudiante o Candidato(a):

Fecha: 19/03/2026

Firma: 

*Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.*

## **DEDICATORIA**

A mi familia.

## **AGRADECIMIENTOS**

Quisiera comenzar agradeciendo a la profesora María Cristina Riff por proponer el tema y brindarme la oportunidad de desarrollar esta memoria; al profesor Claudio Torres, por permitirme interrumpir su clase para obtener los datos necesarios; y a los estudiantes y ayudantes que respondieron las encuestas, cuyas respuestas fueron fundamentales para este trabajo.

A mi padre, por darme la posibilidad de estudiar; a mi tía, por haberme acogido durante los primeros años de carrera; a mi familia en general, por su apoyo constante; y a Ada, por su compañía incondicional durante las largas jornadas de trabajo, siempre presente junto a mi escritorio, incluso en las frías mañanas.

Por último, agradezco a mi pareja, por leer incontables veces este trabajo, obligarme a descansar en las noches y por acompañarme y apoyarme a lo largo de todos estos años. Cada uno de ellos ha sido parte esencial de este proceso, y estoy profundamente agradecida con cada uno.

## RESUMEN

**Resumen**— El presente trabajo busca analizar el problema de asignación de horarios de ayudantía en la Universidad Técnica Federico Santa María (UTFSM). Para esto se evaluarán diversas estrategias de búsqueda con el objetivo de encontrar una solución de calidad a la asignación de horarios, considerando este problema como una alternativa específica al *Educational Timetabling Problem*. En el modelo se busca maximizar la cantidad de estudiantes que pueden asistir a un bloque definido, priorizando aquellos que optimicen la distribución horaria de los estudiantes a través de penalizaciones centradas en la preferencia estudiantil.

**Keywords**— *Educational Timetabling Problem; Algoritmos y técnicas de búsqueda.*

## ABSTRACT

**Abstract**— *The objective of this study is to analyze the problem of scheduling teaching assistant sessions at the Universidad Técnica Federico Santa María (UTFSM). To address this, various search strategies will be evaluated in order to find a high-quality solution for the schedule assignment, considering this problem as a specific variant of the Educational Timetabling Problem. The proposed model seeks to maximize the number of students who can attend a given time block, prioritizing those that optimize student's overall schedule distribution through penalty mechanisms based on student preferences..*

**Keywords**— *Educational Timetabling Problem; Search algorithms and techniques.*

## GLOSARIO

AE: Algoritmo Evolutivo.

BnB: *Branch and Bound*.

CBC: *Coin Branch and Cut*.

CSOP: *Constraint Satisfaction Optimization Problem*.

CSP: *Constraint Satisfaction Problem*.

ETP: *Examination Timetabling Problem*.

IFS: *Iterative Forward Search*.

ILP: *Integer Linear Programming*.

ITC: *International Timetabling Competition*.

MAC: *Maintaining Arc Consistency*.

MILP: *Mixed-Integer Linear Programming*.

SA: *Simulated Annealing*.

STP: *School Timetabling Problem*.

TS: *Tabu Search*.

UCTP: *University Course Timetabling Problem*.

UFRN: *Universidade Federal do Rio Grande do Norte*.

UNITIME: *University Timetabling*.

UTFSM: *Universidad Técnica Federico Santa María*.

## ÍNDICE DE CONTENIDOS

<b>RESUMEN.....</b>	<b>5</b>
<b>ABSTRACT.....</b>	<b>5</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>10</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>11</b>
<b>INTRODUCCIÓN.....</b>	<b>12</b>
<b>CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA.....</b>	<b>14</b>
1.1 CONTEXTO INSTITUCIONAL DE LAS AYUDANTÍAS EN LA UTFSM.....	14
1.2 SISTEMA DE BLOQUES HORARIOS INSTITUCIONALES.....	15
1.3 PROCESO ACTUAL DE ASIGNACIÓN DE HORARIOS DE AYUDANTÍA.....	16
1.4 ACTORES INVOLUCRADOS EN EL PROBLEMA.....	17
1.5 DESCRIPCIÓN GENERAL DEL PROBLEMA DE ASIGNACIÓN DE HORARIOS.....	18
1.6 CRITERIOS Y RESTRICCIONES DEL PROBLEMA.....	18
1.6.1 RESTRICCIONES DEL PROBLEMA.....	19
1.6.2 CRITERIOS DE EVALUACIÓN.....	19
1.7 MOTIVACIÓN Y CONSECUENCIAS DE LA PROBLEMÁTICA.....	19
<b>CAPÍTULO 2: MARCO CONCEPTUAL.....</b>	<b>22</b>
2.1 CONCEPTOS PREVIOS.....	22
2.1.1 CONSTRAINTS SATISFACTION PROBLEM.....	23
2.1.2 PROGRAMACIÓN LINEAL ENTERA.....	24
2.1.3 BRANCH AND BOUND.....	25
2.1.4 CUTTING PLANES.....	26
2.1.5 HILL CLIMBING.....	26
2.1.6 TABU SEARCH.....	27
2.1.7 SIMULATED ANNEALING.....	27
2.1.8 ALGORITMO EVOLUTIVO.....	28
2.2 SOLVERS.....	28
2.2.1 UNITIME.....	28
2.2.2 SOLVER CBC (COIN-OR BRANCH AND CUT).....	30
2.3 EDUCATIONAL TIMETABLING PROBLEM.....	30
2.4 REVISIÓN DE LA LITERATURA.....	31
2.4.1 SOLVING THE ACADEMIC TIMETABLE PROBLEM THINKING ON STUDENT NEEDS [13].....	31
2.4.2 SIMULATED ANNEALING WITH PENALIZATION FOR UNIVERSITY COURSE TIMETABLING [14].....	32
<b>CAPÍTULO 3: PROPUESTA DE SOLUCIÓN.....</b>	<b>35</b>

3.1 FORMULACIÓN MATEMÁTICA.....	35
3.1.1 PARÁMETROS.....	35
3.1.2 VARIABLES.....	36
3.1.3 DOMINIO.....	36
3.1.4 RESTRICCIONES.....	36
3.1.5 FUNCIÓN OBJETIVO.....	38
3.2 REPRESENTACIÓN.....	39
3.2.1 ESTRUCTURA DE SOLUCIÓN (MATRIZ $X_{ijk}$ ).....	39
3.3 SIMULATED ANNEALING.....	39
3.3.1 SOLUCIÓN INICIAL (ALGORITMO GREEDY).....	40
3.3.2 OPERADORES DE VECINDARIO.....	40
3.3.3 PROCESO DE EVALUACIÓN Y ACEPTACIÓN.....	43
3.3.4 CRITERIOS DE PARADA.....	44
3.3.5 PSEUDOCÓDIGO.....	45
3.3.6 CALIBRACIÓN DE PARÁMETROS.....	46
3.3.7 CALIBRACIÓN DE LA FUNCIÓN FITNESS.....	50
3.3.8 CONFIGURACIÓN FINAL SELECCIONADA.....	51
3.4 SOLVER.....	52
3.4.1 VARIABLES.....	53
3.4.2 FUNCIÓN OBJETIVO.....	55
3.4.3 RESTRICCIONES.....	55
3.4.4 PROCESO DE RESOLUCIÓN.....	56
<b>CAPÍTULO 4: VALIDACIÓN DE SOLUCIÓN.....</b>	<b>57</b>
4.1 DATOS.....	57
4.2 PROCESO DE EVALUACIÓN.....	63
4.3 RESULTADOS EXPERIMENTALES.....	63
4.3.1 DESEMPEÑO DE SIMULATED ANNEALING CON AYUDANTÍAS PREVIAMENTE ASIGNADAS.....	63
4.3.2 DESEMPEÑO DE SOLVER CON AYUDANTÍAS PREVIAMENTE ASIGNADAS.....	66
4.3.3 ANÁLISIS DE VARIABILIDAD DE ALGORITMOS.....	70
4.4 CASO DE ESTUDIO.....	74
4.5 DISCUSIÓN DE RESULTADOS.....	80
<b>CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>82</b>
5.1 CONCLUSIONES.....	82
5.2 TRABAJO FUTURO.....	84
5.2.1 SISTEMA MULTIPROPÓSITO.....	84
5.2.2 EVOLUCIÓN HACIA UN MODELO MULTI-ASIGNATURA.....	85

5.2.3 SISTEMA DE INFORMACIÓN HÍBRIDO.....	86
5.3 REFLEXIÓN FINAL.....	86
<b>BIBLIOGRAFÍA.....</b>	<b>88</b>

## ÍNDICE DE FIGURAS

Figura 1: Árbol del problema.....	20
Figura 2: Óptimos locales y globales en R2.....	23
Figura 3: Ilustración de la secuencia del algoritmo Branch and Bound.....	25
Figura 4: Horario en UniTime.....	29
Figura 5: Representación de la solución.....	39
Figura 6: Solución inicial.....	40
Figura 7: Movimiento por día.....	41
Figura 8: Movimiento por bloque.....	42
Figura 9: Movimiento swap.....	43
Figura 10: Pseudocódigo Simulated Annealing.....	45
Figura 11: Ejemplo de horarios de estudiantes.....	58
Figura 12: Disponibilidad total por bloque.....	58
Figura 13: Gráficos de porcentaje de ocupación.....	59
Figura 14: Horario distribuido a los estudiantes y ayudantes.....	60
Figura 15: Disponibilidad total por bloque INF285 extraoficial.....	60
Figura 16: Disponibilidad total por bloque INF285 oficial.....	61
Figura 17: Gráfica comparativa de %Fitness Baseline v/s SA.....	65
Figura 18: Gráfica comparativa de conflictos Baseline v/s SA.....	66
Figura 19: Gráfica comparativa de %Fitness Baseline v/s Solver.....	68
Figura 20: Gráfica comparativa de conflictos Baseline v/s Solver.....	69
Figura 21: Brecha porcentual (Gap %) de SA respecto al solver por instancia.....	72
Figura 22: Fitness promedio y desviación estándar de SA por instancia.....	73
Figura 23: Comparación de tiempos promedio de ejecución entre SA y Solver.....	74
Figura 24: Gráficas de análisis de Baseline, INF295.....	75
Figura 25: Gráficas de análisis de Simulated Annealing, INF295.....	77
Figura 26: Gráficas de análisis de Solver, INF295.....	79

## ÍNDICE DE TABLAS

Tabla 1: Bloques de horarios de la UTFSM.....	15
Tabla 2: Comparación entre soluciones (manual vs. algoritmo).....	32
Tabla 3: Configuraciones iniciales del primer experimento.....	46
Tabla 4: Comparación de resultados, experimento 1.....	47
Tabla 5: Configuraciones iniciales del segundo experimento.....	48
Tabla 6: Comparación de resultados, experimento 2.....	48
Tabla 7: Configuraciones iniciales del tercer experimento.....	49
Tabla 8: Comparación de resultados, experimento 3.....	49
Tabla 9: Ponderación de restricciones blandas.....	50
Tabla 10: Mejor configuración encontrada.....	51
Tabla 11: Mejores resultados por test.....	51
Tabla 12: Comparación de ayudantías previamente asignadas.....	62
Tabla 13: Comparación de ayudantías previamente asignadas.....	64
Tabla 14: Comparación de Baseline v/s solver.....	67
Tabla 15: Comparación de calidad de solución SA v/s solver.....	71
Tabla 16: Comparación de tiempos de ejecución SA v/s solver.....	73

## INTRODUCCIÓN

La asignación de horarios en contextos educativos constituye un problema relevante dentro de la gestión académica, debido a la necesidad de coordinar múltiples recursos, restricciones y preferencias, tanto institucionales como humanas. En cuanto a la planificación de actividades complementarias, como ayudantías, laboratorios, etc., desempeñando un rol fundamental en el apoyo al aprendizaje de los estudiantes, debido a que estas instancias permiten reforzar contenidos, resolver dudas y mejorar el rendimiento académico. Sin embargo, la efectividad depende en gran medida de una correcta asignación horaria que favorezca la asistencia y el aprovechamiento de estos espacios.

De esta manera, el presente trabajo de memoria aborda el problema de asignación de horarios de ayudantía en trece asignaturas universitarias, considerando la disponibilidad de estudiantes y ayudantes, así como un conjunto de restricciones y criterios que influyen directamente en la calidad de la solución. Además de centrarse en una asignatura en específico para analizar la distribución de la calidad de la solución entre los estudiantes.

Este problema puede ser modelado como una variante del *Educational Timetabling Problem*, el cual pertenece a la familia de problemas de optimización combinatoria caracterizados por su alta complejidad y por la dificultad de encontrar soluciones óptimas mediante métodos exactos en tiempos razonables. Es por esto que se propone un enfoque de resolución basado en técnicas de optimización combinatoria, utilizando tanto el uso de un *solver* como de una metaheurística, con el objetivo de obtener asignaciones horarias de buena calidad que mejoren la asistencia a las ayudantías y optimicen el uso de los recursos disponibles. La propuesta no busca únicamente generar una solución factible, sino también analizar el desempeño de los distintos enfoques considerados, evaluando aspectos como la calidad de las soluciones obtenidas, el tiempo de ejecución y su adaptabilidad al problema planteado.

Desde el punto de vista metodológico, el desarrollo de la memoria contempla una etapa de modelación formal del problema, seguida por la implementación de los algoritmos seleccionados y la definición de métricas de evaluación que permitan comparar los resultados. Para validar la propuesta, se utilizan datos reales correspondientes a asignaturas impartidas en la Universidad Técnica Federico Santa María (UTFSM), lo que permite analizar el comportamiento de las soluciones en un escenario representativo del ambiente académico.

La estructura de esta memoria se divide en cinco capítulos que se explican a continuación:

- **Capítulo 1:** Presenta la definición del problema, describiendo el contexto institucional, el proceso actual, los actores involucrados y los criterios y restricciones relevantes.
- **Capítulo 2:** Introduce el marco conceptual, donde se revisan los principales conceptos teóricos relacionados con la optimización y el *Educational Timetabling Problem*. Además de mostrar algunas de las diferentes metaheurísticas y *solvers*.

- **Capítulo 3:** Se detalla la modelación del problema y las estrategias de solución propuestas. Además de los diferentes experimentos realizados para la calibración de los parámetros.
- **Capítulo 4:** Muestra los experimentos realizados, validando la solución propuesta y analizando los resultados obtenidos.
- **Capítulo 5:** Se exponen las conclusiones del trabajo, sintetizando los principales aportes metodológicos y experimentales de la investigación. Además, se analiza críticamente el desempeño de los enfoques propuestos desde una perspectiva técnica y práctica, se discuten sus implicancias para la planificación académica real y se proponen líneas de trabajo futuro orientadas a ampliar el alcance y la aplicabilidad del modelo.

## CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

Como primer capítulo se tiene como objetivo describir y delimitar el problema de asignación de horarios de ayudantía en el contexto de educación superior, específicamente en la UTFSM. Se busca contextualizar la relevancia de las ayudantías dentro del proceso formativo de los estudiantes y evidenciar las dificultades prácticas asociadas a la definición de sus horarios, las cuales motivan la necesidad de un enfoque sistemático y computacional para su resolución.

La correcta asignación de horarios de ayudantía constituye un elemento clave para el apoyo académico de los estudiantes, ya que impacta directamente en la asistencia, el aprovechamiento de las instancias de refuerzo y la equidad en el acceso a estos espacios. Sin embargo, este proceso se ve condicionado por múltiples factores, tales como la disponibilidad de los ayudantes, la diversidad de horarios académicos de los estudiantes, las restricciones institucionales y consideraciones asociadas al tiempo de traslado y seguridad dentro del campus.

En este contexto, el problema de asignación de horarios de ayudantía puede entenderse como un problema de decisión con múltiples restricciones y objetivos potencialmente conflictivos, donde no siempre es posible satisfacer de manera simultánea todas las preferencias de los actores involucrados. La complejidad de este escenario aumenta a medida que crece la cantidad de estudiantes, asignaturas y combinaciones horarias posibles, dificultando la obtención de soluciones de calidad mediante métodos manuales o heurísticas<sup>1</sup> simples.

### 1.1 CONTEXTO INSTITUCIONAL DE LAS AYUDANTÍAS EN LA UTFSM

En la educación superior las ayudantías constituyen una instancia fundamental de apoyo al proceso de enseñanza, especialmente en asignaturas con alta carga académica. En la UTFSM, el rol del ayudante se encuentra definido en el reglamento institucional, el cual establece que “se considera ayudante a la persona que coopera con el profesor en labores específicas” [1]. Esta definición general abarca diferentes tipos de ayudantías. Sin embargo, para este estudio se considerarán exclusivamente los ayudantes de cátedra, quienes tienen a su cargo la realización de cátedras de apoyo dirigidas a los estudiantes.

Las ayudantías docentes cumplen un rol complementario a las clases regulares que imparte el docente, permitiendo reforzar el contenido, resolver dudas y profundizar en la materia que, por restricciones de tiempo o tamaño de los cursos, no siempre pueden ser abordadas a detalle por el profesor. En este sentido, la efectividad de una ayudantía depende en gran medida de la asistencia estudiantil, la cual está influenciada por el horario en que esta se imparte.

En la UTFSM, las ayudantías pueden realizarse de forma presencial o en modalidad remota, dependiendo de la naturaleza de la asignatura y las decisiones del equipo docente. No obstante, en ambos casos, la definición del horario debe ajustarse a los bloques horarios establecidos institucionalmente, los cuales estructuran la jornada académica de los estudiantes y ayudantes.

---

<sup>1</sup> Una heurística es un método o estrategia que busca obtener una solución aceptable en un tiempo razonable, sin garantizar que sea óptima. Un ejemplo de esto es asignar la ayudantía en el primer bloque en que el ayudante se encuentre disponible.

La elección de un bloque inadecuado puede generar conflictos con otras actividades académicas, desincentivar la asistencia o incluso excluir a un subconjunto significativo de estudiantes.

Dado que los ayudantes suelen ser estudiantes de cursos superiores, su disponibilidad horaria también se ve limitada por su propia carga académica, lo que introduce una nueva dimensión de restricciones al problema. De esta manera, la asignación del horario de ayudantía debe considerar simultáneamente la disponibilidad del ayudante y las posibilidades reales de asistencia de los estudiantes, dentro de las normas establecidas por la institución educacional.

Este escenario demuestra que la definición de horarios de ayudantía no corresponde a una decisión trivial, sino a un problema que requiere equilibrar múltiples factores académicos, operativos y humanos, lo que justifica su análisis formal en este trabajo.

## 1.2 SISTEMA DE BLOQUES HORARIOS INSTITUCIONALES

La planificación académica en la UTFSM se organiza a partir de un sistema de bloques horarios discretos, los cuales estructuran tanto las cátedras como las actividades complementarias, entre ellas las ayudantías. Cada bloque corresponde a un intervalo de tiempo fijo y predefinido, lo que permite estandarizar la asignación de salas, coordinar la disponibilidad de docentes y facilitar la planificación global de horarios académicos.

En la Tabla 1 se presentan los bloques horarios establecidos por la institución, junto con sus respectivos rangos de tiempo. Este sistema divide la jornada académica en pares de bloques consecutivos, que se extienden desde las primeras horas de la mañana hasta la noche. La utilización de bloques discretos, en lugar de horarios continuos, impone una restricción estructural al problema de asignación, ya que limita el conjunto de alternativas posibles a combinaciones específicas definidas previamente por la institución.

Tabla 1: Bloques de horarios de la UTFSM.  
Fuente: Elaboración propia.

Bloque	Horario	Bloque	Horario
1-2	08:15 - 09:25	11-12	16:05 - 17:15
3-4	09:40 - 10:50	13-14	17:30 - 18:40
5-6	11:05 - 12:15	15-16	18:55 - 20:05
7-8	12:30 - 13:40	17-18	20:20 - 21:30
9-10	14:40 - 15:50	19-20	21:40 - 22:50

La existencia de bloques horarios fijos simplifica ciertos aspectos administrativos, aun así también introduce desafíos adicionales desde el punto de vista de la optimización. En particular, no todos los bloques presentan las mismas condiciones para los estudiantes, ya sea por razones de carga académica, tiempos de traslado o factores externos como la iluminación natural y la percepción de seguridad dentro y fuera del campus. Estos elementos hacen que algunos bloques sean menos preferidos que otros, generando una demanda de distribución desigual.

De esta manera, dado que los estudiantes suelen cursar múltiples asignaturas por semestre, la compatibilidad entre los bloques horarios de las cátedras y los posibles bloques de ayudantía resulta crítica. La asignación de una ayudantía en un bloque genera choques horarios o extensos períodos de espera entre clases puede afectar negativamente la asistencia, aun cuando dicho bloque horario sea formalmente válido desde el punto de vista institucional. En consecuencia, el sistema de bloques de horarios mostrados en la Tabla 1 no solo define el dominio de decisión del problema, sino que también condiciona los criterios que deben ser considerados al evaluar la calidad de una asignación.

### **1.3 PROCESO ACTUAL DE ASIGNACIÓN DE HORARIOS DE AYUDANTÍA**

Actualmente, la asignación de horarios de ayudantía en la UTFSM se realiza, en la mayoría de los casos<sup>2</sup>, mediante mecanismos informales, tales como encuestas en línea o consultas presenciales realizadas durante los últimos minutos de una cátedra. Estos métodos buscan recoger la disponibilidad o preferencia horaria de los estudiantes para definir un bloque que resulte conveniente para la mayoría. Sin embargo, este método presenta diversas limitaciones que afectan la representatividad y calidad de la decisión final.

En el caso de las encuestas en línea, la participación suele estar condicionada por el plazo establecido para responderlas, lo que implica que solo una fracción del total de estudiantes expresa su disponibilidad. De forma similar, cuando la definición del horario se realiza de forma presencial, la decisión se basa únicamente en la opinión de los estudiantes que asisten a esa instancia particular, excluyendo a quienes no se encuentran presentes por motivos académicos, personales o logísticos. Como resultado, el conjunto de preferencias considerado no representa necesariamente a la totalidad del curso.

Otra limitación relevante de este proceso es que los estudiantes tienden a seleccionar únicamente los bloques que consideran más convenientes desde una perspectiva individual, sin declarar todas las opciones horarias en las que podrían asistir. Esto introduce un sesgo adicional en la información recopilada, reduciendo artificialmente el espacio de alternativas consideradas y favoreciendo la elección de bloques populares, aun cuando estos puedan generar conflictos o dificultades para un subconjunto significativo de estudiantes.

El proceso actual de asignación puede caracterizarse como una toma de decisiones basada en información parcial y no estructurada, lo que dificulta la obtención de soluciones que equilibren adecuadamente las necesidades de los distintos actores involucrados. Esta situación evidencia la necesidad de abordar el problema mediante un enfoque más sistemático, que permita

---

<sup>2</sup> En algunos casos, la institución asigna los horarios de ayudantía antes de que los estudiantes elijan sus asignaturas.

integrar de manera explícita múltiples criterios y restricciones en la definición del horario de ayudantía.

#### **1.4 ACTORES INVOLUCRADOS EN EL PROBLEMA**

El problema de asignación de horarios de ayudantía involucra a distintos actores, cada uno con necesidades, restricciones y objetivos particulares. La interacción entre estos actores genera un conjunto de interés que, en muchos casos, puede resultar contrapuestos, lo que incrementa la complejidad del problema y refuerza la necesidad de abordarlo de manera sistemática.

- **Estudiantes:** Constituyen el principal grupo beneficiario de las ayudantías, ya que estas instancias están orientadas a reforzar contenidos, resolver dudas y apoyar el proceso de aprendizaje. No obstante, los estudiantes presentan una alta heterogeneidad en sus horarios académicos, carga de asignaturas, años de ingreso y trayectorias curriculares. A esto se suma la diversidad en los tiempos de traslados hacia el campus y las condiciones personales de cada estudiante, factores que influyen directamente en la posibilidad real de asistir a una ayudantía en un bloque determinado. Por lo tanto, un horario que resulta adecuado para un subconjunto del curso, puede ser inviable para otro, lo que plantea el desafío de maximizar la asistencia sin excluir sistemáticamente a ciertos grupos.
- **Ayudantes:** Generalmente son estudiantes de cursos superiores, pero también poseen restricciones horarias propias derivadas de su carga académica, actividades personales y, en algunos casos, compromisos laborales. La disponibilidad del ayudante constituye una restricción fundamental, que limita el conjunto de bloques en los cuales es posible programar la ayudantía. Además, al igual que los estudiantes, los ayudantes pueden verse afectados por ventanas extensas entre actividades o por la asignación de horarios en bloques nocturnos, lo que debe ser considerado para asegurar condiciones razonables de desempeño.
- **Institución:** Establece el marco normativo y operativo en el cual se desarrolla el proceso de asignación. Esto incluye la definición de los bloques horarios, la disponibilidad de salas, las políticas académicas y los reglamentos asociados a las ayudantías. Si bien estas restricciones permiten organizar de manera eficiente la planificación académica global, también imponen límites que reducen la flexibilidad del problema, acotando las alternativas factibles de asignación.

La coexistencia de estos actores y sus respectivas restricciones implica que la definición del horario de ayudantía debe buscar un equilibrio entre la factibilidad operativa y la calidad de la solución desde el punto de vista académico y humano. En consecuencia, el problema no puede abordarse únicamente desde la perspectiva de un actor aislado, sino que requiere integrar de manera explícita las necesidades y limitaciones de los involucrados.

## **1.5 DESCRIPCIÓN GENERAL DEL PROBLEMA DE ASIGNACIÓN DE HORARIOS**

La asignación de un horario semanal para cada ayudantía docente se define por un día y bloque horario institucional, de manera que se logra una solución factible y de calidad para los actores involucrados. Esta decisión debe tomarse considerando un conjunto de restricciones operativas y académicas, así como criterios orientados a maximizar la asistencia estudiantil y mejorar las condiciones en que se desarrollan las ayudantías.

Desde un punto de vista general, el problema consiste en seleccionar, para una determinada asignatura, un bloque horario que sea compatible con la disponibilidad del ayudante y con los horarios académicos de los estudiantes inscritos. Para ello, se dispone de los horarios de cátedra de los estudiantes, los bloques disponibles según la planificación institucional y las restricciones propias del contexto universitario. A partir de estos datos, es necesario evaluar el impacto de cada posible asignación sobre la asistencia potencial y sobre la experiencia académica de los estudiantes.

Es importante destacar que no todas las asignaciones factibles presentan la misma calidad. Si bien un bloque puede cumplir con las restricciones básicas de disponibilidad, este puede generar efectos indeseados, tales como extensos períodos de espera entre clases, la obligación de asistir al campus únicamente para asistir a la ayudantía o la asignación en horarios nocturnos con menor percepción de seguridad. Estos factores no invalidan formalmente una solución, pero sí afectan negativamente su conveniencia desde el punto de vista estudiantil.

Adicionalmente, la diversidad de trayectorias académicas de los estudiantes implica que el problema no puede resolverse satisfaciendo únicamente a un subconjunto del curso. La coexistencia de estudiantes con distintas combinaciones de asignaturas y horarios genera un escenario en el cual las decisiones óptimas para algunos pueden resultar desfavorables para otros, lo que introduce un componente de compromiso inherente al problema. En este contexto, la calidad de una asignación debe evaluarse de manera global, considerando el impacto agregado sobre el conjunto de estudiantes.

En consecuencia, el problema de asignación de horarios de ayudantía puede caracterizarse como un problema de decisión con múltiples criterios y restricciones, en el cual se busca identificar una solución que equilibre factibilidad, conveniencia y equidad. Esta caracterización sienta las bases para su posterior formulación como un problema de optimización, la cual se desarrolla en los capítulos siguientes.

## **1.6 CRITERIOS Y RESTRICCIONES DEL PROBLEMA**

El problema de asignación de horarios de ayudantía se encuentra condicionado por un conjunto de criterios y restricciones que permiten definir tanto la factibilidad como la calidad de las soluciones generadas. Estos elementos cumplen un rol central en la modelación del problema, ya que delimitan el espacio de búsqueda y orientan el proceso de optimización hacia configuraciones deseables desde el punto de vista académico.

### **1.6.1 RESTRICCIONES DEL PROBLEMA**

Las restricciones consideradas pueden clasificarse entre aquellas obligatorias para que una solución se considere válida y otras que corresponden a las preferencias de los estudiantes. Dentro del primer grupo se consideran, entre otras, que el ayudante debe tener el bloque horario libre para poder realizar una ayudantía y que el ayudante solo puede realizar una ayudantía al mismo tiempo. El incumplimiento de cualquier restricción obligatoria implica que la solución se descarta inmediatamente.

Por otro lado, las restricciones relacionadas a las preferencias de los estudiantes, están orientadas a mejorar la calidad de la solución sin comprometer su factibilidad. Estas restricciones se relacionan principalmente con criterios de comodidad y equidad para los estudiantes, tales como la preferencia por concentrar ayudantías en determinados días, la reducción de tiempos muertos entre actividades académicas y evitar jornadas excesivamente extensas. El incumplimiento de estas restricciones no invalida la solución, pero genera penalizaciones que son incorporadas en la evaluación global de la asignación.

### **1.6.2 CRITERIOS DE EVALUACIÓN**

Los criterios de evaluación definen la forma en que se mide la calidad de una solución factible. En este problema en particular, dichos criterios se encuentran directamente asociados al cumplimiento de las restricciones por preferencia de los estudiantes, mediante una función objetivo que penaliza configuraciones no deseables y favorece aquellas que se ajustan de mejor manera a las necesidades de los estudiantes.

En particular, se busca priorizar soluciones que minimicen la asignación de ayudantías en bloques extremos, adyacentes a alguna actividad relacionada a la asignatura y promuevan una distribución más equilibrada de la carga académica. Estos criterios permiten comparar distintas soluciones factibles y seleccionar aquella que presenta un mejor desempeño global bajo el enfoque de optimización propuesto.

La definición clara de los criterios y restricciones del problema resulta fundamental para garantizar la coherencia del modelo y para permitir una correcta interpretación de los resultados obtenidos. Asimismo, estos elementos constituyen la base sobre la cual se construye la función de evaluación utilizada por las técnicas de búsqueda y optimización implementadas.

## **1.7 MOTIVACIÓN Y CONSECUENCIAS DE LA PROBLEMÁTICA**

La planificación de horarios constituye una de las tareas más críticas en el ámbito educativo, dado que impacta directamente en la organización académica, el aprovechamiento de los recursos disponibles y la experiencia de los estudiantes. En particular, la asignación de horarios de ayudantía en la UTFSM representa un desafío, ya que no solo depende de la disponibilidad de los ayudantes, sino también de la heterogeneidad de los horarios de los estudiantes y de factores asociados a la seguridad y los tiempos de traslado hacia el campus.

Tal como se describió en la sección 1.3, el proceso actual de definición de horarios se basa principalmente en encuestas presenciales o en línea, las cuales presentan limitaciones importantes en términos de representatividad. Este enfoque tiende a considerar únicamente a los estudiantes que responden dentro de un plazo acotado o que asisten a una instancia específica, dejando fuera a una fracción significativa del curso. Como consecuencia, los horarios definidos suelen ser poco representativos, lo que se traduce en una baja asistencia a las ayudantías y en una utilización ineficiente de estos espacios de apoyo académico.

Adicionalmente, la falta de un procedimiento sistemático para integrar múltiples criterios en la definición del horario puede generar efectos negativos en el estudiantado. Entre estos efectos se encuentran la asignación de ayudantías en bloques poco convenientes, la generación de extensos tiempos muertos entre actividades académicas y la obligación de desplazarse al campus exclusivamente para asistir a una ayudantía, lo que resulta especialmente relevante en contextos donde existen consideraciones de seguridad asociadas a determinados horarios.

En la Figura 1 se presenta el árbol del problema, elaborado a partir del análisis del proceso actual de asignación de ayudantías en la UTFSM. En este se identifican las principales causas que originan la dificultad de definir un horario representativo, tales como la ausencia de un procedimiento formal, la alta diversidad de disponibilidades horarias de los estudiantes y el limitado horario de los ayudantes. Estas causas convergen en el problema central de asignación ineficiente de horarios de ayudantía, generando efectos como la baja asistencia estudiantil, la falta de priorización de bloques estratégicos y la pérdida de tiempo en coordinaciones poco efectivas.

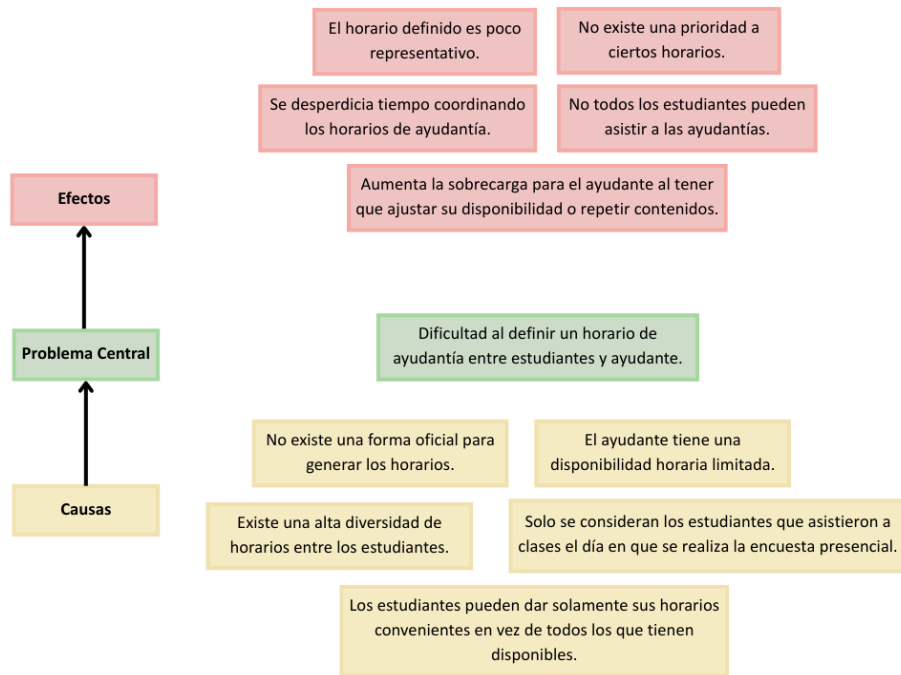


Figura 1: Árbol del problema.  
Fuente: Elaboración propia.

Si esta problemática no es abordada de manera sistemática en el corto o mediano plazo, es esperado que se mantengan las ineficiencias descritas, afectando de forma sostenida la efectividad de las ayudantías como herramienta de apoyo académico. En este contexto, se justifica la necesidad de proponer un enfoque computacional que permita modelar explícitamente el problema, integrar múltiples restricciones y criterios, y apoyar la toma de decisiones mediante técnicas de optimización adecuadas al nivel de complejidad del escenario analizado.

Este trabajo de memoria se orienta a abordar de manera sistemática el problema descrito, estableciendo objetivos claros que permitan guiar el desarrollo y evaluación de la propuesta. Es por esto que para enfrentar este problema se propone analizar el desempeño de una metaheurística seleccionada y el uso de un *solver* especializado con el fin de determinar la técnica más adecuada para maximizar la asistencia estudiantil y optimizar la distribución horaria. Para cumplir este propósito se plantean los siguientes objetivos específicos:

- Modelar el problema.
- Resolver el problema utilizando *solver* y metaheurística seleccionada.
- Evaluar el rendimiento considerando calidad de solución, tiempo de ejecución y adaptabilidad al problema.
- Identificar las condiciones en las que cada algoritmo presenta un mejor desempeño.
- Validar los resultados obtenidos con datos reales de la asignatura INF295, Inteligencia Artificial, de la Universidad Técnica Federico Santa María.

## CAPÍTULO 2: MARCO CONCEPTUAL

Este capítulo revisa los conceptos, técnicas y algoritmos claves utilizados para abordar el *Educational Timetabling Problem*, así como sus principales variantes y enfoques de resolución presentes en la literatura. En estudios similares, es necesario analizar e identificar las metaheurísticas y *solvers* más utilizados para este tipo de problema, con el fin de generar una comprensión sobre la versatilidad, tanto en sus posibles soluciones como en sus variantes.

Para comprender el rol que juegan estos algoritmos, resulta necesario introducir brevemente la noción de optimización y las razones que justifican el uso de metaheurísticas y *solvers*. En ingeniería, la optimización se refiere al proceso de buscar una solución que minimice o maximice una función objetivo, evaluando alternativas dentro de un espacio de búsqueda hasta alcanzar un punto en el cual no es posible mejorar la calidad de la solución mediante los mecanismos definidos [2].

En el contexto del problema propuesto, la optimización se aborda mediante técnicas de optimización combinatoria, incluyendo metaheurísticas y enfoques basados en *solvers*. Estas técnicas corresponden a marcos algorítmicos generales diseñados para guiar procesos de búsqueda en espacios de solución complejos, ya sea mediante exploración heurística del espacio de búsqueda o a través del modelado explícito de restricciones y funciones objetivos.

En particular, las metaheurísticas permiten explorar de manera más amplia regiones prometedoras del espacio de búsqueda, favoreciendo la identificación de soluciones de alta calidad. Aunque pueden aplicarse a una gran variedad de problemas, habitualmente requieren ajustes menores para adaptarse a un dominio particular. La preferencia por el uso de metaheurísticas en problemas como el *Educational Timetabling Problem* se debe a que, al tratarse de un problema *NP-hard*<sup>3</sup>, las buenas soluciones suelen concentrarse en áreas específicas del espacio de búsqueda, lo que hace especialmente relevante el uso de estrategias de exploración balanceada, mecanismos de escape de óptimos locales y modelos de optimización capaces de incorporar restricciones de forma explícita.

### 2.1 CONCEPTOS PREVIOS

Antes de abordar los algoritmos utilizados, es necesario precisar algunos conceptos propios de la optimización combinatoria. El primero de ellos es el espacio de búsqueda, el cual corresponde al conjunto total de soluciones posibles que pueden generarse a partir de las variables y sus dominios. Cada punto de este espacio representa una asignación completa y potencialmente válida del problema. En problemas complejos, el tamaño del espacio crece de forma combinatoria respecto al número de eventos, recursos y restricciones, lo que hace inviable explorar todas las soluciones en un tiempo razonable. Además, este espacio suele presentar una estructura altamente irregular, donde soluciones de distinta calidad pueden encontrarse distribuidas en múltiples regiones. En este contexto, como se puede ver en la Figura

---

<sup>3</sup> Sería un poco extenso explicar en detalle qué significan las clases P, NP, *NP-hard* y NP-completo. Sin embargo, en palabras simples, un problema *NP-hard* es uno cuya dificultad crece muy rápidamente a medida que aumenta su tamaño, y para el cual no se conoce un método eficiente que garantice encontrar siempre la mejor solución en un tiempo razonable.

2, un óptimo local corresponde a una solución cuya calidad es superior a la de todas las soluciones en su vecindario inmediato, mientras que el óptimo global es la mejor solución posible dentro de todo el espacio. La dificultad radica en que los métodos de búsqueda pueden quedar atrapados en óptimos locales, lo que hace necesario el uso de metaheurísticas capaces de equilibrar exploración<sup>4</sup> y explotación<sup>5</sup> para navegar eficientemente este espacio y evitar convergencias prematuras.

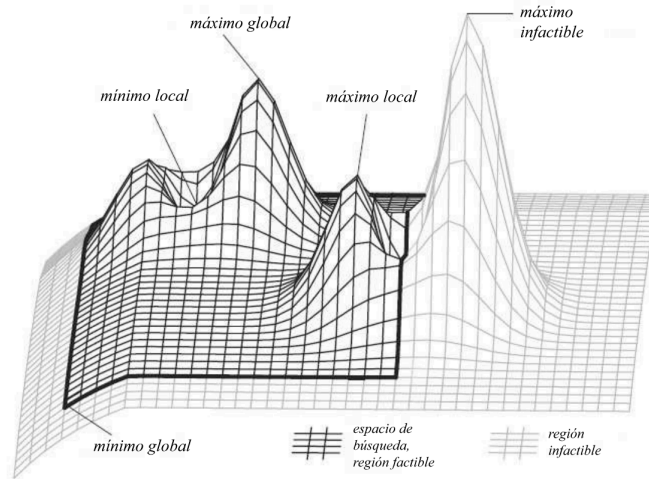


Figura 2: Óptimos locales y globales en  $R^2$ .

Fuente: Optimización Combinatoria, de la teoría a la práctica [2].

Dentro de los enfoques utilizados para resolver problemas de optimización combinatoria se distinguen dos grandes familias; los métodos de búsqueda local y los métodos de búsqueda poblacional. La búsqueda local opera a partir de una única solución inicial, la cual es modificada mediante pequeños cambios o movimientos que generan su vecindario inmediato. A través de un criterio de selección, generalmente basado en la mejora de la función objetivo, se elige una solución del vecindario para continuar iterando. En contraste, la búsqueda poblacional no trabaja sobre una sola solución, sino sobre un conjunto de soluciones simultáneas, o población, cuyos individuos se combinan y modifican mediante operadores como cruce y mutación para producir nuevas generaciones. Mientras la búsqueda local explota de manera intensiva una región específica del espacio, los enfoques poblacionales favorecen una exploración más amplia y diversificada.

### 2.1.1 CONSTRAINTS SATISFACTION PROBLEM

Un problema de satisfacción de restricciones (CSP) es aquel que presenta un conjunto finito de variables  $X = \{x_1, x_2, \dots, x_n\}$ , de las cuales cada variable  $x_i$  posee un dominio asociado  $D_i$  que especifica el valor que puede obtener. La interacción entre estas variables está regulada por un conjunto de restricciones  $C = \{c_1, c_2, \dots, c_m\}$ , las cuales describen las combinaciones de valores

<sup>4</sup> Capacidad del algoritmo para recorrer diversas regiones del espacio de búsqueda, con el objetivo de evitar la convergencia prematura hacia óptimos locales.

<sup>5</sup> Refinamiento y mejora constante de la solución.

permitidas para los subconjuntos de variables. El objetivo de un CSP es encontrar una asignación completa  $X \rightarrow D$  que le asigne a cada variable un valor dentro de su dominio y que satisfaga todas las restricciones definidas [3]. Además, existen los problemas de optimización de satisfacción de restricciones, conocidos como CSOP por sus siglas en inglés, los cuales extienden el modelo clásico de CSP incorporando una función objetivo que asigna un valor numérico a cada solución con el fin de identificar la más óptima entre todas las factibles [4].

En el contexto de *Timetabling* y otros problemas combinatorios, las restricciones suelen clasificarse en duras y blandas. Las restricciones duras representan condiciones que deben cumplirse estrictamente para que una solución sea considerada válida. Por ejemplo, asegurar que una ayudantía se programe únicamente en bloques donde el ayudante pueda impartir la cátedra. Ya que cualquier violación de una restricción dura invalida la solución.

Por otro lado, las restricciones blandas corresponden a preferencias o condiciones deseables cuyo incumplimiento no invalida la solución, pero sí afecta su calidad mediante la función objetivo. Un ejemplo es penalizar la asignación de una ayudantía en un horario demasiado temprano o demasiado tarde para los estudiantes. Esta distinción permite modelar de manera flexible problemas reales en los que no siempre es posible satisfacer simultáneamente todas las preferencias. Al mismo tiempo, habilita la formulación de funciones de evaluación que equilibren la factibilidad con la calidad de la solución, aspecto crucial en problemas de optimización *NP-hard* como el *Educational Timetabling Problem*.

### **2.1.2 PROGRAMACIÓN LINEAL ENTERA**

La programación Lineal Entera (ILP) ha sido ampliamente utilizada para abordar problemas de *Timetabling*, debido a su capacidad para modelar de forma exacta decisiones discretas, restricciones estructurales y criterios de optimización mediante variables enteras, funciones objetivo lineales y un conjunto explícito de restricciones. En este tipo de enfoque, el problema se formula como un modelo matemático en el cual se busca minimizar o maximizar una función objetivo, sujeta a un conjunto de restricciones que definen la factibilidad de la solución.

En el contexto del *Educational Timetabling Problem*, el uso de ILP resulta especialmente adecuado, ya que la mayoría de las decisiones involucradas son de naturaleza binaria o discreta, como la asignación de una ayudantía a un bloque horario, la asignación de un ayudante a una sala o la participación de un estudiante en una instancia determinada. Estas decisiones pueden representarse mediante variables binarias que indican la presencia o ausencia de una asignación específica, permitiendo modelar de forma precisa las condiciones del problema.

Una de las principales ventajas de los enfoques basados en ILP es que corresponden a técnicas completas, lo que implica que, si el modelo está correctamente formulado, el *solver* es capaz de encontrar una solución óptima global o determinar que el problema no posee una solución factible. Esta característica resulta particularmente relevante desde un punto de vista académico, ya que permite establecer soluciones de referencia y evaluar la calidad de otros enfoques aproximados, como las metaheurísticas.

No obstante, esta garantía de optimalidad conlleva un alto costo computacional, dado que el *Educational Timetabling Problem* es de tipo *NP-hard* y el tamaño del espacio de búsqueda crece exponencialmente con el número de variables y restricciones. En consecuencia, los modelos ILP pueden volverse intratables para instancias de gran tamaño o con un elevado número de restricciones, lo que limita su aplicabilidad directa en escenarios reales de gran escala.

### 2.1.3 BRANCH AND BOUND

El método *Branch and Bound* (BnB) es una técnica de optimización exacta ampliamente utilizada para resolver problemas de programación entera y combinatoria. Su objetivo principal es encontrar la solución óptima global explorando de manera sistemática el espacio de soluciones factibles, evitando la enumeración exhaustiva mediante el uso de cotas que permiten descartar regiones no prometedoras.

El procedimiento se basa en la construcción de un árbol de búsqueda, donde cada nodo representa un subproblema obtenido al dividir (*branching*) el problema original mediante la fijación de valores para una o más variables de decisión. Para cada subproblema, se calcula una cota inferior o superior (*bounding*) de la función objetivo. Un ejemplo de esto se puede apreciar a continuación en la Figura 3.

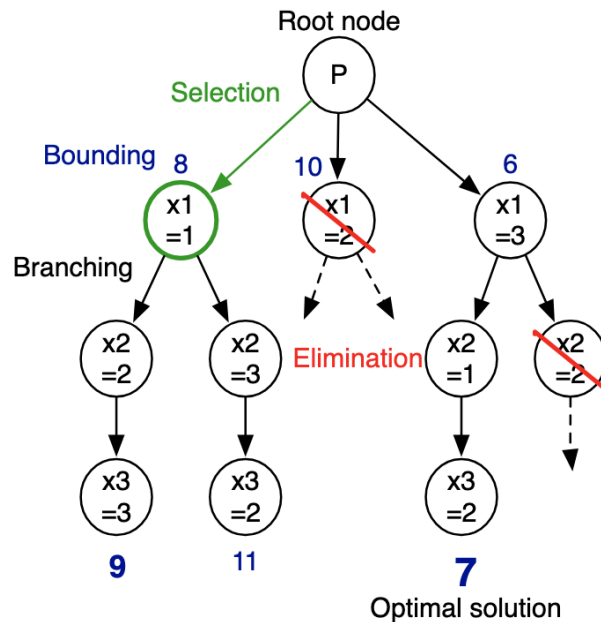


Figura 3: Ilustración de la secuencia del algoritmo Branch and Bound.

Fuente: Parallel hybrid optimization methods for permutation based problems [5].

Si la cota obtenida para un subproblema indica que ninguna de sus soluciones puede mejorar la mejor solución factible encontrada hasta ese momento, dicho nodo es descartado del proceso de búsqueda. Este mecanismo permite reducir significativamente el espacio de búsqueda y garantiza, bajo condiciones adecuadas, la obtención de una solución óptima global.

#### 2.1.4 CUTTING PLANES

La técnica de *Cutting Planes* constituye un complemento fundamental del método BnB en la resolución de problemas de ILP. Su propósito es fortalecer la relajación lineal<sup>6</sup> del problema original mediante la incorporación de restricciones adicionales, denominadas cortes.

El proceso consiste en identificar desigualdades válidas que son satisfechas por todas las soluciones enteras del problema, pero violadas por la solución fraccionaria obtenida al resolver la relajación lineal. Al agregar estos cortes al modelo, se reduce el conjunto factible de la relajación, mejorando las cotas utilizadas en el procedimiento BnB y acelerando la convergencia hacia la solución óptima.

En la práctica, los *solver* modernos de programación entera, combinan dinámicamente la generación de cortes con el proceso de ramificación, dando lugar a esquemas híbridos conocidos como *Branch and Cut*. Esta integración permite abordar instancias de mayor tamaño y complejidad, mejorando tanto la eficiencia computacional como la capacidad del *solver* para manejar restricciones estructuralmente complejas.

#### 2.1.5 HILL CLIMBING

*Hill Climbing* es una metaheurística de búsqueda local de carácter reparador que opera a partir de una solución inicial potencialmente subóptima. En cada iteración, el algoritmo aplica un movimiento, como *swap* o *shift*, para generar un conjunto de soluciones vecinas, conocido como vecindario. Del cual, una vez generado, existen dos estrategias clásicas de selección:

- **Best-improvement:** Genera y evalúa todo el vecindario, para después elegir la mejor solución disponible.
- **First-improvement:** Genera vecinos de manera secuencial y selecciona el primero que produzca una mejora en la función objetivo.

Una vez seleccionada la nueva solución, el procedimiento se repite generando un vecindario actualizado. Para su ejecución, *Hill Climbing* requiere de una representación de la solución, una función de evaluación, una solución inicial y un conjunto de movimientos definidos para explorar el espacio de búsqueda. Su simplicidad de implementación y bajo costo computacional lo convierten en un método eficiente para obtener soluciones aceptables en tiempos reducidos. Sin embargo, su principal limitación es que tiende a quedar atrapado en óptimos locales, debido a que no posee mecanismos de escape y explora el espacio de búsqueda de forma estrictamente ascendente.

---

<sup>6</sup> Permite eliminar temporalmente la restricción de que algunas variables deben tomar valores enteros, permitiendo que adopten valores fraccionarios. Esto transforma el problema en uno más fácil de resolver, aunque la solución obtenida puede no ser válida para el problema original.

### 2.1.6 TABU SEARCH

*Tabu Search* (TS) es una metaheurística de búsqueda local diseñada para superar las limitaciones de métodos como *Hill Climbing*. Su principio fundamental consiste en incorporar mecanismos de memoria que guían la exploración del espacio de búsqueda, permitiendo evitar ciclos y escapar de óptimos locales. Para esto mantiene una lista tabú en la cual se registran temporalmente soluciones recientemente exploradas, impidiendo que el algoritmo regrese a esta solución durante un tiempo determinado de iteraciones. Este mecanismo, junto con criterios como la aspiración<sup>7</sup> y estrategias de intensificación y diversificación, intensifica la probabilidad de que el método explore regiones más amplias y prometedoras del espacio de búsqueda.

La efectividad del algoritmo depende de la adecuada definición de la estructura de la memoria, el tamaño de la lista tabú y de las reglas asociadas a su actualización, aspectos que influyen directamente en su capacidad de equilibrar exploración y explotación. Como señalan Batista y Glover, TS se caracteriza por extender de manera sistemática la búsqueda local mediante el control explícito de la información pasada, mejorando significativamente la calidad de las soluciones en problemas combinatorios complejos [6].

### 2.1.7 SIMULATED ANNEALING

*Simulated Annealing* (SA) es una metaheurística inspirada en el proceso físico de enfriamiento de metales. Al igual que *Hill Climbing*, opera a partir de una solución inicial y aplica pequeños cambios para generar soluciones vecinas. No obstante, implementa un mecanismo probabilístico que facilita la aceptación de soluciones de calidad inferior. Esta probabilidad se define como:

$$P = e^{-\frac{\Delta f}{T}},$$

Donde  $\Delta f$  corresponde al incremento en la función objetivo y  $T$  representa la temperatura actual del sistema. A medida que avanzan las iteraciones, la temperatura se reduce siguiendo un proceso de enfriamiento controlado por un parámetro  $\alpha$ , de modo que la aceptación de peores soluciones se vuelve cada vez menos frecuente. Este comportamiento permite escapar de óptimos locales durante las primeras etapas y, progresivamente, concentrar la búsqueda en regiones prometedoras del espacio de soluciones.

La base teórica de este comportamiento ha sido estudiada formalmente en la literatura, destacando los resultados de convergencia establecidos por Granville, et al. [7], donde se demuestra que, bajo ciertos esquemas de enfriamiento, SA converge hacia el óptimo global. No obstante, la calidad práctica de los resultados depende fuertemente de la calibración de sus parámetros, tales como la temperatura inicial y final, la tasa de enfriamiento y el número de iteraciones, los cuales deben ajustarse cuidadosamente para lograr un equilibrio adecuado entre exploración y explotación, dado que valores altos de temperatura favorecen una

---

<sup>7</sup> Permite aceptar un movimiento tabú si la solución presenta una gran mejora.

búsqueda exploratoria del espacio de soluciones, mientras que valores bajos intensifican el refinamiento local de soluciones prometedoras.

### **2.1.8 ALGORITMO EVOLUTIVO**

Los Algoritmos Evolutivos (AE) son metaheurísticas poblacionales inspiradas en los principios de la evolución natural. Los cuales operan a partir de una población inicial de soluciones candidatas, más conocidas como cromosomas, de las cuales algunas se modifican mediante operadores de variación, como el cruce y la mutación, mientras que otras pasan directamente a la siguiente generación mediante el elitismo, asegurando que los individuos más aptos se preserven sin cambios. Este proceso genera nuevas poblaciones a lo largo de múltiples iteraciones. En cada generación, los individuos son evaluados mediante una función de evaluación, y se les aplica un operador de selección que favorece a aquellos con un mejor desempeño, guiando así el proceso evolutivo hacia regiones prometedoras del espacio de búsqueda.

Este enfoque destaca por su gran capacidad de exploración global, lo que lo hace especialmente adecuado para problemas combinatorios complejos como el *Educational Timetabling Problem*, donde existen múltiples óptimos locales y restricciones. Diversos estudios han aplicado AE a problemas de asignación de horarios, demostrando su eficacia para producir soluciones de alta calidad en entornos con múltiples restricciones y estructuras de solución de alta dimensionalidad, como se evidencia en el estudio de Pascual y Chiazso [8], quienes implementan un algoritmo evolutivo para la asignación de horarios universitarios.

No obstante, su principal limitación radica en el costo computacional, ya que cada generación requiere evaluar múltiples individuos, especialmente cuando la función objetivo es compleja o involucra una gran cantidad de restricciones. Es por esto que la calibración de parámetros como el tamaño de la población, los criterios de reemplazo, las probabilidades de cruce y mutación resultan fundamentales para equilibrar la exploración, explotación y eficiencia computacional.

## **2.2 SOLVERS**

### **2.2.1 UNITIME**

UniTime es un sistema integral de programación académica que facilita la creación de horarios de cursos y exámenes, la gestión de modificaciones, la asignación de salas y la distribución de estudiantes en secciones [9]. Se trata de un *software* de código abierto basado en la librería *CPSolver*, la cual implementa el algoritmo *Iterative Forward Search* (IFS) para resolver problemas de satisfacción de restricciones.

IFS combina técnicas de búsqueda local con los principios de búsqueda sistemática mediante la extensión incremental de asignaciones parciales factibles. Asimismo, incorpora estadísticas basadas en conflictos y el mantenimiento dinámico de la consistencia de arco (MAC), lo que permite reducir eficientemente el espacio de búsqueda y priorizar decisiones con mayor impacto [10]. Este *solver* ha sido validado en diversas competencias internacionales de

*TimeTabling*, demostrando su eficiencia en contextos académicos reales. Destacando por su robustez y su capacidad para manejar restricciones complejas y estructuras de gran escala, típicas de instituciones universitarias.

La Figura 4 muestra un ejemplo de visualización generada por UniTime, donde se aprecia la distribución de cursos a lo largo de la semana, la superposición de actividades y la densidad de eventos por bloque. Esta representación permite dimensionar gráficamente la complejidad operativa del *Educational Timetabling Problem* que el sistema debe resolver.

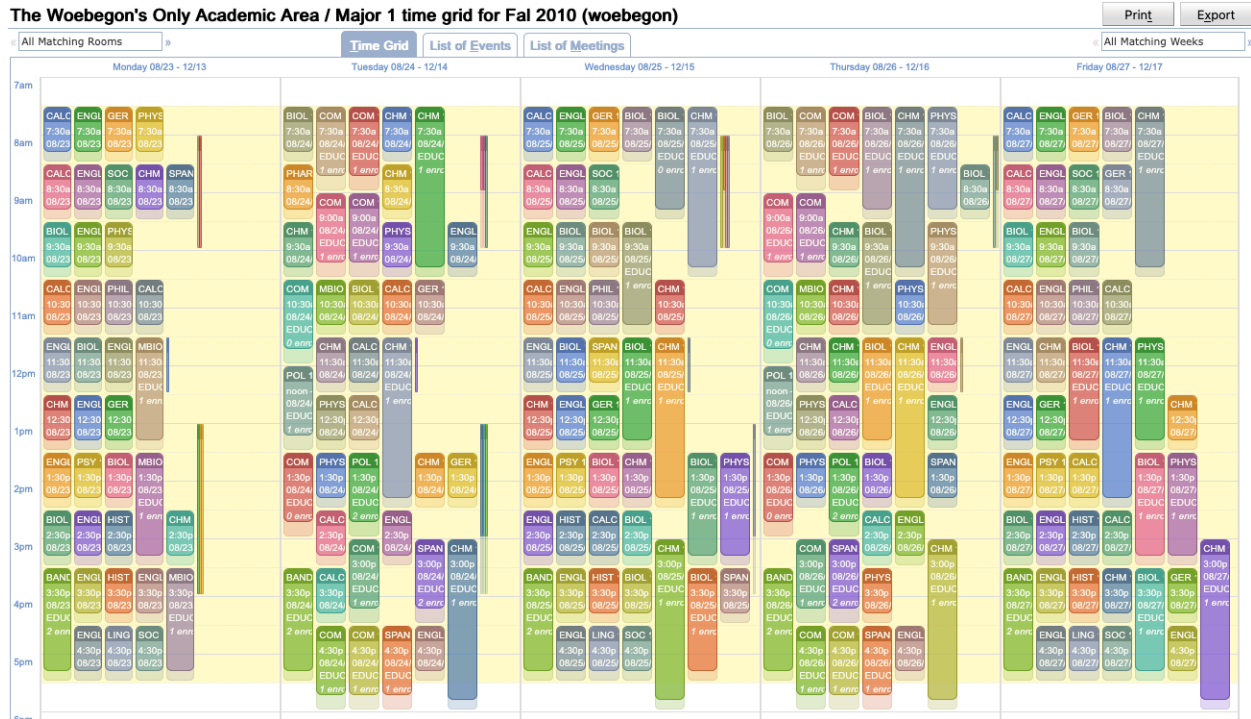


Figura 4: Horario en UniTime.  
Fuente: UniTime Demo

UniTime cuenta con cinco módulos principales, entre los cuales destacan:

- **Course Timetabling Extension:** Para la programación de cursos y actividades asociadas.
- **Student Sectioning Extension:** Orientado a la asignación óptima de estudiantes a secciones.
- **Examination Timetabling Extension:** Para la planificación de exámenes finales.
- **Instructor Scheduling Extension:** Para la asignación de docentes a actividades en función de restricciones individuales e institucionales.

Estos módulos permiten abordar diferentes variantes del *Educational Timetabling Problem*, proporcionando una plataforma flexible y extensible para apoyar procesos académicos reales.

### **2.2.2 SOLVER CBC (COIN-OR BRANCH AND CUT)**

COIN-OR CBC (*Coin-or Branch and Cut*) es un *solver* de código abierto ampliamente utilizado para la resolución de problemas de ILP y Programación Lineal Entera Mixta (MILP)<sup>8</sup>. Forma parte del proyecto *Computational Infrastructure for Operations Research* (COIN-OR) y ha sido empleado en diversos dominios de optimización combinatoria, incluyendo problemas de planificación y asignación de horarios.

CBC implementa un enfoque exacto basado en una combinación de técnicas de *Branch and Bound* y *Cutting Planes*, lo que le permite explorar de manera sistemática el espacio de soluciones enteras definido por el modelo. A través de este esquema híbrido, el *solver* es capaz de generar cotas ajustadas y descartar regiones del espacio de búsqueda que no conducen a soluciones óptimas.

Una de las principales características de CBC es su capacidad de garantizar la obtención de soluciones óptimas globales, siempre que el modelo esté correctamente formulado y el problema sea computacionalmente tratable. No obstante, el desempeño del *solver* se ve condicionado por el tamaño del modelo, el número de variables enteras y la densidad de restricciones, factores que pueden provocar un crecimiento significativo en los tiempos de ejecución en instancias de gran escala.

### **2.3 EDUCATIONAL TIMETABLING PROBLEM**

La asignación de horarios puede considerarse uno de los trabajos más importantes en la planificación del año escolar, tanto para la educación básica, media o superior. El sistema educativo es algo primordial en la actualidad que acompaña a las personas por una parte significativa de sus vidas. Por ejemplo, en Chile se cuenta con un sistema educativo que se divide en educación parvularia, básica, media y superior, aumentando la complejidad del problema con los años. Por lo general, en los niveles de educación básica y media, los estudiantes se dividen en cursos y salas. En otras palabras, cada curso está asignado a una sala y en esta se imparten todas las asignaturas correspondientes a este curso. Para este caso, la complejidad se centra solo en la asignación de horarios.

Por otro lado, la educación superior complejiza el problema eliminando el concepto de curso, brindándole a los alumnos la facultad de elegir las materias que quieran, aunque con ciertas restricciones. Con respecto a las actividades académicas, además de las clases impartidas por el profesor, se consideran las ayudantías, laboratorios y actividades extracurriculares que se asignan después de tomada la asignatura.

Es en este contexto que el *Educational Timetabling Problem* se caracteriza como el desafío de asignar un conjunto de actividades académicas con tiempo, recursos y espacio limitado, respetando las restricciones de su naturaleza. En términos formales, se deben cumplir las restricciones duras como la capacidad de salas, disponibilidad de los profesores y el no

---

<sup>8</sup>A diferencia de ILP, en MILP las variables de decisión pueden ser continuas (reales), enteras o binarias, lo que permite modelar simultáneamente decisiones discretas y cantidades continuas dentro de un mismo problema de optimización.

solapamiento de los estudiantes. Además, se busca optimizar las restricciones blandas, como minimizar los tiempos entre clases y respetar las preferencias horarias.

El primer registro del *Educational Timetabling Problem* se encuentra en el trabajo de Gotlieb en 1963, el cual propuso un modelo para la construcción de horarios de clases y profesores [11]. Con el tiempo el problema ha ido evolucionando, y es durante la década de 1990 que la literatura muestra que se comenzaron a formalizar tres variantes claves de este problema [12].

- ***School Timetabling Problem (STP)***: En un ámbito escolar se realiza la asignación de horarios de todas las asignaturas de la institución, evitando que los profesores tengan dos clases al mismo tiempo y viceversa. Esta variante se entiende como el problema clásico del *Educational Timetabling Problem*.
- ***University Course Timetabling Problem (UCTP)***: Su comportamiento es parecido al STP, pero además de evitar que los profesores tengan dos clases al mismo tiempo, se minimizan los choques de horarios de los estudiantes, es decir, que el alumno tenga dos asignaturas al mismo tiempo. Esto complica el problema inicial, ya que se introducen conflictos al nivel del estudiante.
- ***Examination Timetabling Problem (ETP)***: El ambiente es el mismo que en la variante UCTP, pero además se debe realizar la asignación de horarios para exámenes, en los que se pueden presentar alumnos comunes entre asignaturas. Por lo tanto, la variante ETP debe minimizar la superposición de exámenes de materias y distribuir los exámenes entre sí lo más posible.

## 2.4 REVISIÓN DE LA LITERATURA

### 2.4.1 SOLVING THE ACADEMIC TIMETABLE PROBLEM THINKING ON STUDENT NEEDS [13].

Este estudio aborda el problema de asignación de horarios considerando explícitamente las necesidades de los estudiantes, diferenciándose de la mayoría de los trabajos existentes, que priorizan restricciones institucionales o docentes. Para ello, los autores utilizaron datos reales del programa de Licenciatura en Sistemas Informáticos de la Universidade Federal do Rio Grande do Norte (UFRN), abarcando el periodo desde el primer semestre de 2011 hasta el segundo semestre de 2015.

Para modelar adecuadamente el problema, se entrevistó a coordinadores académicos y a un investigador de pedagogía, definiéndose un conjunto de criterios orientados a mejorar la experiencia del estudiante:

- **Evitar bloques al final del día**: Minimiza las clases en los últimos horarios del día, debido a la dependencia de buses interurbanos con horarios fijos.
- **Reducir tiempos muertos**: Se evita la existencia de intervalos libres entre las clases para prevenir la dispersión de los estudiantes.

- **Evitar clases consecutivas de una misma asignatura:** Para mitigar el cansancio y la pérdida de atención por estar haciendo continuamente una misma actividad.
- **Incorporar preferencias del profesor:** Considerando la disponibilidad y compatibilidad de los docentes, ya que también realizan investigaciones u otras actividades académicas.
- **Corrección de choques tras mutación:** El algoritmo genético repara automáticamente soluciones no factibles generadas por el operador de mutación.

Con base a estos criterios, se diseñó una función de penalización en la cual una solución es mejor mientras menor sea su puntaje. Los autores de la UFRN implementan un algoritmo genético, comparándolo posteriormente con los horarios construidos manualmente por los coordinadores del programa. La Tabla 2 muestra los resultados obtenidos.

Tabla 2: Comparación entre soluciones (manual vs. algoritmo).  
Fuente: Adaptado del trabajo de Almeida et al. [13].

Semestre	Penalización manual	Penalización algoritmo
2012.1	19	0
2012.2	21	3
2013.1	23	0
2013.2	25	2
2014.1	23	2
2014.2	12	1
2015.1	24	1
2015.2	17	17

En la mayoría de los casos, el algoritmo genético produjo soluciones considerablemente mejores que las elaboradas manualmente. La única excepción fue el segundo semestre del año 2015 (2015.2), en el que ambas aproximaciones alcanzaron el mismo nivel de penalización. Los autores concluyen que la incorporación de criterios centrados en el estudiante permite obtener horarios de mayor calidad y con una menor carga cognitiva asociada.

Este estudio resulta relevante para la presente memoria, ya que comparte el objetivo de integrar necesidades estudiantiles a la función objetivo. Asimismo, inspira la inclusión de penalizaciones relacionadas con ventanas horarias, disponibilidad de ayudantes, preferencias docentes y la evitación de bloques pocos seguros o pedagógicamente desfavorables. De esta manera, el trabajo revisado sustenta el enfoque adoptado en esta investigación y evidencia el valor de considerar criterios humanos en el *Academic Timetabling Problem* o inclusive en cualquier problema CSOP que esté relacionado con personas.

#### **2.4.2 SIMULATED ANNEALING WITH PENALIZATION FOR UNIVERSITY COURSE TIMETABLING [14].**

El trabajo de Sylejmani et al. [14] presenta un enfoque basado en la metaheurística de *Simulated Annealing* para resolver una variante compleja del *University Timetabling Problem*, introducida en el marco de la *Fourth International Timetabling Competition* (ITC 2019). Este

problema amplía formulaciones clásicas del UCTP al incorporar nuevos elementos, como la asignación de estudiantes a secciones y restricciones de distribución, tanto obligatorias como opcionales.

La propuesta se estructura en dos fases principales. En la primera etapa, el algoritmo se enfoca en encontrar una solución factible que satisfaga las restricciones duras del problema. Posteriormente, la segunda fase busca mejorar la calidad de la solución obtenida, optimizando las restricciones blandas sin perder la factibilidad alcanzada. Este enfoque permite separar explícitamente el objetivo de factibilidad del de optimización, facilitando la exploración eficiente del espacio de soluciones.

### **FUNCIÓN DE EVALUACIÓN COMPUESTA**

El algoritmo utiliza una función de evaluación denominada *Search Penalty*, la cual combina distintos tipos de penalizaciones en una única métrica. Esta función considera dos casos principales:

- **Solución inviable:** cuando existen violaciones a restricciones duras. En este caso, la función suma la penalización dura ( $p_h$ ) con una versión normalizada de la penalización suave ( $p_s$ ) y las penalizaciones por sobrecupo de estudiantes ( $p_c$ ). Se utiliza un coeficiente  $c_1 = 0.01$  para evitar que el algoritmo pierda demasiado tiempo en micro optimizaciones antes de lograr una solución factible.
- **Solución viable:** cuando  $p_h = 0$ . En este escenario, la función de evaluación prioriza la reducción del sobrecupo de estudiantes y de las penalizaciones suaves, utilizando un coeficiente  $c_2$  suficientemente grande para asegurar que la capacidad de las aulas sea el principal criterio de optimización.

Este cambio dinámico en la función de evaluación permite adaptar el comportamiento del algoritmo según el estado de la solución durante el proceso de búsqueda.

### **PENALIZACIÓN GRADUAL ADAPTATIVA**

Con el fin de evitar el estancamiento en óptimos locales, el *solver* incorpora un mecanismo de penalización gradual y adaptativa. Cuando el algoritmo no presenta mejoras tras un número determinado de iteraciones, se identifican las asignaciones específicas responsables de los conflictos persistentes. Estas combinaciones reciben una penalización adicional almacenada en una estructura interna, incentivando al algoritmo a explorar nuevas configuraciones en iteraciones posteriores.

Paralelamente, las penalizaciones asociadas a asignaciones que no están siendo utilizadas se reducen progresivamente, permitiendo que estas alternativas vuelvan a ser consideradas en etapas futuras del proceso de búsqueda. Este mecanismo actúa como una estrategia de diversificación controlada.

## EVALUACIÓN ENFOCADA EN RESTRICCIONES DIFÍCILES

Adicionalmente, cuando ciertas restricciones obligatorias persisten sin cumplirse durante varios ciclos, el algoritmo entra en una fase de búsqueda enfocada. En esta etapa se utiliza una función de evaluación modificada, en la cual las penalizaciones asociadas a las restricciones problemáticas dominan el cálculo de la *Search Penalty*. De este modo, el algoritmo prioriza explícitamente la resolución de estos conflictos por sobre la mejora de la puntuación asociada a restricciones blandas.

Gracias a la combinación de estas estrategias adaptativas, el *solver* logra encontrar soluciones factibles para todos los escenarios evaluados en la competencia, obteniendo resultados competitivos frente a otros enfoques presentados en ITC 2019. Este trabajo demuestra que el uso de mecanismos de penalización dinámicos y funciones de evaluación adaptativas permite abordar de manera efectiva problemas de *Timetabling* universitario con múltiples restricciones simultáneas, justificando el uso de *Simulated Annealing* como una alternativa robusta para este tipo de problemas.

En síntesis, el *Educational Timetabling Problem* constituye un problema de optimización combinatoria de alta complejidad, caracterizado por la coexistencia de restricciones duras y blandas y por un espacio de búsqueda de crecimiento exponencial. La literatura revisada evidencia que tanto los enfoques exactos basados en ILP como las metaheurísticas han demostrado ser estrategias efectivas para abordar este tipo de problemas, cada una con ventajas y limitaciones en términos de optimalidad, escalabilidad y flexibilidad.

## CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

Como se explicó anteriormente, el objetivo principal consiste en asignar horarios de ayudantías que resulten adecuados para los estudiantes, considerando sus respectivas disponibilidades horarias. En este contexto, se permite la realización de múltiples ayudantías por asignatura, cuya cantidad depende directamente del número de ayudantes disponibles, con el fin de maximizar la cantidad de estudiantes que pueden asistir a la actividad.

Con el propósito de obtener una representación general y consistente del problema, los horarios son modelados utilizando bloques de tiempo discretos, lo que facilita su formulación matemática y posterior resolución computacional.

### 3.1 FORMULACIÓN MATEMÁTICA

Cada estudiante y ayudante cuenta con un horario de clases previamente definido, el cual determina los bloques de tiempo en los que se encuentra ocupado. A partir de esta información, se busca determinar la asignación de horarios de ayudantías que maximice la participación estudiantil, respetando las restricciones de disponibilidad de estudiantes y ayudantes, así como las restricciones institucionales del problema. A continuación se describen los parámetros y estructuras de datos utilizadas para modelar el problema.

#### 3.1.1 PARÁMETROS

- Matriz de tres dimensiones que almacena los horarios de todos los estudiantes:

- $E_{ijk} \in \{0, 1, 2\}$ : Estado del estudiante  $k$  en el día  $j$  y bloque  $i$ , definido como:

$$E_{ijk} = \begin{cases} 0 & \text{si está libre} \\ 1 & \text{si tiene una actividad} \\ 2 & \text{si tiene una actividad y está relacionada a la asignatura} \end{cases}$$

- Matriz de tres dimensiones que almacena los horarios de los ayudantes:

- $A_{ijk} \in \{0, 1\}$ : Estado del ayudante  $k$  en el día  $j$  y bloque  $i$ , definido como:

$$A_{ijk} = \begin{cases} 0 & \text{si está libre} \\ 1 & \text{si tiene otra actividad} \end{cases}$$

- Matriz de dos dimensiones que almacena los bloques prohibidos, en los que la universidad no permite asignar alguna ayudantía:

- $F_{ij} \in \{0, 1\}$ : El horario está inhabilitado en el día  $j$  y bloque  $i$  para impartir clases, definido como:

$$F_{ij} = \begin{cases} 0 & \text{si el bloque está habilitado} \\ 1 & \text{si no} \end{cases}$$

### 3.1.2 VARIABLES

Se define la siguiente variable de decisión para representar la asignación de ayudantías:

- $X_{ijk} \in \{0, 1\}$ : Estado de la ayudantía en el día  $j$ , bloque  $i$ , impartida por el ayudante  $k$ .

$$X_{ijk} = \begin{cases} 1 & \text{si la ayudantía se asigna} \\ 0 & \text{en caso contrario} \end{cases}$$

Esta variable constituye la solución del problema y es utilizada tanto en la formulación de la función objetivo como en las restricciones del modelo.

### 3.1.3 DOMINIO

Los dominios de los índices utilizados en el modelo se definen a continuación:

- Días: Corresponden a los días en que se pueden realizar las clases; por simplicidad se utilizarán los días hábiles lunes, martes, miércoles, jueves y viernes. Para la implementación del código se utilizará  $j \in \{0, 1, 2, 3, 4\}$ .
- Bloques: Representa los bloques de horario que se encuentran disponibles. Ya que los utilizados por la UTFSM son 20 bloques, pero cada horario de ayudantía utiliza dos bloques seguidos; por lo tanto, se considerarán 10 bloques y se representarán como  $i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .
- Individuos: Corresponden a los participantes del sistema, tanto estudiantes como ayudantes. Para la matriz de estudiantes  $E$ , el índice se define como  $k \in \{0, \dots, n_E - 1\}$  mientras que para las matrices asociadas a ayudantes y a la variable de decisión  $X$ , se define como  $k \in \{0, \dots, n_A - 1\}$ . Donde  $n_E$  representa el número total de estudiantes y  $n_A$  el número total de ayudantes.

A partir de las variables y dominios definidos, a continuación se presentan las restricciones duras y blandas que rigen el problema de asignación de ayudantías.

### 3.1.4 RESTRICCIONES

Las restricciones se encargan de limitar las soluciones factibles y estas se dividen en restricciones duras, que se deben cumplir para que la solución sea válida, y restricciones blandas, las cuales no son necesarias de cumplir, pero al no hacerlo se obtiene una penalización.

#### RESTRICCIONES DURAS

- Un ayudante solo puede ser asignado a un bloque horario en el que se encuentre disponible. Para ello, se impone que no exista alguna superposición entre una asignación y un bloque en el que el ayudante tenga otra actividad.

$$X_{ijk} + A_{ijk} \leq 1, \forall i, j, k$$

- En cada bloque horario  $(i, j)$  puede asignarse a lo más un ayudante, evitando la superposición de ayudantías en un mismo horario.

$$\sum_k X_{ijk} \leq 1, \forall i, j$$

- Existen bloques horarios en los cuales la universidad no permite la realización de ayudantías. Esta restricción garantiza que ninguna ayudantía se asignará en un bloque prohibido.

$$X_{ijk} + F_{ij} \leq 1, \forall i, j, k$$

### RESTRICCIONES BLANDAS

Las restricciones blandas corresponden a las preferencias del estudiante al momento de elegir una ayudantía. Estas tendrán un costo menor a las restricciones duras, pero igualmente tendrán un impacto en la función objetivo, disminuyendo el *fitness*<sup>9</sup> de la solución.

- Penalización por ventanas generadas: Esta restricción busca reducir los tiempos muertos o ventanas generadas en el horario del estudiante por asignar la ayudantía.
  - El peso de penalización se calcula como:

$$W_{ventana}(i, j) = \frac{Ventana}{6}$$

- La cantidad de Ventanas corresponde a la suma de bloques libres que se generan antes y después de la ayudantía asignada en el bloque  $i$  del día  $j$ . Estos se calculan a partir de la última clase antes de la ayudantía (bloque  $l$ ) y la primera clase después de la ayudantía (bloque  $m$ ):

$$Ventana = Ventana_l + Ventana_m$$

- La ventana generada antes de la ayudantía ( $Ventana_l$ ) se define como:

$$Ventana_l = \begin{cases} 0 & \text{si } l = -1 \\ i - l - 1 & \text{si } l > -1 \end{cases}$$

- La ventana generada después de la ayudantía ( $Ventana_m$ ) se define como:

$$Ventana_m = \begin{cases} 0 & \text{si } m = -1 \\ m - i - 1 & \text{si } m > i \end{cases}$$

---

<sup>9</sup> Corresponde al valor obtenido al evaluar la solución en la función objetivo. Para este caso particular, a un mayor *fitness*, la solución es de mejor calidad.

- Penalización por asistir un día libre: Esta restricción busca evitar asignar la ayudantía un día en el que el estudiante no tenga otras actividades académicas, manteniendo así su día libre completo.
  - El peso de la penalización por día libre se asigna cuando  $l$  y  $m$  valen  $-1$ , ya que esto implica que no existe ninguna clase antes ni después de la ayudantía. Esto se puede resumir en:

$$W_{libre} = \begin{cases} 0.5 & \text{si } l = -1 \text{ y } m = -1 \\ 0 & \text{si no} \end{cases}$$

- Penalización por bloques nocturnos: Esta restricción penaliza la asignación de horarios en bloques considerados poco deseables por los estudiantes, los cuales son muy tarde. Los bloques a evitar corresponden a los horarios 17-18, 18-20.

$$W_{nocturno} = \begin{cases} 0.4 & \text{si } i \in \{8, 9\} \\ 0 & \text{si no} \end{cases}$$

- Penalización por bloques borde: Esta restricción es similar a la anterior, pero con un peso de penalización menor, ya que los bloques 1-2 y 15-16 son menos críticos que los nocturnos.

$$W_{borde} = \begin{cases} 0.3 & \text{si } i \in \{0, 7\} \\ 0 & \text{si no} \end{cases}$$

- Penalización por adyacencia a una clase relacionada: Esta restricción evita que la ayudantía se realice inmediatamente antes o después de una clase obligatoria relacionada a la asignatura. Para esto se utiliza el estado  $E_{ijk} = 2$ , el cual indica que el estudiante tiene una actividad relacionada a la asignatura, si la distancia entre el bloque con valor 2 y la ayudantía asignada es 0, entonces se realiza la penalización.

$$W_2 = \begin{cases} 0.5 & \text{si } distancia = 0 \\ 0 & \text{si no} \end{cases}$$

### 3.1.5 FUNCIÓN OBJETIVO

La función objetivo maximiza la participación de los estudiantes ( $P_a$ ) y resta todas las penalizaciones blandas.

$$\max Z = P_a - \sum_k (\sum_i \sum_j \text{Penalización}) X_{ijk}$$

Donde la Penalización es la suma de los pesos de las restricciones blandas violadas por la asignación de la matriz  $X_{ijk}$ , la cual se expresa de la siguiente forma:

$$\text{Penalización} = W_{ventana} + W_{libre} + W_{nocturno} + W_{borde} + W_2$$

## 3.2 REPRESENTACIÓN

### 3.2.1 ESTRUCTURA DE SOLUCIÓN (MATRIZ $X_{ijk}$ )

La solución del problema se representa mediante el conjunto de variables de decisión  $X_{i,j,k}$ , el cual puede interpretarse como un tensor binario de orden tres, con dimensiones  $[i][j][k]$ , correspondientes al bloque horario, día y ayudante respectivamente. Esta estructura modela la asignación de horarios de ayudantía, indicando si un ayudante específico es asignado a un determinado bloque en un día dado.

Para facilitar su comprensión, en la Figura 5 se presenta un ejemplo ilustrativo de esta representación. En dicho ejemplo, la solución asigna al estudiante  $k = 0$  el bloque  $i = 2$  del día  $j = 1$  (bloque 5-6, día martes), lo que se representa como  $X_{2,1,0} = 1$ . De esta manera análoga, el ayudante  $k = 1$  es asignado al bloque  $i = 4$  del día  $j = 3$  (bloque 9-10, día jueves), resultando en  $X_{4,3,1} = 1$ .

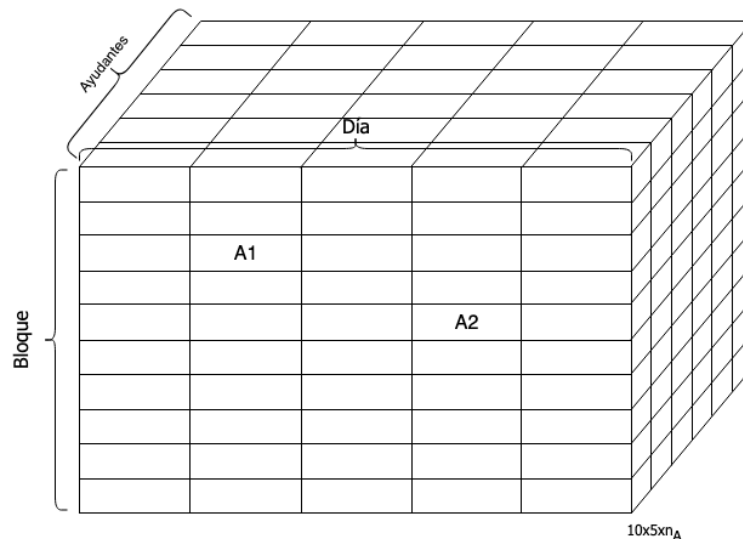


Figura 5: Representación de la solución.

Fuente: Elaboración propia.

Con el fin de simplificar la visualización, la Figura 5 muestra una proyección bidimensional de la solución, donde cada capa corresponde a un ayudante fijo. En este esquema, una capa contiene valores nulos en todas sus posiciones, excepto en aquella que representa el bloque y día asignados al ayudante correspondiente. Cabe destacar que esta figura constituye a una representación gráfica auxiliar, mientras que la solución completa corresponde al tensor  $X_{i,j,k}$  definido previamente.

## 3.3 SIMULATED ANNEALING

*Simulated Annealing* es una metaheurística de búsqueda local estocástica inspirada en el proceso físico de enfriamiento de metales. A diferencia de su predecesor, *Hill Climbing*, SA

incorpora un mecanismo probabilístico que permite aceptar soluciones de peor calidad en etapas tempranas del ciclo, tal como se describe en el trabajo seminal de Rutenbar [15]. Este comportamiento controlado le permite escapar de óptimos locales y explorar regiones más amplias del espacio de búsqueda, lo cual resulta especialmente relevante en problemas combinatorios de alta complejidad como el Educational Timetabling Problem.

### 3.3.1 SOLUCIÓN INICIAL (ALGORITMO *GREEDY*)

En el contexto de este trabajo, SA se utiliza para mejorar la solución inicial generada por un algoritmo *greedy*, el cual irá asignando los bloques de ayudantes en el primer bloque que tengan disponibles, cumpliendo así con las restricciones duras. Por ejemplo, como se puede apreciar en la Figura 6, el primer bloque disponible para el ayudante  $A_1$  es el bloque 2 del día lunes, el de  $A_2$  es el bloque 0 del día lunes, y el de  $A_3$  es el bloque 7 del día lunes.

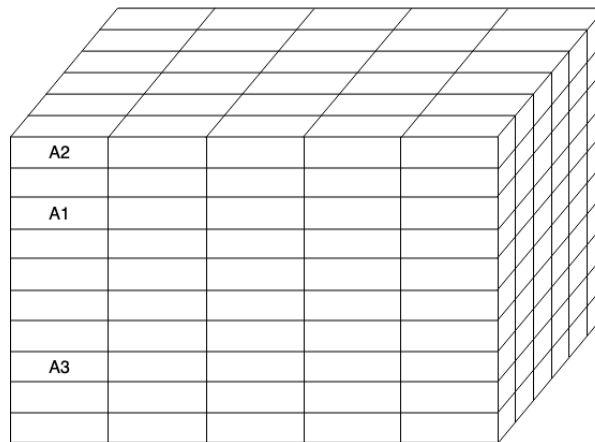


Figura 6: Solución inicial.  
Fuente: Elaboración propia.

### 3.3.2 OPERADORES DE VECINDARIO

A partir de una solución inicial como la representada en la Figura 6, se definen tres movimientos primordiales para la exploración del espacio de soluciones. Van Bulck et al. demuestran que la combinación de múltiples vecindarios con diferentes movimientos mejora significativamente el rendimiento de *Simulated Annealing* en problemas de *timetabling*, lo que respalda la elección de los operadores propuestos a continuación como mecanismo de exploración del espacio de soluciones [16].

Por simplicidad, para la explicación de los movimientos se mostrará únicamente una capa de la matriz  $X_{i,j,k}$ .

#### TRASLADO POR DÍA

Este operador modifica el día asignado a un ayudante, manteniendo fijo el bloque horario. El bloque previamente asignado se libera y se asigna el mismo bloque en un día distinto. Por

ejemplo, al trasladar el bloque del ayudante  $k = 0$  desde el día  $j = 1$  al día  $j = 3$ , se realizan las siguientes operaciones:

$$X_{2,1,0} = 0$$

$$X_{2,3,0} = 1$$

La Figura 7 ilustra este movimiento.

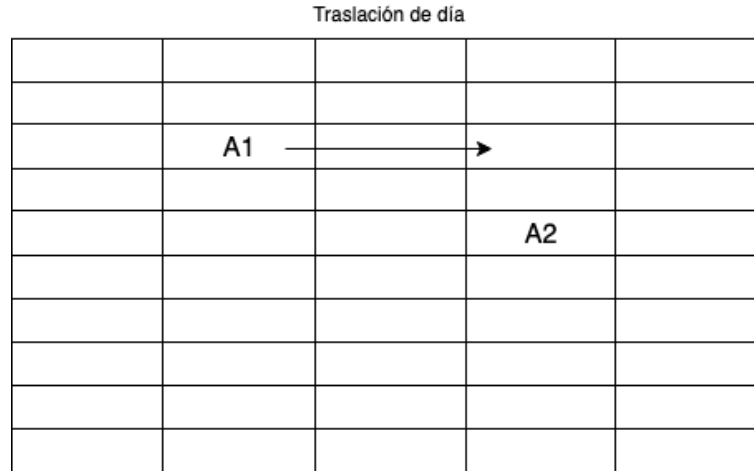


Figura 7: Movimiento por día.  
Fuente: Elaboración propia.

### TRASLADO POR BLOQUE

En este caso, el ayudante mantiene el día asignado, pero se modifica el bloque horario. El bloque original se libera y se asigna un nuevo bloque dentro del mismo día. Por ejemplo, al trasladar la asignación del ayudante  $k = 1$  al bloque  $i = 0$  en el día  $j = 3$ , se tiene:

$$X_{4,3,1} = 0$$

$$X_{0,3,1} = 1$$

Este movimiento se representa en la Figura 8.

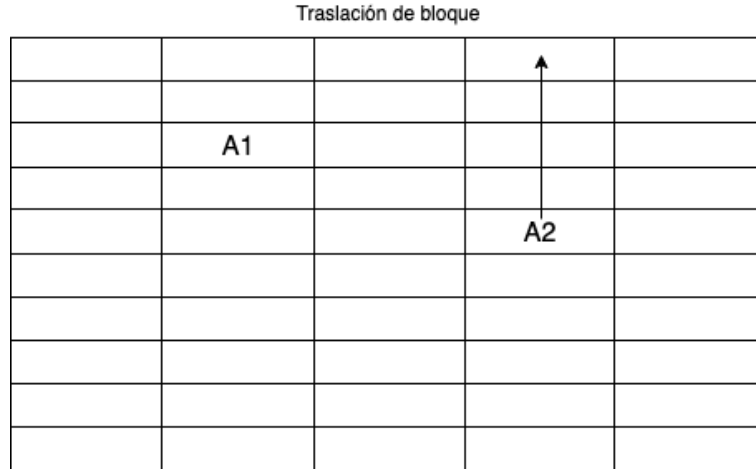


Figura 8: Movimiento por bloque.  
Fuente: Elaboración propia.

**INTERCAMBIO ENTRE AYUDANTES (SWAP)**

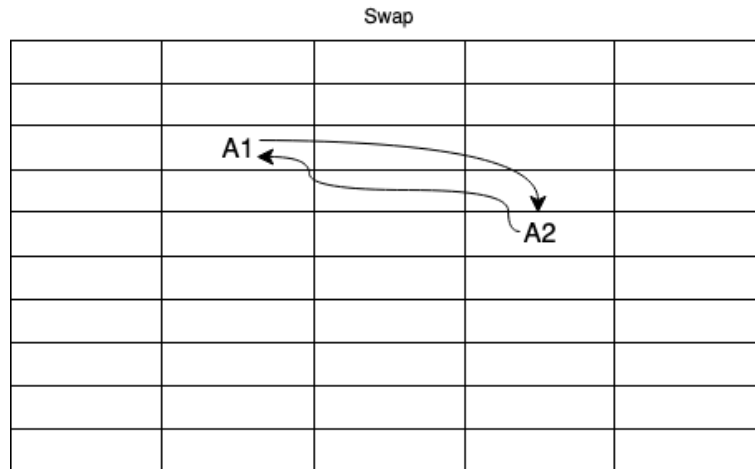
Este operador intercambia las asignaciones horarias completas entre dos ayudantes, por lo tanto, este movimiento solo se realiza si la cantidad de ayudantes es superior a dos. Para ello, se consideran las posiciones iniciales  $(i_1, j_1)$  e  $(i_2, j_2)$  de los ayudantes  $k_1$  y  $k_2$ , respectivamente. Primero se liberan ambas asignaciones:

$$X_{i_1 j_1 k_1} = 0, \quad X_{i_2 j_2 k_2} = 0$$

Luego, se asigna a cada ayudante el horario originalmente asignado al otro:

$$X_{i_1 j_1 k_2} = 1, \quad X_{i_2 j_2 k_1} = 1$$

La Figura 9 ilustra este movimiento. La factibilidad de la nueva asignación se evalúa posteriormente mediante las restricciones del modelo.

Figura 9: Movimiento *swap*.

Fuente: Elaboración propia.

### 3.3.3 PROCESO DE EVALUACIÓN Y ACEPTACIÓN

#### EVALUACIÓN DEL *FITNESS*

Durante la ejecución del algoritmo *Simulated Annealing*, en cada iteración se genera una solución vecina a partir de la solución actual mediante la aplicación de uno de los operadores de vecindario definidos previamente. Dicha solución candidata es sometida a un proceso de evaluación con el fin de determinar su calidad y decidir su posible aceptación.

En primer lugar, se verifica el cumplimiento de las restricciones duras definidas en la Sección 3.1.4. En caso de que la solución viole alguna de estas restricciones, se considera no factible y es descartada del proceso de búsqueda, sin ser evaluada por la función objetivo.

Para las soluciones factibles, el *fitness* se calcula utilizando la función objetivo definida en la Sección 3.1.5. Esta función busca maximizar la participación de los estudiantes en las ayudantías, representada por  $P_a$ , y simultáneamente minimizar las penalizaciones asociadas al incumplimiento de las restricciones blandas. Dichas penalizaciones corresponden a la suma ponderada de los distintos criterios de calidad horaria evaluados sobre las asignaciones activas de la solución  $X_{i,j,k}$ .

El valor resultante del *fitness* corresponde a una medida escalar que permite comparar las soluciones alternativas, donde valores mayores indican una mejor calidad global de la asignación. Este valor es utilizado directamente por el algoritmo para guiar el proceso de búsqueda y para determinar la aceptación de soluciones vecinas según la regla probabilística de SA, la cual se describe en la siguiente subsección. Cabe destacar que esta formulación permite balancear explícitamente el beneficio global de la asignación con la calidad individual de los horarios, evitando soluciones que maximicen la participación a costa de una alta penalización para los estudiantes.

### REGLA DE ACEPTACIÓN

El algoritmo *Simulated Annealing* emplea una regla de aceptación basada en la comparación entre la solución actual y una solución candidata generada mediante un operador vecindario. En primer lugar, se calcula la variación del valor de la función objetivo como:

$$\Delta f = f_{nuevo} - f_{actual}$$

Dado que el objetivo del problema es maximizar la función de evaluación, toda solución candidata que produzca una mejora ( $\Delta f > 0$ ) es aceptada de manera inmediata, pasando a convertirse en la nueva solución actual.

En caso de que la solución candidata sea peor que la actual ( $\Delta f \leq 0$ ), esta aún puede ser aceptada con una probabilidad  $P$ , definida como:

$$P = e^{\frac{\Delta f}{T}},$$

donde  $T$  corresponde a la temperatura actual del sistema. La solución es aceptada si esta probabilidad es mayor que un valor aleatorio generado a partir de una distribución uniforme en el intervalo  $[0, 1]$ .

Este mecanismo permite aceptar, de forma controlada, soluciones de menor calidad durante las primeras etapas del algoritmo, favoreciendo la exploración del espacio de búsqueda y reduciendo el riesgo de convergencia prematura hacia óptimos locales. A medida que la temperatura disminuye, la probabilidad de aceptar soluciones peores se reduce progresivamente, promoviendo un comportamiento más explotativo en las etapas finales de búsqueda.

#### 3.3.4 CRITERIOS DE PARADA

El proceso de búsqueda del algoritmo *Simulated Annealing* finaliza cuando se cumple al menos una de las siguientes condiciones de término.

- La temperatura actual  $temp_{curr}$  desciende por debajo de la temperatura final  $temp_{final}$  definida previamente como parámetro del algoritmo
- Se alcanza el número máximo de iteraciones permitidas,  $iteration_{max}$ , establecido al inicio de la ejecución

Una vez satisfecha cualquiera de estas condiciones, el algoritmo detiene su ejecución y retorna la mejor solución factible encontrada a lo largo de todo el proceso de enfriamiento, independientemente de la solución en la que se encuentre la última iteración. Estas condiciones permiten balancear adecuadamente la exploración del espacio de búsqueda y el costo computacional del algoritmo.

### 3.3.5 PSEUDOCÓDIGO

El flujo de trabajo del algoritmo de *Simulated Annealing* utilizado para resolver el problema de asignación de ayudantías, se resume a continuación en el pseudocódigo presentado en la Figura 10.

---

**Algorithm 1** Simulated Annealing

---

```

1: while  $temp_{curr} > temp_{final}$  and  $iteration < iteration_{max}$  do
2:    $solution_{new} = random\_move(solution_{curr})$ 
3:   if not  $validate\_solution(solution_{new})$  then
4:      $fitness_{new} = -inf$ 
5:   else
6:      $fitness_{new} = fitness(solution_{new})$ 
7:   end if
8:    $fitness_{delta} = fitness_{new} - fitness_{curr}$ 
9:   if  $fitness_{delta} > 0$  or  $e^{\frac{fitness_{delta}}{temp_{curr}}} > random(0, 1)$  then
10:     $solution_{curr} = solution_{new}$ 
11:     $fitness_{curr} = fitness_{new}$ 
12:    if  $fitness_{curr} > fitness_{best}$  then
13:       $solution_{best} = solution_{curr}$ 
14:       $fitness_{best} = fitness_{curr}$ 
15:    end if
16:  end if
17:   $temp_{curr} = temp_{curr} * alpha$ 
18:   $iteration = iteration + 1$ 
19: end while

return  $solution_{best}$ 

```

---

Figura 10: Pseudocódigo *Simulated Annealing*.  
Fuente: Elaboración propia.

En la figura se integran las distintas etapas descritas previamente:

- Generación de soluciones vecinas mediante operadores de vecindario.
- Validación de restricciones duras.
- Evaluación del *fitness*.
- Regla de aceptación probabilística.
- Esquema de enfriamiento.

El algoritmo mantiene y actualiza la mejor solución encontrada a lo largo del proceso, la cual es retornada una vez que se cumple alguno de los criterios de parada definidos.

### 3.3.6 CALIBRACIÓN DE PARÁMETROS

El rendimiento de *Simulated Annealing* depende fuertemente de la calibración de sus parámetros principales, especialmente de la temperatura inicial ( $T_0$ ), tasa de enfriamiento ( $\alpha$ ) y el número de iteraciones ( $Iteration_{max}$ ). Estudios recientes, como Bellio et al., respaldan esta afirmación, presentando un esquema de calibración basado en características del problema, obteniendo mejoras significativas en instancias del *Curriculum-based Course Timetabling* (CB-CTT) [17]. Este resultado refuerza la necesidad de ajustar experimentalmente los parámetros en lugar de utilizar valores fijos o heurísticos generales.

Bajo esta motivación, se definieron tres conjuntos experimentales diseñados para explorar de manera sistemática distintas dinámicas de enfriamiento y niveles de exploración del espacio de búsqueda. Para cada experimento se seleccionaron rangos específicos de los principales parámetros de SA y posteriormente se generaron todas las combinaciones posibles mediante un producto cartesiano. El conjunto completo de configuraciones fue almacenado en archivos CSV disponibles en la carpeta *results/configurations* del repositorio [18], permitiendo garantizar reproducibilidad, trazabilidad y la ejecución automatizada de los experimentos.

En total se realizaron tres experimentos independientes, cada uno con un número distinto de combinaciones y considerando la mejor configuración aquella con un mayor valor de *fitness* y, en caso que más de una posea el mayor *fitness*, un menor tiempo de ejecución.

#### PRIMER EXPERIMENTO

Este primer conjunto experimental se diseñó para establecer un rango de exploración inicial con un enfriamiento conservador, utilizando valores de temperatura inicial ( $T_0$ ) moderados y una tasa de enfriamiento ( $\alpha$ ) estándar. Se generaron 2000 configuraciones mediante el producto cartesiano de los parámetros que se pueden ver listados en la Tabla 3.

Tabla 3: Configuraciones iniciales del primer experimento.  
Fuente: Elaboración propia.

Parámetro	Símbolo	Valores
Temperatura inicial	$T_0$	$[10^2, 10^3, 10^4, 10^5, 10^6]$
Temperatura final	$T_f$	$[10^{-3}, 10^{-2}, 10^{-1}, 1, 10]$
Factor de enfriamiento	$\alpha$	$[0.90, 0.93, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999]$
Iteraciones máximas	$Iteration_{max}$	$[5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 2.5 \times 10^5]$

El algoritmo se ejecutó utilizando dos instancias de prueba:

- Test A: INF285 Computación Científica, segundo semestre del año 2025.
- Test B: INF295 Inteligencia Artificial, primer semestre del año 2025.

Los resultados de la mejor configuración encontrada para cada instancia se resumen en la Tabla 4, en la cual se agregó el valor de *fitness* porcentual para tener una mejor comparación entre las diferentes asignaturas. Tras el procesamiento de las 2000 configuraciones, la solución óptima obtenida por el algoritmo difirió significativamente entre las dos instancias.

Tabla 4: Comparación de resultados, experimento 1.  
Fuente: Elaboración propia.

Variable	Test A	Test B
config_id	cfg_122	cfg_161
initial_temp	100.0	100.0
final_temp	1.0	10.0
alpha	0.9	0.9
max_iter	10000	5000
time	0.07	0.05
final_fitness	26.34	17.34
Fitness porcentual	84.95	66.67

- Test A (INF285): Se identificó la configuración *cfg\_122* como la más eficiente. Esta configuración alcanzó un valor de *fitness* de 26.34, lo que equivale a un 84.95% del *fitness* porcentual. La solución fue obtenida con un rango de enfriamiento amplio ( $T_0 = 100$  a  $T_f = 1$ ) y 10000 iteraciones máximas, en un tiempo de ejecución de 0.07 segundos.
- Test B (INF295): La mejor configuración corresponde a *cfg\_161*, el cual posee un *fitness* porcentual de 66.67%. Aunque es un resultado competitivo, es trivial en comparación al obtenido en el Test A, lo que indica que la estructura de restricciones de la asignatura presentó un mayor desafío para el algoritmo bajo esta gama de parámetros. Esta solución fue obtenida con un máximo de 5000 iteraciones en 0.05 segundos, destacando la eficiencia computacional del algoritmo en esta instancia.

El análisis de este primer experimento sugiere que una exploración más profunda, es decir, con una mayor cantidad de iteraciones, y un enfriamiento más lento (menor  $T_f$ ) podrían ser necesarios para alcanzar óptimos superiores.

## SEGUNDO EXPERIMENTO

Este segundo conjunto experimental se diseñó para profundizar la exploración en la dinámica de enfriamiento, utilizando un factor de enfriamiento ( $\alpha$ ) más bajo y un rango más amplio de temperaturas iniciales ( $T_0$ ) para evaluar si un proceso más metódico puede evitar los óptimos locales del primer experimento.

Se generaron 3024 configuraciones. El diseño de los parámetros iniciales se detalla en la Tabla 5, destacando un rango amplio de  $T_0$  (desde 0.1 hasta 500) y el uso de un factor de enfriamiento  $\alpha$  que incluye un valor más bajo. Los *tests* fueron ejecutados en las mismas instancias: Test A (INF285) y Test B (INF295).

Tabla 5: Configuraciones iniciales del segundo experimento.  
Fuente: Elaboración propia.

Parámetro	Símbolo	Valores
Temperatura inicial	$T_0$	$[10^{-1}, 1, 10, 50, 100, 500]$
Temperatura final	$T_f$	$[10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1]$
Factor de enfriamiento	$\alpha$	$[0.85, 0.90, 0.93, 0.95, 0.97, 0.99, 0.995]$
Iteraciones máximas	$Iteration_{max}$	$[10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5]$

Los resultados de la mejor configuración encontrada para cada instancia se resumen en la Tabla 6 a continuación:

Tabla 6: Comparación de resultados, experimento 2.  
Fuente: Elaboración propia.

Variable	Test A	Test B
config_id	cfg_970	cfg_717
initial_temp	50.0	10.0
final_temp	1.0	1.0
alpha	0.85	0.85
max_iter	50000.0	10000.0
time	0.05	0.03
final_fitness	26.34	17.34
Fitness porcentual	84.95	66.67

- Test A: La configuración más óptima (*cfg\_970*) mantiene el mismo *fitness* porcentual máximo del primer experimento (84.95%). Sin embargo, en el test actual, se requirió de una exploración significativamente más extensa (50000 iteraciones en 0.05 segundos) en comparación con el experimento 1 (10000 iteraciones en 0.07 segundos). El tiempo de ejecución más bajo sugiere una mayor eficiencia del código o una convergencia más rápida hacia el óptimo.
- Test B: Nuevamente, se obtuvo el mismo *fitness* porcentual máximo (66.67%) con la configuración *cfg\_717*. Lo más notable es la extrema eficiencia de esta solución, la cual se logró en tan solo 0.03 segundos y 10000 iteraciones, comenzando de una temperatura inicial muy baja ( $T_0 = 10$ ).

Los resultados del segundo experimento refuerzan la conclusión de que los valores de *fitness* porcentual 84.95% y 66.67% son óptimos locales recurrentes para las instancias INF285 e INF295, respectivamente. Esto sugiere que el mayor desafío para el algoritmo no es el factor de enfriamiento o la temperatura inicial, sino la estructura inherente de las restricciones duras y blandas en la función objetivo. Para la selección final, se priorizan las configuraciones que logran estos *fitness* en el menor tiempo posible.

### TERCER EXPERIMENTO

El tercer conjunto experimental se enfocó en un enfriamiento lento mediante el uso de factores  $\alpha$  cercanos a 1, combinados con una alta cantidad de iteraciones máximas. El objetivo fue maximizar la exploración del espacio de búsqueda, evaluando si el tiempo adicional de búsqueda permitía superar los óptimos recurrentes de 84.95% y 66.67%.

Se generaron un total de 2400 configuraciones mediante el producto cartesiano de los parámetros listados en la Tabla 7. Para las temperaturas, se empleó un espaciado logarítmico, lo que permitió explorar diversos órdenes de magnitud y cubrir un espectro amplio de condiciones iniciales. Esta configuración inicial privilegió factores de enfriamiento altos para favorecer una búsqueda prolongada y profunda. Los *tests* fueron ejecutados en las mismas instancias que los experimentos anteriores.

Tabla 7: Configuraciones iniciales del tercer experimento.  
Fuente: Elaboración propia.

Parámetro	Símbolo	Valores
Temperatura inicial	$T_0$	$[10^2, 6.3^2, 3.9^3, 2.5^4, 1.5^5 10^6]$
Temperatura final	$T_f$	$[10^{-3}, 10^{-2}, 10^{-1}, 1, 10]$
Factor de enfriamiento	$\alpha$	$[0.90, 0.93, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999]$
Iteraciones máximas	$Iteration_{max}$	$[5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 2.5 \times 10^5]$

Los resultados de la mejor configuración encontrada para cada instancia se resumen en la Tabla 8:

Tabla 8: Comparación de resultados, experimento 3.  
Fuente: Elaboración propia.

Variable	Test A	Test B
config_id	cfg_133	cfg_166
initial_temp	100.0	100.0
final_temp	1.0	10.0
alpha	0.95	0.93
max_iter	50000	5000
time	0.14	0.06
final_fitness	26.34	17.34
Fitness porcentual	84.95	66.67

Este experimento confirma que la estrategia de enfriamiento lento y de mayor exploración no logró superar los límites del *fitness* porcentual establecido en los experimentos anteriores. Esto sugiere que los valores 84.95% y 66.67% representan el mejor óptimo local recurrente dentro del espacio de búsqueda del modelo.

- Test A: La configuración óptima *cfg\_133* alcanzó el óptimo en 0.14 segundos y con una cantidad máxima de 50000 iteraciones, lo que la convierte en la opción menos eficiente hasta ahora en comparación con los otros experimentos para el mismo *fitness*.

- Test B: La configuración óptima *cfg\_166* fue más lenta que las obtenidas en los anteriores experimentos, confirmando la superioridad de la eficiencia de la configuración *cfg\_717* del segundo experimento.

### 3.3.7 CALIBRACIÓN DE LA FUNCIÓN *FITNESS*

Si bien la calibración de los parámetros del algoritmo Simulated Annealing permite controlar el proceso de exploración del espacio de búsqueda, la calidad de las soluciones depende directamente de la formulación de la función objetivo. En particular, el uso de restricciones blandas ponderadas implica un compromiso entre distintos criterios, tales como la comodidad del estudiantado y la maximización de la participación.

Para analizar este equilibrio, se realizaron experimentos utilizando la asignatura INF276. Esta elección se fundamenta en observaciones preliminares en las que, bajo una configuración sin ponderación de pesos, el algoritmo presentaba una degradación significativa del rendimiento. Se ejecutaron pruebas con las mejores configuraciones obtenidas en la calibración de los parámetros, es decir, con las configuraciones *cfg\_122* del Test A y *cfg\_717* del Test B. Se variaron los pesos asociados a las restricciones blandas mediante un producto cartesiano de los valores detallados en la Tabla 9, generando un total de 16807 configuraciones cuyos resultados se pueden encontrar en la carpeta *results/configurations/weight* del repositorio [18]. Estos valores fueron seleccionados para garantizar una representatividad diversa en el rango  $[0.1, 1.0]$ . Si bien los cinco parámetros cuentan con siete valores posibles, variables como  $W_{nocturno}$  inician su ponderación en 0.5, a diferencia del 0.1 aplicado a otros factores, debido a su mayor relevancia para el estudiantado.

El objetivo fue evaluar si un incremento en la penalización de ciertos conflictos permite obtener soluciones de alta calidad percibida sin degradar significativamente la capacidad de asistencia.

Tabla 9: Ponderación de restricciones blandas.

Fuente: Elaboración propia.

Parámetro	Símbolo	Valores
W_FREE_DAY	$W_{libre}$	[0.1, 0.25, 0.4, 0.55, 0.7, 0.85, 1.0]
W_SLOT_EVE	$W_{nocturno}$	[0.5, 0.583, 0.6, 0.75, 0.83, 0.916, 1.0]
W_SLOT_DAY	$W_{borde}$	[0.1, 0.25, 0.4, 0.55, 0.7, 0.85, 1.0]
W_WINDOWS	$W_{ventana}$	[0.5, 0.583, 0.6, 0.75, 0.83, 0.916, 1.0]
W_SLOT_2	$W_{libre}$	[0.1, 0.25, 0.4, 0.55, 0.7, 0.85, 1.0]

Tras la experimentación, se identificó que la configuración que maximiza el *fitness* con la menor cantidad de conflictos y tiempo de ejecución corresponde a los pesos  $W_{libre} = 0.1$ ,  $W_{nocturno} = 1.0$ ,  $W_{borde} = 0.1$ ,  $W_{ventana} = 0.5$  y  $W_2 = 0.7$ . Esta combinación resultó en un *fitness* total de 47.01, lo que representa un valor porcentual del 88.7%.

Tabla 10: Mejor configuración encontrada.  
Fuente: Elaboración propia.

Parámetro	Valores
config_id	cfg_2063
W_FREE_DAY	0.1
W_SLOT_EVE	1.0
W_SLOT_DAY	0.1
W_WINDOWS	0.5
W_SLOT_2	0.7
time	0.13
final_fitness	47.01
Fitness porcentual	88.7

### 3.3.8 CONFIGURACIÓN FINAL SELECCIONADA

Para la selección de la configuración final del algoritmo SA, se prioriza la eficiencia computacional, es decir, que tenga un menor tiempo de ejecución. Dado que los experimentos de calibración demostraron que el valor del *fitness* máximo se mantuvo constante entre las diferentes configuraciones, la priorización de la eficiencia no sacrifica la calidad de la solución. Por lo tanto, se seleccionaron las dos configuraciones que lograron el mejor *fitness* porcentual para cada instancia con el menor tiempo registrado. Los *sets* de parámetros seleccionados se resumen en la Tabla 11.

Tabla 11: Mejores resultados por test.  
Fuente: Elaboración propia

Variable	Test A	Test B
config_id	cfg_122	cfg_717
initial_temp	100.0	10.0
final_temp	1.0	1.0
alpha	0.9	0.85
max_iter	10000	10000
time	0.07	0.03
final_fitness	26.34	17.34
Fitness normalizado	84.95	66.67

Adicionalmente, todas las ejecuciones integran la ponderación de la función *fitness* optimizada en la Sección 3.3.6, utilizando los pesos:

- $W_{libre} = 0.1,$
- $W_{nocturno} = 1.0,$
- $W_{borde} = 0.1,$

- $W_{ventana} = 0.5$ ,
- $W_2 = 0.7$ .

Para maximizar la probabilidad de aproximarse al óptimo global, se implementará una estrategia de ejecución dual por cada asignatura:

- Primera ejecución: Utiliza la configuración *cfg\_122*, la más eficiente para Test A.
- Segunda ejecución: Utiliza la configuración *cfg\_717*, la más eficiente para Test B.

Finalmente, el sistema seleccionará la mejor solución obtenida entre ambas ejecuciones basándose en los siguientes criterios jerárquicos:

- Criterio principal: Se elige la solución que posea un *fitness* más alto.
- Criterio desempate: En caso de que exista más de una solución con un mismo *fitness*, se elige la opción que tenga una mayor cantidad de participantes.

### 3.4 SOLVER

Con el objetivo de contar con un punto de referencia sólido para la evaluación del algoritmo propuesto, se implementó un modelo de Programación Lineal Entera que permite abordar el problema de asignación de ayudantías desde un enfoque exacto. A diferencia de las metaheurísticas, este tipo de técnicas exploran el espacio de soluciones de manera sistemática, garantizando soluciones óptimas bajo la formulación matemática definida, siempre que los recursos computacionales lo permitan.

No obstante, el propósito del *solver* en este trabajo no es reemplazar al algoritmo *Simulated Annealing* ni proponerlo como una solución final al problema, sino utilizarlo como una herramienta de comparación controlada frente al *Baseline*<sup>10</sup>. Para que dicha comparación sea significativa, fue necesario adaptar la formulación del modelo ILP de manera que refleje los mismos criterios de evaluación utilizados en el enfoque metaheurístico, en particular aquellos asociados a la maximización de la participación estudiantil y a la calidad horaria de las asignaciones.

En este capítulo se describe la formulación del modelo ILP propuesto, las adaptaciones realizadas para incorporar restricciones blandas y penalizaciones de calidad, así como el proceso de resolución mediante un *solver* exacto. De esta forma, se establece un marco común de evaluación que permite contrastar de manera coherente los resultados obtenidos por el *Baseline*, el *solver* y el algoritmo metaheurístico propuesto.

A diferencia del algoritmo SA, cuyo carácter estocástico permite explorar soluciones de alta calidad en espacios de búsqueda complejos y de gran dimensión, el enfoque de ILP presenta un comportamiento determinista, orientado a la obtención de soluciones óptimas respecto de la

---

<sup>10</sup> Solución obtenida a través de los datos oficiales; corresponde a las ayudantías vigentes al momento de la recolección de datos.

formulación matemática definida. Sin embargo, este tipo de modelos exactos suele enfrentar limitaciones prácticas asociadas al crecimiento del número de variables y restricciones, lo que impacta directamente en los tiempos de resolución y en su escalabilidad.

En este contexto, el uso del solver ILP es proporcionar un punto de referencia controlado que permita evaluar la calidad de las soluciones obtenidas por el algoritmo SA bajo un mismo marco de criterios. De esta manera, la comparación entre ambos enfoques permite analizar el compromiso entre optimalidad teórica y eficiencia computacional, reforzando la validez del enfoque propuesto para escenarios de mayor complejidad.

Para la resolución del modelo ILP, se utilizó el solver *COIN-OR CBC (Coin Branch and Cut)*, a través de la librería de Python PuLP. Este tipo de enfoque ha sido aplicado con éxito en problemas reales de *timetabling* mediante herramientas basadas en ILP y CBC [19]. CBC es un *solver* de código abierto basado en técnicas de *branch and bound* y *cutting planes*, diseñado para resolver problemas de programación lineal entera de manera eficiente. Su integración con PuLP permite una formulación clara y reproducible del modelo, lo que facilita la experimentación y el análisis de resultados sin la necesidad de depender de *software* propietario.

La elección del PuLP y CBC se fundamenta en su uso de investigación académica, su accesibilidad y su capacidad para obtener soluciones óptimas en tiempos razonables para instancias de tamaño moderado. Estas características lo convierten en una herramienta adecuada para estudiar el desempeño de enfoques exactos en el *Educational Timetabling Problem* y compararlos con métodos aproximados basados en metaheurísticas.

### 3.4.1 VARIABLES

El problema de asignación de ayudantías se formuló como un modelo de ILP cuyo objetivo es determinar los bloques horarios en los que se realizan ayudantías, de modo que se maximice la participación estudiantil y se minimicen las penalizaciones asociadas a criterios de calidad horaria. La formulación del modelo se diseñó de manera consistente con la representación y función de evaluación utilizadas en el algoritmo metaheurístico, permitiendo una comparación directa entre ambos enfoques.

El modelo considera explícitamente la disponibilidad de estudiantes y ayudantes, así como las restricciones institucionales relacionadas con bloques prohibidos. A diferencia de un modelo ILP tradicional enfocado únicamente en la factibilidad, esta formulación incorpora variables adicionales que permiten capturar la asistencia efectiva de los estudiantes y modelar las penalizaciones blandas de forma lineal.

En particular, se definen las siguientes variables:

- $X_{ijk} \in \{0, 1\}$ : El ayudante  $k$  imparte una ayudantía el día  $j$  y bloque  $i$

$$X_{ijk} = \begin{cases} 1 & \text{si imparte la ayudantía} \\ 0 & \text{en caso contrario} \end{cases}$$

- $Y_{kij} \in \{0, 1\}$ : El estudiante  $k$  puede asistir a una ayudantía asignada en el día  $j$ , bloque  $i$

$$Y_{kij} = \begin{cases} 1 & \text{si asiste} \\ 0 & \text{en caso contrario} \end{cases}$$

- $Z_k \in \{0, 1\}$ : El estudiante  $k$  puede asistir al menos a una ayudantía durante la semana

$$Z_k = \begin{cases} 1 & \text{si puede asistir} \\ 0 & \text{en caso contrario} \end{cases}$$

Estas variables permiten modelar explícitamente la cobertura estudiantil, alineando el objetivo del *solver* con el criterio de participación maximizada utilizado tanto en el *Baseline* como en el algoritmo. Además, se incorporan variables auxiliares para representar penalizaciones blandas que, en el enfoque metaheurístico, son evaluadas de manera secuencial. Bajo esta formulación, el modelo introduce variables binarias que permiten representar tanto la asignación de ayudantías como la asistencia efectiva de los estudiantes. En particular, la penalización asociada a la asistencia en días sin clases se modela mediante la variable:

$$D_{kj} \in \{0, 1\},$$

la cual indica si el estudiante  $k$  asiste a una ayudantía en el día  $j$  en el que no posee otras actividades académicas regulares. Esta variable permite internalizar la penalización por día libre dentro de la función objetivo de manera lineal.

Para penalizaciones de mayor complejidad estructural, como la generación de ventanas horarias o la asistencia en bloques extremos, se precomputó el costo asociado a cada combinación estudiante/día/bloque. Estos costos se incorporan como coeficientes constantes en la función objetivo, permitiendo aproximar el comportamiento del *fitness* definido previamente sin comprometer la linealidad del modelo.

La función objetivo del modelo ILP se define entonces como la maximización del número de estudiantes que pueden asistir al menos a una ayudantía durante la semana, representado por la suma de las variables  $Z_k$ , menos un conjunto de penalizaciones ponderadas asociadas a criterios de calidad horaria. Los pesos utilizados para cada penalización corresponden exactamente a los definidos en la función *fitness* del algoritmo metaheurístico, garantizando coherencia y comparabilidad entre ambos enfoques.

De esta forma, el modelo ILP no solo busca soluciones factibles, sino que también internaliza los mismos compromisos entre cobertura y calidad presentes en el algoritmo *Simulated Annealing*, permitiendo utilizar el *solver* como un punto de referencia consistente dentro del marco experimental del trabajo.

### 3.4.2 FUNCIÓN OBJETIVO

El objetivo del modelo es maximizar la participación estudiantil en las ayudantías, incorporando simultáneamente criterios de calidad horaria mediante penalizaciones blandas. Para ello, se define una función objetivo que prioriza la asignación de al menos una ayudantía semanal por estudiante, representada a través de las variables  $Z_k$  y penaliza aquellas configuraciones que deterioran la calidad de la solución desde el punto de vista académico y organizacional.

Formalmente, la función objetivo se construye como la maximización de la suma de las variables  $Z_k$ , que indican si el estudiante  $k$  puede asistir al menos a una ayudantía durante la semana, menos un conjunto de términos de penalización asociados a distintos criterios de calidad horaria. Estos criterios incluyen, entre otros, la asistencia en días sin otras actividades académicas regulares, la generación de ventanas horarias y la asignación de ayudantías en bloques extremos del día.

De manera formal, la función objetivo del modelo ILP puede expresarse como:

$$\max(\sum_k Z_k - \sum_k \sum_j \alpha D_{kj} - \sum_k \sum_i \sum_j \beta_{ij} Y_{kij})$$

- $\alpha$  representa el peso de penalización por día libre.
- $\beta_{ij}$  encapsula los costos precomputados.

Las penalizaciones se modelan de manera lineal mediante variables auxiliares y coeficientes precomputados, permitiendo aproximar el comportamiento de la función *fitness* utilizada en el algoritmo metaheurístico. Los pesos asociados a cada penalización corresponden exactamente a los definidos en dicho *fitness*, garantizando que el compromiso entre cobertura estudiantil y calidad horaria sea consistente en ambos enfoques.

De esta forma, la función objetivo del modelo ILP no solo busca soluciones factibles, sino que internaliza los mismos criterios de evaluación utilizados por el algoritmo *Simulated Annealing*, permitiendo utilizar el *solver* como un punto de referencia coherente dentro del marco experimental del trabajo.

### 3.4.3 RESTRICCIONES

El modelo de Programación Lineal Entera considera las mismas restricciones duras y blandas definidas previamente en la Sección 3.1.4, garantizando coherencia entre la formulación matemática del problema y los distintos enfoques de resolución utilizados en este trabajo.

Las restricciones duras aseguran la factibilidad de las asignaciones de ayudantías, incluyendo la superposición de bloques horarios, el respeto de la disponibilidad de estudiantes y ayudantes, y la exclusión de bloques prohibidos. Estas restricciones se modelan directamente mediante desigualdades lineales sobre las variables de decisión  $X_{ijk}$  y  $Y_{kij}$ .

Por su parte, las restricciones blandas no se imponen como condiciones estrictas, sino que se incorporan a la función objetivo a través de penalizaciones ponderadas. En el caso del *solver*, algunas de estas penalizaciones se representan mediante variables auxiliares, como  $D_{kj}$ , mientras que otras se integran mediante costos precomputados asociados a combinaciones específicas de estudiante, día y bloque horario.

De esta forma, el modelo ILP mantiene los mismos criterios de evaluación definidos en la formulación matemática, permitiendo una comparación directa y consistente con el *Baseline* y el algoritmo metaheurístico propuesto.

### 3.4.4 PROCESO DE RESOLUCIÓN

El modelo formulado fue resuelto utilizando un *solver* exacto, el cual explora de manera sistemática el espacio de soluciones definido por las variables, restricciones y función objetivo descritas previamente. Dado el carácter exacto de este enfoque, el *solver* garantiza la obtención de una solución óptima respecto a la formulación matemática planteada, siempre que los recursos computacionales disponibles lo permitan.

El proceso de resolución se ejecuta de manera independiente para cada instancia del problema, utilizando como entrada la información de disponibilidad de los estudiantes y ayudantes, así como las restricciones institucionales asociadas a los bloques prohibidos. Para cada caso, el *solver* genera una asignación de ayudantías que maximiza la cobertura estudiantil efectiva, penalizando simultáneamente aquellas configuraciones que presentan menor calidad horaria según los criterios definidos.

Una vez obtenida la solución, se extraen métricas de interés comparables con las utilizadas en el *Baseline* y en el algoritmo metaheurístico, tales como el número de estudiantes que pueden asistir a al menos una ayudantía y el valor total de la función objetivo. Adicionalmente, se registra el tiempo de ejecución del *solver*, permitiendo analizar el desempeño computacional del enfoque exacto frente al algoritmo *Simulated Annealing*.

De esta forma, el *solver* ILP se incorpora como un punto de referencia controlado dentro del marco experimental del trabajo, proporcionando una base sólida para la comparación de resultados y el análisis del compromiso entre optimalidad y costo computacional.

## CAPÍTULO 4: VALIDACIÓN DE SOLUCIÓN

Los experimentos se ejecutaron en un entorno controlado, utilizando un computador con procesador Apple M1, 8 GB de memoria RAM y sistema operativo macOS Sequoia 15.6.1. El desarrollo de la solución se realizó en el lenguaje de programación Python, seleccionado por su amplia disponibilidad de librerías orientadas a la modelación, optimización y análisis de datos. En particular, se emplearon herramientas como NumPy para la manipulación de estructuras de datos, la librería PuLP para la implementación del *solver* y Matplotlib para la generación de visualizaciones comparativas. A partir de este entorno y conjunto de datos, se diseñó un proceso experimental orientado a evaluar comparativamente la calidad, factibilidad y comportamiento computacional de los algoritmos propuestos.

El conjunto de datos utilizado corresponde a los horarios de 600 estudiantes de la carrera de Ingeniería Civil Informática y a los 120 ayudantes registrados durante el segundo semestre académico del año 2025 en la UTFSM. Con el objetivo de asegurar transparencia, trazabilidad y reproducibilidad de los resultados, el código fuente del proyecto se encuentra disponible en un repositorio público de GitHub, el cual incluye la implementación de los módulos principales y los experimentos realizados [18]. No obstante, debido al carácter sensible de la información, los datos completos no se encuentran disponibles públicamente, incorporándose únicamente los horarios de las asignaturas INF285 2025-2 e INF295 2025-1 a modo de ejemplo, con previo consentimiento de estudiantes y ayudantes.

A partir de este conjunto de datos y del entorno descrito, se diseñó un proceso de evaluación orientado a validar el desempeño del algoritmo propuesto frente al *Baseline* y al modelo exacto, cuyos resultados se presentan y analizan en las secciones siguientes.

### 4.1 DATOS

Si bien se cuenta con los registros de 600 estudiantes de la carrera, no todas las asignaturas requieren de ayudantías de cátedra. Existen casos como la asignatura INF309 Trabajo de Título, que se desarrolla de manera mayoritariamente autónoma, o INF228 Taller de Desarrollo de Proyectos de Informática, donde el apoyo del ayudante se orienta a cada grupo de estudiantes de forma particular. Por este motivo, y porque se consideraron solo las asignaturas del Departamento de Informática, del total de 134 asignaturas registradas, solo se consideraron aquellas que efectivamente contaban con ayudantías de cátedra durante el semestre, reduciendo el conjunto de análisis a 13 asignaturas.

Los datos correspondientes a los horarios de los estudiantes y ayudantes presentan la estructura mostrada en la Figura 11, donde se utiliza el formato de 10 bloques horarios por día previamente definidos. A partir de estos horarios, se eliminaron las ayudantías ya registradas en el sistema, con el objetivo de analizar exclusivamente la disponibilidad real de los bloques para una nueva asignación.

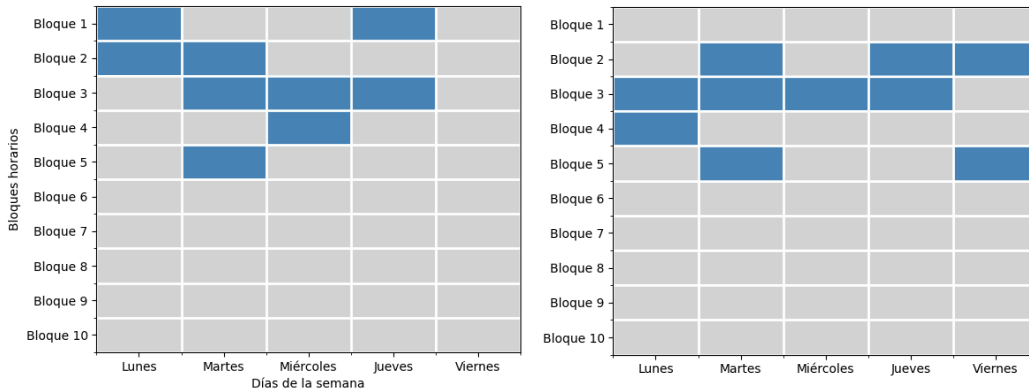


Figura 11: Ejemplo de horarios de estudiantes.  
Fuente: Elaboración propia.

Con todos los horarios procesados, se realizó un análisis de la distribución de la disponibilidad por bloque horario. Como se observa en la Figura 12, los bloques 4 de los días martes y jueves aparecen como completamente disponibles. Esto se debe a que dichos bloques corresponden a los denominados “bloques libres” definidos institucionalmente por la universidad, razón por la cual la asignación de ayudantías en estos espacios queda completamente descartada.

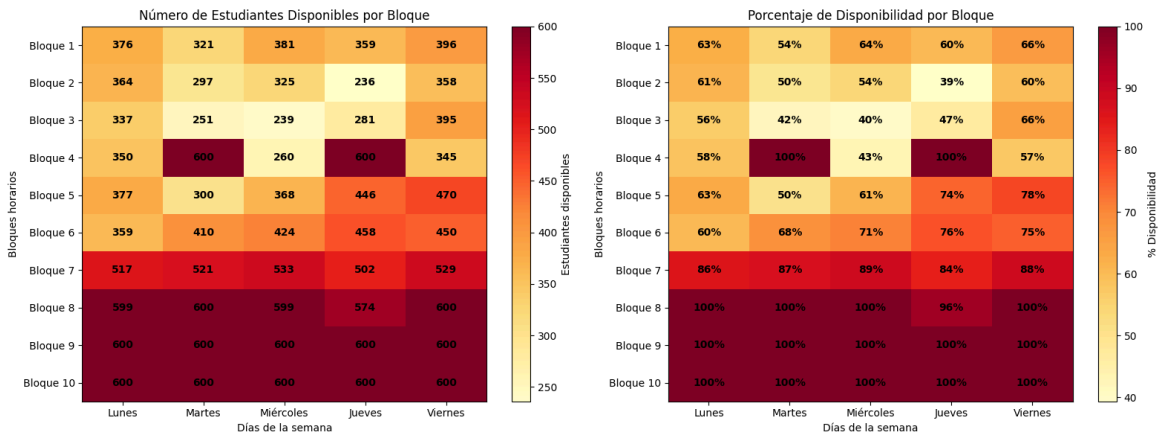


Figura 12: Disponibilidad total por bloque.  
Fuente: Elaboración propia.

Adicionalmente, se realizó un análisis de los horarios de los estudiantes con el fin de caracterizar su distribución de ocupación a lo largo de la semana. Para ello, se construyeron los gráficos de porcentaje de ocupación mostrados en la Figura 13, los cuales permiten observar cómo se distribuyen las clases de los estudiantes tanto por día de la semana como por bloque horario.

Los resultados evidencian que los días presentan una carga relativamente pareja; no obstante, se observa un mayor porcentaje de ocupación en los bloques 5 y 6, los cuales corresponden a los bloques académicos 3-4 y 5-6 de la UTFSM. Este análisis resulta fundamental para comprender los patrones de carga académica.

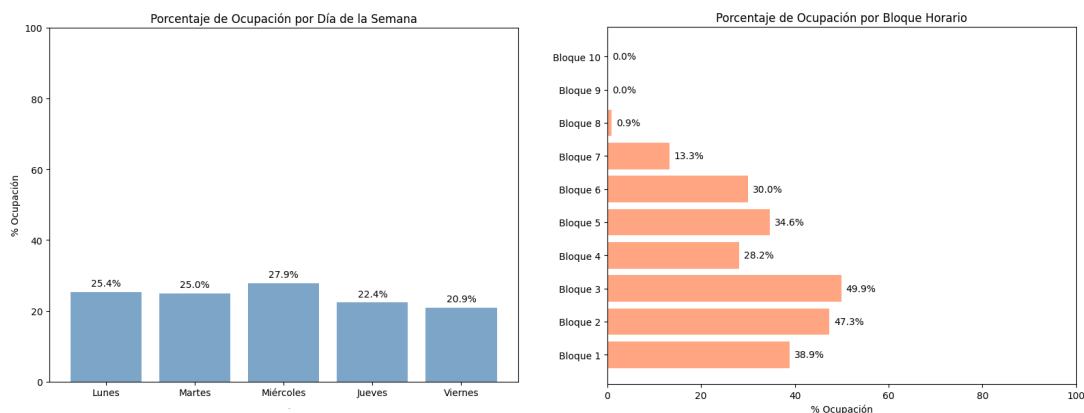


Figura 13: Gráficos de porcentaje de ocupación.

Fuente: Elaboración propia.

Este doble conjunto de datos permite contrastar la información institucional con la disponibilidad real del estudiantado, constituyendo un caso de estudio relevante para analizar las limitaciones del sistema actual y evaluar el impacto de contar con información más precisa en la asignación de horarios de ayudantía.

Antes de continuar, es necesario destacar el caso particular de la asignatura INF285, para la cual se dispone de dos fuentes de información complementarias. Por una parte, se cuenta con los datos oficiales de horarios obtenidos a través del departamento de informática, los cuales forman parte del conjunto de las ayudantías consideradas para este estudio. Adicionalmente, para esta asignatura se obtuvo información directa sobre la disponibilidad horaria proporcionada de los estudiantes inscritos, la cual también incluye compromisos no reflejados en el sistema oficial.

Para ello, se coordinó con el profesor encargado de la cátedra de INF285 segundo semestre de Casa Central, para poder distribuir un horario en blanco, como se muestra en la Figura 14, entre el estudiantado para que pudieran marcar los días y bloques que están ocupados. Esta misma dinámica se realizó en los bloques de ayudantías con los ayudantes. Esto permitió que los estudiantes y ayudantes pudieran ingresar libremente sus bloques con conflictos, independientemente de si estos eran parte del plan académico o no. Esto posibilitó la identificación de estudiantes con bloques no disponibles incluso en el Bloque 9, una circunstancia que no se manifiesta en la información oficial proporcionada por la universidad, en la cual dichos bloques se presentan como disponibles (Figuras 12 y 13).

Bloque	Lunes	Martes	Miércoles	Jueves	Viernes
1-2					
3-4					
5-6					
7-8					
9-10					
11-12					
13-14					
15-16					
17-18					
19-20					

Figura 14: Horario distribuido a los estudiantes y ayudantes.  
Fuente: Elaboración propia.

A partir de esta información, fue posible construir mapas de calor que representan la distribución de la ocupación horaria del estudiantado, los cuales se presentan en la Figura 15. En la cual se puede observar que en los bloques prohibidos solo existe un 94% y 87% de disponibilidad, algo imposible para los datos oficiales. Además, se puede apreciar que en bloques nocturnos que suelen estar libres, existe por lo menos un estudiante que tiene esos bloques ocupados.

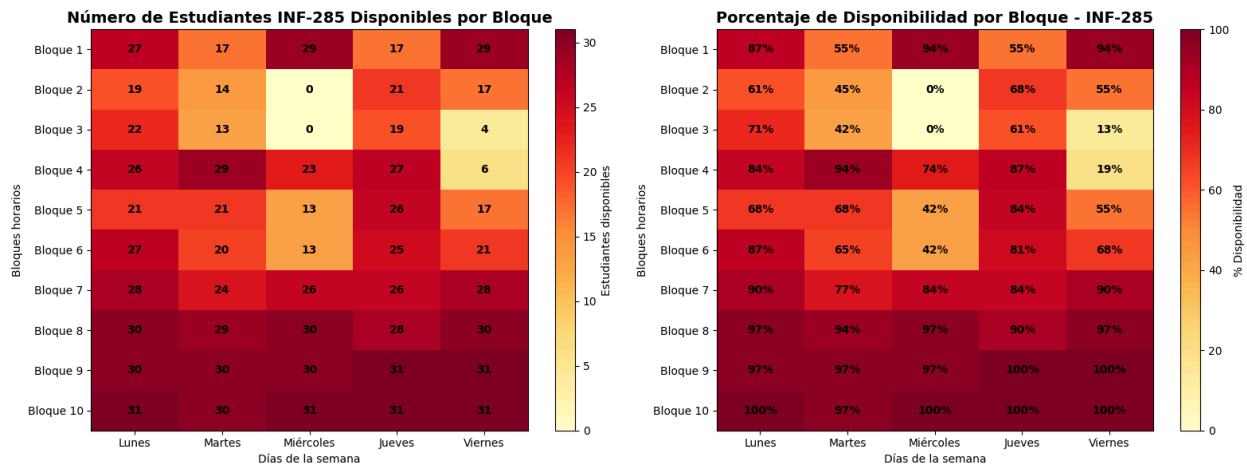


Figura 15: Disponibilidad total por bloque INF285 extraoficial.  
Fuente: Elaboración propia.

Para efectos comparativos, la Figura 16 presenta el mapa de calor construido a partir de los horarios oficiales proporcionados por la institución. A diferencia de la información extraoficial, en estos datos los bloques institucionalmente definidos como libres presentan una disponibilidad del 100%, lo que implica que ningún estudiante registra compromisos en dichos horarios. Asimismo, los bloques nocturnos muestran niveles de disponibilidad completos o cercanos al 100%, configurando un escenario estructuralmente más “parejo” que el observado en la información recopilada directamente desde el estudiantado.

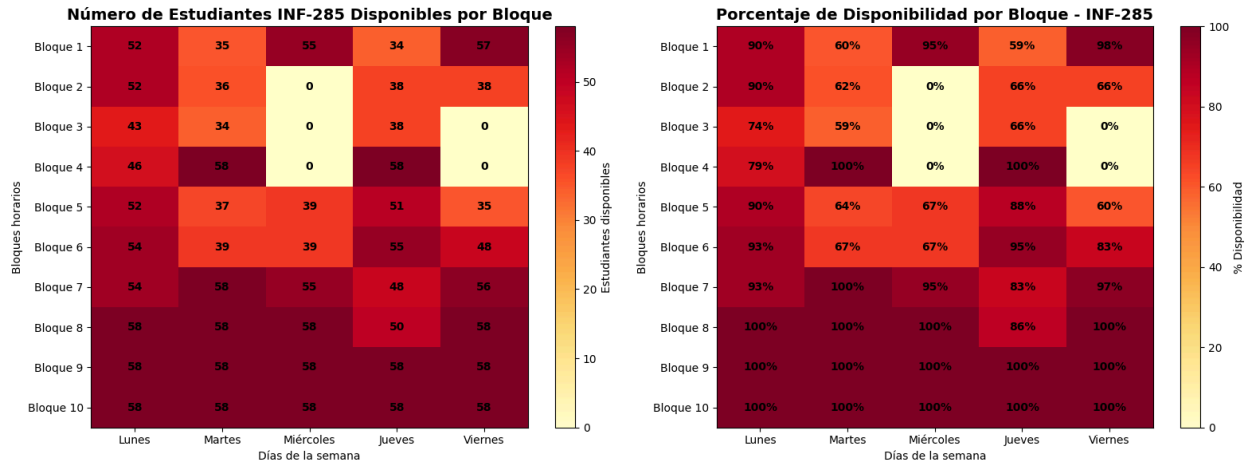


Figura 16: Disponibilidad total por bloque INF285 oficial.

Fuente: Elaboración propia.

El contraste entre ambas figuras evidencia que la información institucional tiende a sobreestimar la disponibilidad real de los estudiantes, al no capturar compromisos extracurriculares, laborales u otras actividades no registradas formalmente. Esta diferencia tiene implicancias directas en la calidad potencial de las asignaciones realizadas bajo supuestos incompletos, ya que un modelo que opera sobre datos idealizados podría generar soluciones factibles en términos formales, pero subóptimas desde la experiencia real del estudiantado.

No obstante, la información extraoficial no se incorpora en el análisis experimental, debido a que no representa al total de los estudiantes inscritos en la asignatura, ya que la recolección se realizó en un contexto de asistencia parcial. Por este motivo, y con el fin de mantener uniformidad entre las asignaturas analizadas, se utilizan exclusivamente los datos obtenidos mediante los sistemas institucionales.

Dado que los datos utilizados corresponden a registros actualizados, se dispone del horario efectivamente asignado para la realización de las ayudantías. Esto permite llevar a cabo un análisis exhaustivo de la efectividad del proceso de asignación actualmente utilizado. Para ello, se identificaron las 13 ayudantías consideradas en el estudio junto con sus respectivos horarios, y se constató esta información con los horarios de los estudiantes inscritos en cada asignatura, evaluando así su potencial de asistencia.

La Tabla 12 representa un resumen comparativo de estas ayudantías, considerando como métricas el porcentaje de participación estudiantil y el valor de la función *fitness*. Esta última evalúa la calidad del horario asignado en función del cumplimiento de las restricciones blandas, las cuales representan preferencias horarias del estudiantado.

Con el objetivo de normalizar el valor de la función *fitness* y facilitar la comparación entre instancias de distinto tamaño, se define el porcentaje de *fitness* (*%Fitness*) como:

$$\%Fitness = \frac{Fitness}{n_E} \times 100,$$

Bajo esta definición, un valor de %Fitness igual a 100 corresponde a una solución ideal, en la cual todos los estudiantes pueden asistir al menos a una ayudantía y no se presenta ninguna penalización asociada a alguna restricción blanda. Valores inferiores reflejan una disminución en la cantidad de asistentes y/o la presencia de penalizaciones relacionadas con la calidad horaria de la asignación.

Tabla 12: Comparación de ayudantías previamente asignadas.  
Fuente: Elaboración propia.

Sigla	Total Estudiantes	Participantes	Con conflicto	% Participación	% Fitness
INF246	58	58	0	100.00	99.02
INF134	35	35	0	100.00	95.48
INF155	80	79	1	98.75	91.79
INF280	59	59	0	100.00	90.90
INF245	33	33	0	100.00	83.23
INF285	58	55	3	94.83	82.39
INF253	64	57	7	89.06	82.06
INF256	18	16	2	88.89	81.02
INF295	53	46	7	86.79	80.75
INF152	35	29	6	82.86	78.10
INF276	53	51	2	96.23	73.96
INF322	72	49	23	68.06	60.81
IWI131	1	0	1	0.00	0.00

Como se puede observar en la Tabla 12, la mayoría de las asignaturas presentan una participación superior al 70%. Cabe destacar el caso de IWI131, asignatura que tiene un solo estudiante registrado. Esto se debe a que los horarios obtenidos solamente son del departamento de informática, además de que IWI131 es una asignatura de la malla curricular antigua, por lo que su cantidad de asistentes cada vez es menor.

Aunque los casos analizados en la tabla tengan una participación alta, el algoritmo propuesto está pensado para asignar ayudantías a asignaturas que ya realizan la asignación una vez comenzada el periodo escolar, es decir, los horarios de los posibles asistentes no deben por qué tener los mismos horarios. Es por esto, que algunas de estas asignaturas presentan un 100% de participación, porque los estudiantes al momento de inscribirse a la asignatura, ya sabían de antemano si la ayudantía les generará un choque horario o no. Sin embargo, casos críticos como INF322, con solo un 68.06% de participación y un 60.81% de *fitness*, demuestran que el sistema manual presenta dificultades.

En conjunto, este análisis de datos permite caracterizar el contexto real en el cual se evalúan los algoritmos propuestos, evidenciando tanto las limitaciones de la información institucional como los desafíos asociados a la asignación manual de ayudantías. Estas observaciones motivaron el diseño del proceso de evaluación descrito en la siguiente sección, orientado a comparar de manera sistemática el desempeño del *Baseline*, el *solver* y el algoritmo *Simulated Annealing*.

## 4.2 PROCESO DE EVALUACIÓN

Para comparar el desempeño del algoritmo *Simulated Annealing* respecto a la solución actualmente utilizada y del algoritmo base, se emplean las siguientes métricas de evaluación:

- **Calidad de solución (conflictos):** se mide a partir de la cantidad de estudiantes que no pueden asistir a la ayudantía asignada, lo que implica el incumplimiento de una o más restricciones duras asociadas a cada estudiante. Esta métrica permite evaluar directamente la factibilidad de la solución generada.
- **Nivel de satisfacción (%Fitness):** evalúa el grado de cumplimiento de las restricciones blandas del problema, las cuales representan criterios de preferencia del estudiantado. Entre estas se consideran, por ejemplo, la generación de ventanas horarias, la asignación en bloques tempranos y otros aspectos definidos previamente en el modelo.

Cada ejecución del algoritmo se inicializa a partir de una solución base generada mediante un algoritmo *greedy*. Las configuraciones iniciales de los parámetros corresponden a las definidas en la sección anterior. Posteriormente, los resultados obtenidos se comparan utilizando las métricas descritas, con el objetivo de identificar el método que presenta el mejor equilibrio entre calidad de la solución y tiempo de ejecución.

Dado el carácter estocástico del algoritmo SA, cada instancia fue ejecutada múltiples veces de manera independiente con el fin de analizar la estabilidad y variabilidad de los resultados. Para cada asignatura se realizaron 1000 ejecuciones, registrando la media, desviación estándar y valores máximos y mínimos del *fitness*.

## 4.3 RESULTADOS EXPERIMENTALES

### 4.3.1 DESEMPEÑO DE SIMULATED ANNEALING CON AYUDANTÍAS PREVIAMENTE ASIGNADAS

Con el objetivo de evaluar el desempeño del algoritmo propuesto, se utilizaron como referencia las ayudantías previamente asignadas, denominadas en este trabajo como *Baseline*. Estas asignaciones, descritas en la Sección 4.1 y resumidas en la Tabla 12, se comparan con las soluciones obtenidas mediante el algoritmo SA, considerando como métricas el porcentaje de *fitness* y la cantidad de conflictos asociados a las restricciones duras.

Los resultados de esta comparación se resumen en la Tabla 13, la cual resume, para cada asignatura analizada, el desempeño del *Baseline* y de la solución generada por SA tanto en términos de porcentaje de *fitness* como del número de conflictos detectados.

Tabla 13: Comparación de ayudantías previamente asignadas.  
Fuente: Elaboración propia.

Sigla	Total Estudiantes	Conflictos Baseline	Conflictos SA	% Fitness Baseline	% Fitness SA
INF246	58	0	0	99.02	96.24
INF134	35	0	0	95.49	96.90
INF155	80	1	0	91.79	99.21
INF280	59	0	0	90.90	93.77
INF245	33	0	0	83.24	92.42
INF285	58	3	0	82.38	94.97
INF253	64	7	0	82.06	97.39
INF256	18	2	0	81.00	85.33
INF295	53	7	0	80.75	94.82
INF152	35	6	0	78.09	86.52
INF276	53	2	1	73.96	84.92
INF322	72	23	0	60.81	92.96
IWI131	1	1	0	0.00	100.00

A partir de los resultados mostrados en la Tabla 13, se observa que el algoritmo *Simulated Annealing* logra reducir, y en la mayoría de los casos eliminar completamente, los conflictos asociados a las restricciones duras. Este comportamiento evidencia una mejora significativa en la factibilidad de las soluciones respecto al *Baseline*, especialmente en aquellas asignaturas que presentan una alta conflictividad inicial.

En términos de porcentaje de *fitness*, el algoritmo propuesto alcanza, en general, valores superiores a los obtenidos por el *Baseline*. Las mejoras son particularmente notorias en aquellas asignaturas donde la solución base presenta un bajo nivel de satisfacción, lo que indica que SA es capaz de refinar de manera efectiva soluciones iniciales de menor calidad.

Con el fin de facilitar la comparación visual del desempeño en términos de satisfacción, la Figura 17 presenta una gráfica comparativa del porcentaje de *fitness* obtenido por el *Baseline* y por el algoritmo SA para cada asignatura analizada.

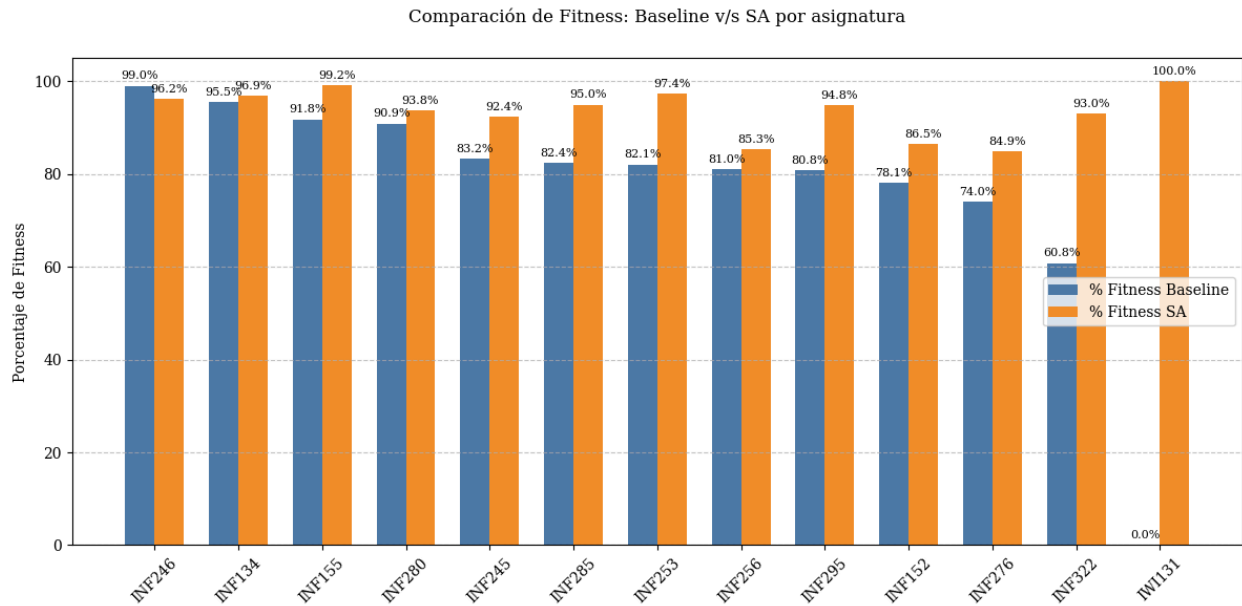


Figura 17: Gráfica comparativa de %Fitness Baseline v/s SA.

Fuente: Elaboración propia.

La Figura 17 refuerza visualmente los resultados presentados anteriormente, mostrando que el algoritmo obtiene, en la mayoría de las asignaturas analizadas, valores de *fitness* superiores a los del *Baseline*. Las mejoras más significativas se observan en aquellas asignaturas donde la solución base presentaba un bajo nivel de satisfacción, lo que sugiere que el algoritmo es particularmente efectivo al refinar soluciones iniciales de baja calidad. En contraste, en los casos donde el *Baseline* ya alcanzaba altos valores de *fitness*, las diferencias entre ambos enfoques son menores o inexistentes, lo cual es coherente con el comportamiento esperado en este tipo de metaheurísticas.

Por otra parte, en la Figura 18 se presenta la comparación del porcentaje de conflictos asociados a restricciones duras entre el *Baseline* y las soluciones obtenidas mediante el algoritmo *Simulated Annealing*.

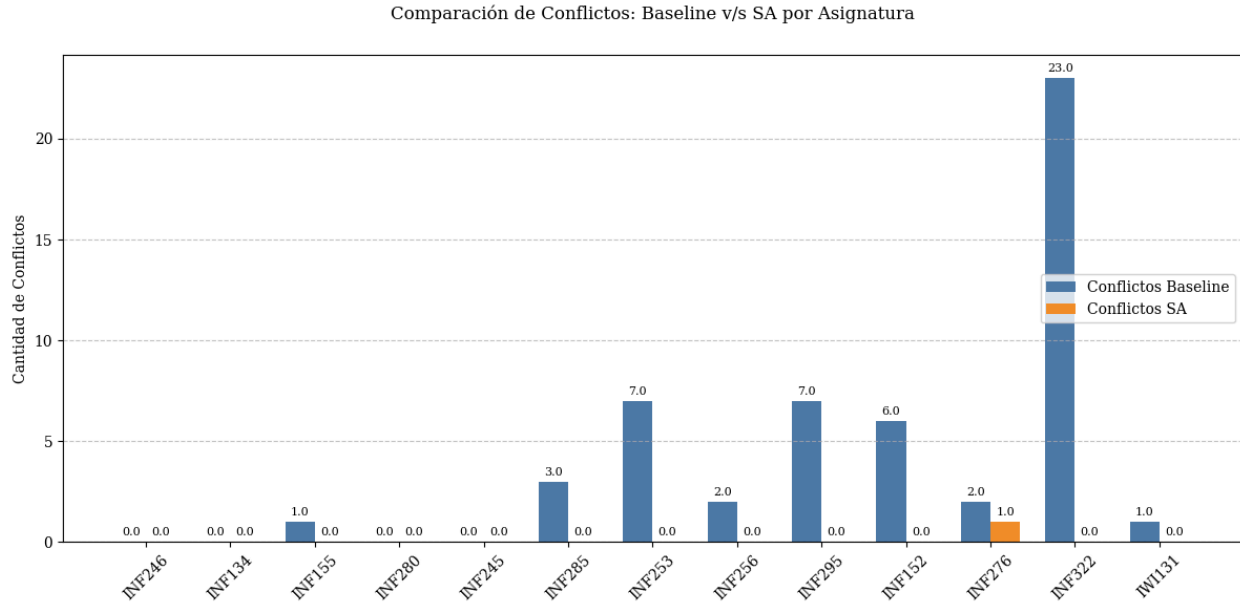


Figura 18: Gráfica comparativa de conflictos *Baseline* v/s SA.  
Fuente: Elaboración propia.

A partir de la Figura 18, se observa que el algoritmo SA logra eliminar prácticamente la totalidad de los conflictos presentes en las asignaciones base, alcanzando valores cercanos a cero en todas las asignaturas analizadas. En contraste, el *Baseline* presenta niveles de conflicto significativos en diversos casos, destacando especialmente aquellas asignaturas con mayor número de estudiantes.

Un caso especialmente representativo corresponde a la asignatura INF322, donde la solución base presenta un alto número de conflictos y un bajo porcentaje de *fitness*, como se observa tanto en la Tabla 13 como en la Figura 18. En contraste, asignaturas como INF246 e INF134, donde el *Baseline* ya corresponde a una solución factible y de alta calidad, no presentan mejoras relevantes al aplicar SA.

En conjunto, estos resultados evidencian la capacidad del algoritmo propuesto para manejar de manera efectiva las restricciones duras del problema, incluso en escenarios con alta conflictividad inicial, garantizando la factibilidad de las soluciones generadas y mejorando la calidad global de las asignaciones de ayudantías.

#### 4.3.2 DESEMPEÑO DE SOLVER CON AYUDANTÍAS PREVIAMENTE ASIGNADAS

Si bien existen *solvers* ampliamente utilizados para problemas de *Educational Timetabling*, como UniTime, estos se encuentran orientados a versiones más generales y complejas del problema, incorporando múltiples tipos de eventos, recursos y restricciones que exceden el alcance de este trabajo. Por este motivo, se optó por utilizar un *solver* más acotado, que permite una comparación directa y controlada con el algoritmo propuesto, manteniendo el foco en el impacto del enfoque metaheurístico sobre el problema específico de asignación de ayudantías.

Con este objetivo, se utiliza la biblioteca PuLP como herramienta de modelado para formular el problema de asignación de ayudantías como un modelo de programación lineal entera, y el *solver* CBC (Coin-or Branch and Cut) para su resolución. Este enfoque se emplea como punto de comparación frente al algoritmo metaheurístico propuesto, permitiendo contrastar el desempeño de una técnica completa, que garantiza soluciones óptimas bajo el modelo de formulado, con una técnica incompleta, orientada a la exploración eficiente del espacio de búsqueda.

La Tabla 14 presenta una comparación cuantitativa entre el *Baseline* y el modelo ILP resuelto mediante el *solver* CBC, considerando como métricas principales el número de conflictos asociados a restricciones duras y el porcentaje de *fitness* obtenido para cada asignatura analizada.

En términos generales, los resultados muestran que el *solver* logra igualar o superar el desempeño del *Baseline* en todas las instancias evaluadas. En particular, se observa que el *solver* elimina completamente los conflictos en la mayoría de asignaturas, incluso en casos donde el *Baseline* presenta un número elevado de conflictos, como en la asignatura INF322. Este comportamiento evidencia la capacidad del modelo ILP para satisfacer de manera estricta las restricciones duras del problema, garantizando soluciones factibles desde el punto de vista de disponibilidad y compatibilidad horaria.

Tabla 14: Comparación de *Baseline* v/s *solver*.

Fuente: Elaboración propia.

Sigla	Total Estudiantes	Conflictos Baseline	Conflictos Solver	% Fitness Baseline	% Fitness Solver
INF246	58	0	0	99.02	99.02
INF134	35	0	0	95.49	96.90
INF155	80	1	0	91.79	99.42
INF280	59	0	1	90.90	94.80
INF245	33	0	0	83.24	95.00
INF285	58	3	0	82.38	96.84
INF253	64	7	0	82.06	99.84
INF256	18	2	0	81.00	85.33
INF295	53	7	0	80.75	96.67
INF152	35	6	0	78.09	86.52
INF276	53	2	0	73.96	88.70
INF322	72	23	0	60.81	96.23
IWI131	1	1	0	0.00	100.00

Respecto al porcentaje de *fitness*, el *solver* obtiene mejoras significativas en comparación con el *Baseline* en prácticamente todas las asignaturas. Destacan especialmente los casos con menor desempeño inicial, como INF322 e IWI131, donde el *Baseline* presenta valores de *fitness* de 60.81% y 0.00%, respectivamente, mientras que el *solver* alcanza valores superiores al 96% e incluso el 100%. Estos resultados indican que el modelo ILP es capaz de identificar configuraciones de ayudantías que maximizan la asistencia estudiantil incluso en escenarios altamente restrictivos.

En asignaturas donde el *Baseline* ya presenta un desempeño elevado, como INF246 e INF134, el *solver* mantiene valores de *fitness* comparables o ligeramente superiores. Esto indica que, en estos casos, la asignación previa ya se encontraba cercana al óptimo bajo la función de

evaluación definida, y que el *solver* actúa principalmente como un mecanismo de validación y refinamiento de la solución existente.

Un aspecto relevante es que las mejoras observadas en el porcentaje de *fitness* no provienen únicamente de la eliminación de conflictos, sino también de una mejor gestión de las restricciones blandas, tales como penalizaciones por ventanas, horarios extremos o asistencia en días sin clases. Dado que estas penalizaciones fueron incorporadas explícitamente en la función objetivo del modelo ILP utilizando los mismos pesos definidos para el algoritmo metaheurístico, los resultados obtenidos permiten una comparación directa y consistente entre ambos enfoques.

En síntesis, los resultados indican que el *solver* proporciona soluciones de alta calidad en términos de asistencia y *fitness*, superando claramente al *Baseline* en los casos más complejos y manteniendo su desempeño competitivo en aquellos donde la asignación inicial ya era favorable. Estos hallazgos validan el uso del modelo ILP como un punto de referencia sólido para evaluar el desempeño del algoritmo propuesto, previo a la comparación directa entre técnicas completas e incompletas.

Al igual que en la Sección 4.3.1, se utiliza una representación gráfica para facilitar el análisis comparativo de las soluciones obtenidas. En la Figura 19 se presenta la comparación del porcentaje de *fitness* alcanzado por el *Baseline* y por el *solver* para cada asignatura analizada.

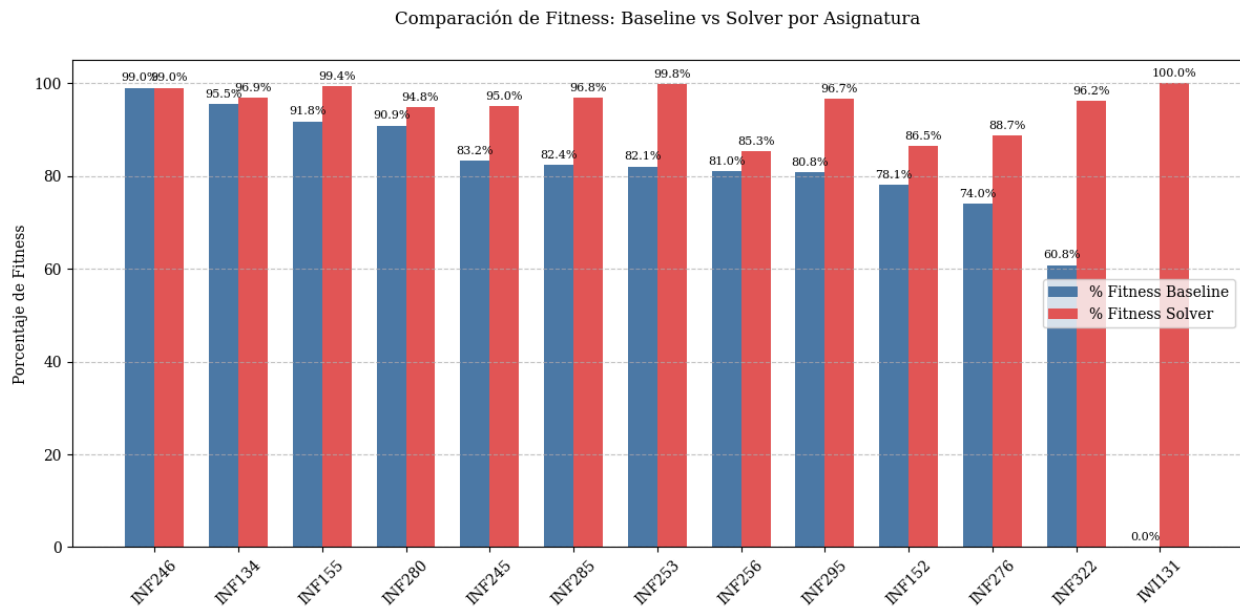


Figura 19: Gráfica comparativa de %Fitness Baseline v/s Solver.

Fuente: Elaboración propia.

La Figura 19 permite visualizar de manera clara y comparativa la diferencia en el porcentaje de *fitness* obtenido por el *Baseline* y el *solver* para cada asignatura analizada. A diferencia de la Tabla 14, esta representación gráfica facilita la identificación de tendencias globales y brechas de desempeño entre ambos enfoques.

En particular, se observa que el *solver* presenta un desempeño consistentemente superior al *Baseline* en la mayoría de las asignaturas, con diferencias especialmente marcadas en aquellas instancias más restrictivas. Casos como INF322, INF295 e INF253 evidencian brechas significativas entre ambas soluciones, donde el *Baseline* alcanza valores de *fitness* considerablemente inferiores, mientras que el *solver* logra valores cercanos al óptimo.

Asimismo, la figura muestra que incluso en asignaturas donde el *Baseline* presenta un buen desempeño inicial, el *solver* mantiene o mejora ligeramente el porcentaje de *fitness*, lo que refuerza la idea de que el modelo ILP no solo corrige asignaciones deficientes, sino que también refina soluciones previamente aceptables.

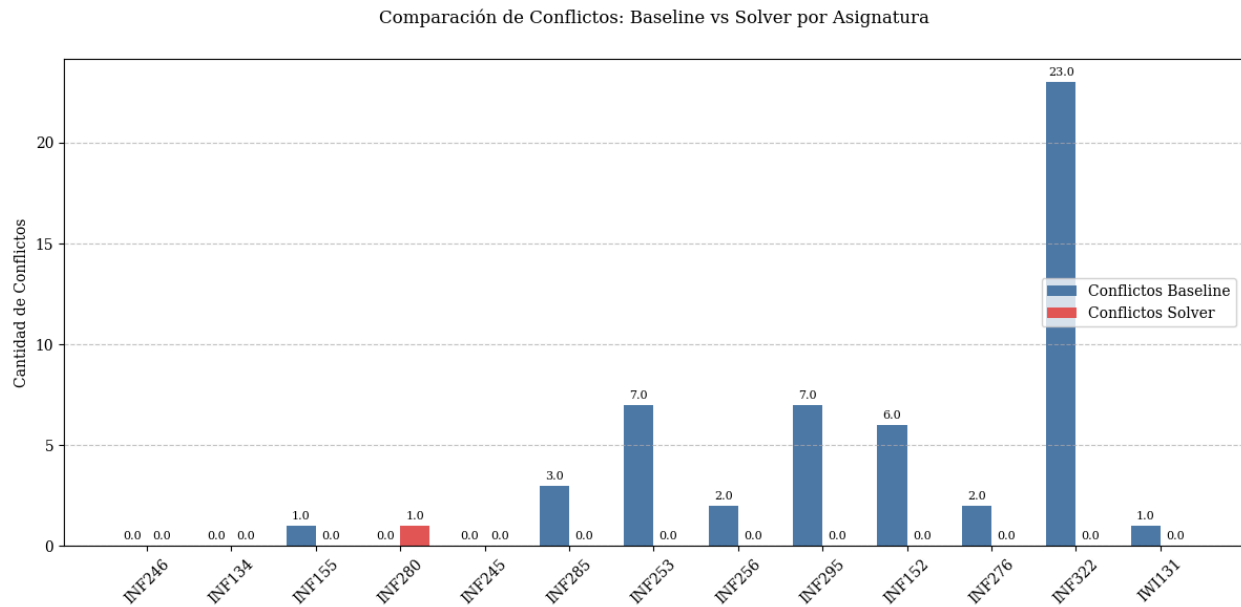


Figura 20: Gráfica comparativa de conflictos *Baseline* v/s Solver.

Fuente: Elaboración propia.

Finalmente, la Figura 20 presenta la comparación del número de conflictos asociados a restricciones duras entre el *Baseline* y el *solver* para cada asignatura. En esta figura se observa que el *Baseline* presenta conflictos en múltiples instancias, alcanzando valores particularmente altos en asignaturas como INF322, lo que evidencia la dificultad de satisfacer simultáneamente las restricciones del problema mediante asignaciones predefinidas.

En contraste, el *solver* logra eliminar prácticamente la totalidad de los conflictos en todas las asignaturas analizadas, confirmando que las mejoras observadas en el porcentaje de *fitness* no corresponden a una compensación entre calidad y factibilidad, sino a una solución que satisfacen de manera efectiva las restricciones del modelo. Este resultado refuerza la validez del *solver* como punto de comparación para los métodos metaheurísticos presentados posteriormente.

En conjunto, los resultados presentados en las secciones anteriores permiten evaluar de manera comparativa el desempeño del algoritmo metaheurístico propuesto y del modelo de

programación lineal entera frente a las asignaciones previamente establecidas. Mientras que el *solver* actúa como un referente de solución óptima bajo el modelo formulado, el algoritmo de *Simulated Annealing* demuestra una alta capacidad para generar soluciones factibles y de calidad en escenarios con alta conflictividad inicial. No obstante, este análisis se ha centrado en evaluaciones agregadas por asignatura sin profundizar en el comportamiento de las soluciones a nivel individual.

Con el fin de ilustrar de manera más detallada el impacto práctico de los métodos evaluados, más adelante se presenta un caso de estudio representativo en el cual se analiza en profundidad una asignatura específica. Este enfoque permite observar cómo se manifiestan las mejoras en términos de conflictos, *fitness* y distribución horaria, complementando el análisis cuantitativo global con una evaluación cualitativa de las soluciones obtenidas. No obstante, se considera relevante analizar la variabilidad del algoritmo SA antes de visualizar ese análisis.

### 4.3.3 ANÁLISIS DE VARIABILIDAD DE ALGORITMOS

Dado el carácter estocástico del algoritmo *Simulated Annealing*, se realizó un análisis de estabilidad con el objetivo de evaluar la variabilidad de sus resultados frente a múltiples ejecuciones independientes. Se consideraron 12 asignaturas, excluyendo a IW1131 ya que se considera un outlier<sup>11</sup> que para este experimento corresponde a un gasto de recursos innecesarios. Por lo que para esas 12 asignaturas el algoritmo fue ejecutado 1000 veces bajo las mismas condiciones paramétricas, registrando las métricas de valor *fitness*, número de estudiantes asignados y tiempo de ejecución.

A diferencia del *solver* exacto CBC, cuyo comportamiento es determinista en términos de calidad de solución, presentando una desviación estándar nula en el valor *fitness*, SA puede producir soluciones distintas en cada ejecución debido a la naturaleza probabilística de su proceso de exploración del espacio de búsqueda.

---

<sup>11</sup> Datos extremos que difieren mucho del resto de los datos.

Tabla 15: Comparación de calidad de solución SA v/s *solver*.  
Fuente: Elaboración propia.

Instancia	Estudiantes	Ayudantes	SA Fitness ( $\mu \pm \sigma$ )	CBC Fitness	Gap %
INF134	35	2	32.77 $\pm$ 0.73	33.92	3.39
INF152	35	1	30.07 $\pm$ 0.39	30.28	0.69
INF155	80	4	78.45 $\pm$ 0.41	79.53	1.36
INF245	33	2	30.39 $\pm$ 0.63	31.35	3.06
INF246	58	2	54.9 $\pm$ 1.51	57.43	4.41
INF253	64	4	62.8 $\pm$ 0.41	63.90	1.72
INF256	18	1	15.19 $\pm$ 0.31	15.36	1.10
INF276	53	1	46.11 $\pm$ 1.42	47.01	1.92
INF280	59	2	54.64 $\pm$ 1.04	55.93	2.31
INF285	58	2	55.32 $\pm$ 0.37	56.17	1.52
INF295	53	2	50.36 $\pm$ 0.39	51.24	1.72
INF322	72	2	67.32 $\pm$ 1.22	69.28	2.83
<b>Promedio</b>	–	–	–	–	<b>2.17</b>

La Tabla 15 presenta, para cada instancia, el número de estudiantes, la cantidad de ayudantes disponibles y el valor promedio del *fitness* obtenido por ambos algoritmos. En el caso de SA, se reporta la media y desviación estándar ( $\mu \pm \sigma$ ) calculada a partir de múltiples ejecuciones independientes, lo que permite evaluar la estabilidad del método metaheurístico. Del mismo modo, se incluye el porcentaje de brecha (*Gap %*) respecto del valor obtenido por CBC, el cual se considera como referencia de solución óptima o cuasi-óptima.

A partir de esta tabla se observa que el algoritmo SA logra valores de *fitness* cercanos a los obtenidos por el *solver* exacto en la totalidad de las instancias evaluadas. Los porcentajes *gap* se mantienen en niveles reducidos, lo que indica que el algoritmo metaheurístico es capaz de aproximarse de manera consistente a la solución óptima. En particular, incluso en instancias con un mayor número de estudiantes la diferencia relativa permanece acotada, evidenciando una adecuada capacidad de escalabilidad. Esta tendencia puede visualizarse con mayor claridad en la Figura 21, donde se presenta el *gap* porcentual de SA respecto a CBC para cada instancia.

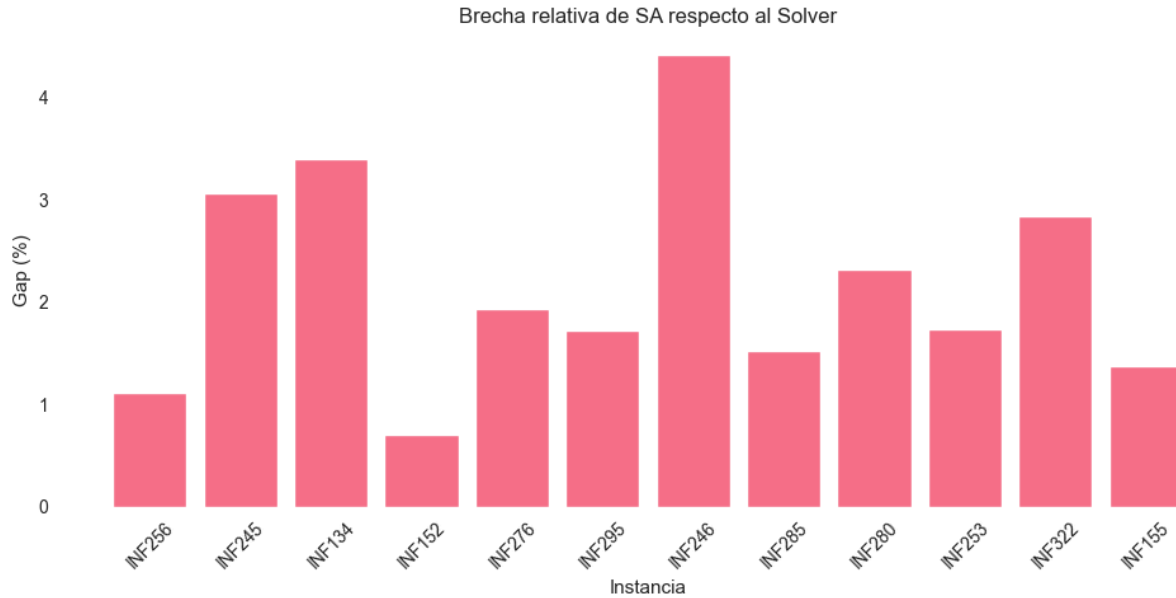


Figura 21: Brecha porcentual (*Gap* %) de SA respecto al *solver* por instancia.  
Fuente: Elaboración propia.

Como se observa en la Figura 21, el *gap* relativo se mantiene acotado en todas las instancias, sin evidenciar un deterioro significativo en instancias de mayor tamaño. Esto refuerza la capacidad del algoritmo SA para mantener calidad de solución incluso en contextos más exigentes.

Desde el punto de vista de la estabilidad, las desviaciones estándar de SA son bajas en comparación con los valores promedio de *fitness*, lo que sugiere que pese a la naturaleza estocástica, el método converge recurrentemente hacia soluciones de una calidad similar.

La estabilidad del algoritmo puede apreciarse gráficamente en la Figura 22, donde se representan los valores promedio de *fitness* junto con su desviación estándar.

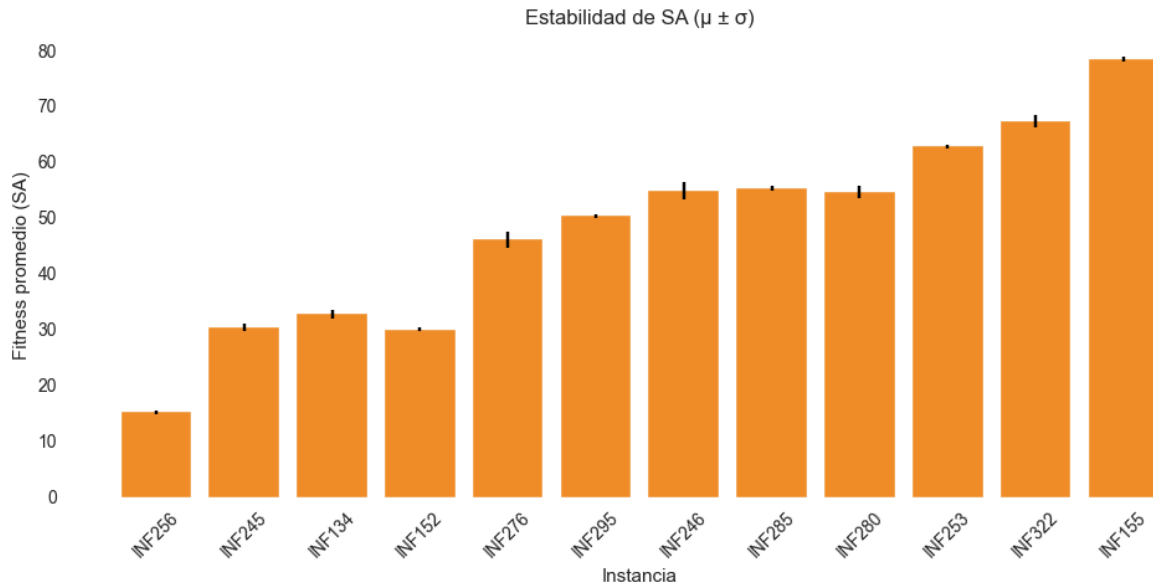


Figura 22: *Fitness* promedio y desviación estándar de SA por instancia.  
Fuente: Elaboración propia.

Se observa que las barras de error presentan magnitudes reducidas en relación con el valor promedio, lo que confirma la consistencia del algoritmo frente múltiples ejecuciones independientes.

Por su parte, la Tabla 16 resume los tiempos de ejecución promedio ( $\mu \pm \sigma$ ) de ambos algoritmos para cada instancia. Esta comparación permite analizar no solo la calidad de las soluciones obtenidas, sino también el costo computacional asociado a cada enfoque, evidenciando el compromiso entre precisión y eficiencia temporal.

Tabla 16: Comparación de tiempos de ejecución SA v/s *solver*.  
Fuente: Elaboración propia.

Instancia	SA Tiempo ( $\mu \pm \sigma$ ) [s]	CBC Tiempo ( $\mu \pm \sigma$ ) [s]
INF134	0.092 ± 0.014	0.499 ± 0.057
INF152	0.097 ± 0.012	0.186 ± 0.015
INF155	0.191 ± 0.022	1.473 ± 0.153
INF245	0.132 ± 0.057	0.926 ± 0.162
INF246	0.241 ± 0.072	0.858 ± 0.285
INF253	0.273 ± 0.047	0.540 ± 0.056
INF256	0.077 ± 0.010	0.148 ± 0.018
INF276	0.231 ± 0.024	0.450 ± 0.046
INF280	0.248 ± 0.035	6.982 ± 0.483
INF285	0.246 ± 0.029	1.132 ± 0.092
INF295	0.252 ± 0.021	0.659 ± 0.077
INF322	0.338 ± 0.026	3.269 ± 0.188

En la tabla se puede apreciar una diferencia significativa entre los algoritmos. SA presenta tiempos promedio considerablemente menores que CBC en la mayoría de las instancias,

especialmente en aquellas de mayor tamaño. Mientras el *solver* garantiza optimalidad, su costo temporal aumenta de forma más pronunciada en comparación con SA, que entrega soluciones cercanas al óptimo con un costo computacional sustancialmente menor.

Con el fin de analizar el comportamiento temporal de los algoritmos en función del tamaño del problema, la Figura 23 presenta la distribución de los tiempos de ejecución agrupados por categoría de número de estudiantes.

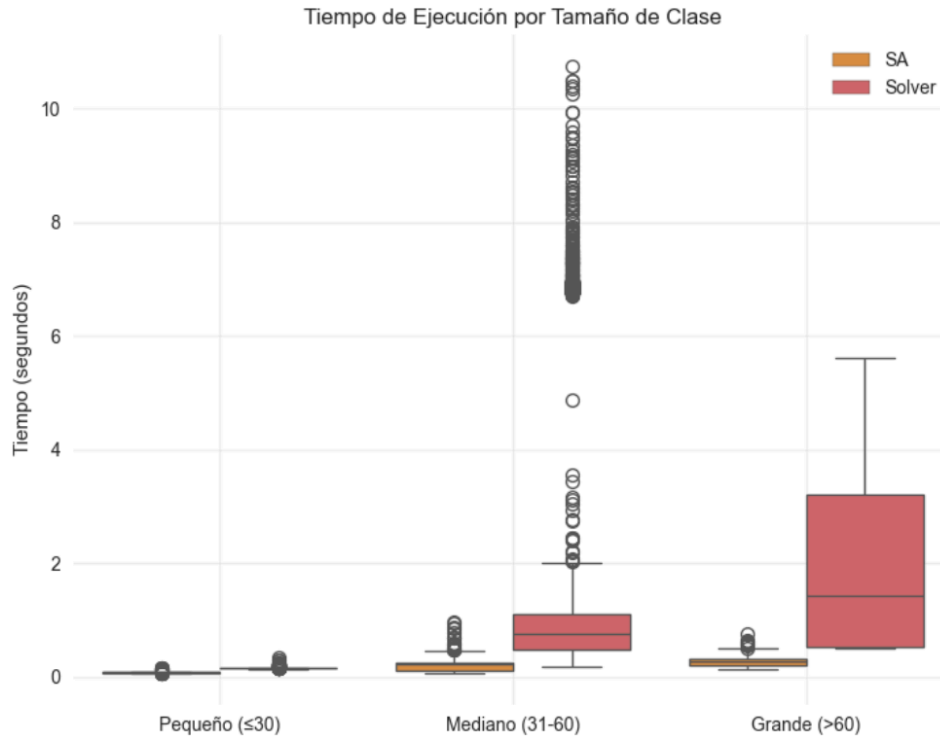


Figura 23: Comparación de tiempos promedio de ejecución entre SA y Solver.

Fuente: Elaboración propia.

La Figura 23 presenta la distribución de los tiempos de ejecución agrupados según el tamaño de la instancia. Se observa que, en clases pequeñas, ambos algoritmos presentan tiempos acotados y con baja dispersión. Sin embargo, a medida que aumenta el tamaño de la clase, el *solver* exhibe una mayor variabilidad temporal, evidenciada por un incremento en el rango intercuartílico y la presencia de múltiples valores atípicos.

En contraste, el algoritmo SA mantiene una distribución más concentrada y estable en todas las categorías de tamaño, con una dispersión considerablemente menor. Esta diferencia sugiere que el *solver* es más sensible a la complejidad estructural del problema, mientras que SA presenta un comportamiento temporal más predecible.

#### 4.4 CASO DE ESTUDIO

Como caso de estudio se presenta la asignatura INF295, correspondiente a la razón por la cual se desarrolla esta memoria. Esta asignatura cuenta con un total de 53 estudiantes y, como se mostró previamente en la Tabla 13, la solución *Baseline* presenta 7 conflictos asociados a

restricciones duras, alcanzando un *fitness* porcentual de 80.75%. En contraste, el algoritmo *Simulated Annealing* genera una solución factible y con un mejor desempeño global, llegando a un *fitness* porcentual de 96.67%.

Con el objetivo de contextualizar los resultados obtenidos por el algoritmo, a continuación se presenta un análisis detallado de la solución *Baseline* para la asignatura elegida. Este análisis permite caracterizar el estado inicial del problema y facilita una comparación directa con la solución propuesta, más allá de las métricas de *fitness* y número de conflictos.

En la Figura 24 se presentan las gráficas correspondientes al análisis del *Baseline*, incluyendo el desglose de penalizaciones por estudiante, el *fitness* porcentual individual, el promedio de penalización por tipo de restricción blanda y la distribución del *fitness* por rangos de calidad.

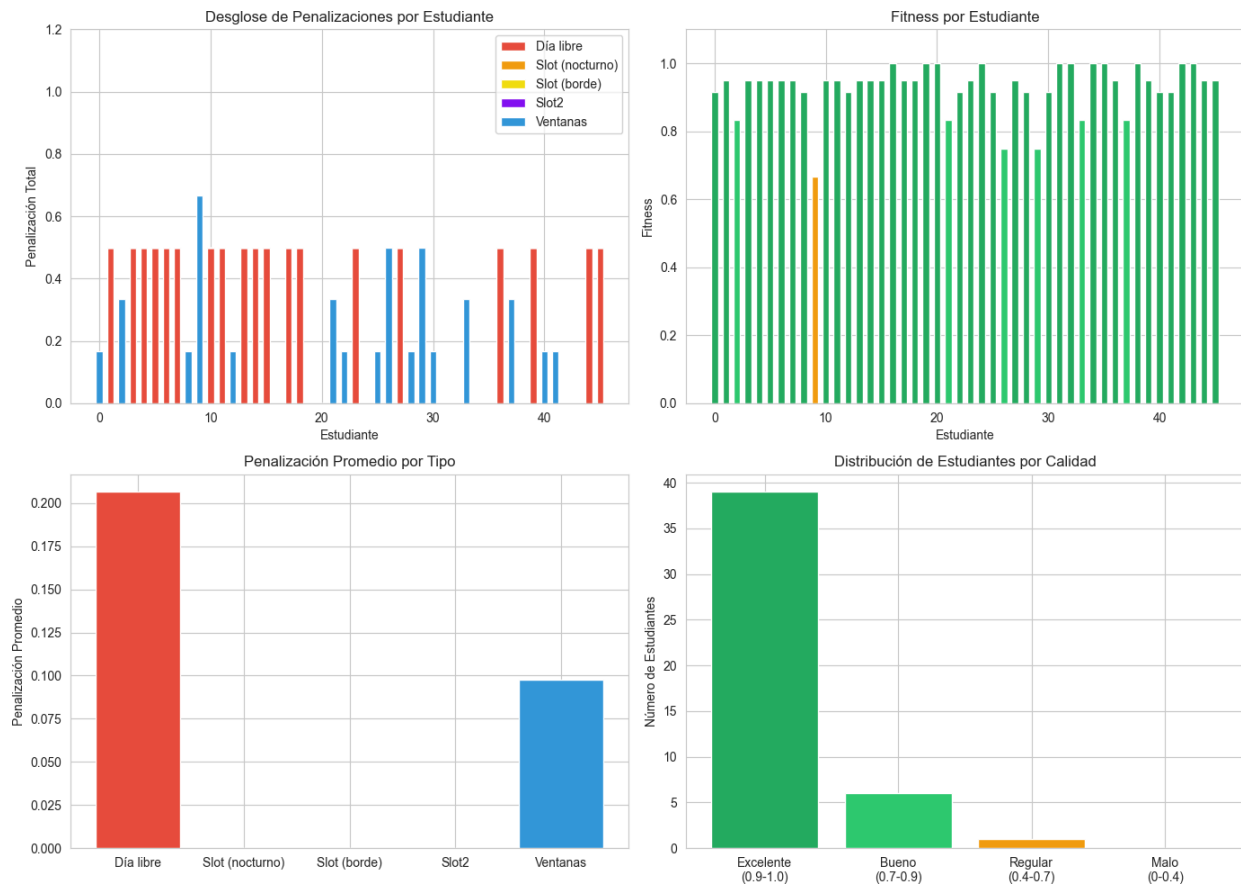


Figura 24: Gráficas de análisis de *Baseline*, INF295.

Fuente: Elaboración propia.

Como se puede observar, la primera gráfica de la Figura 24 muestra los diferentes tipos de penalizaciones que presentan los estudiantes. Cabe aclarar que solamente los estudiantes que cumplen con las restricciones duras, es decir, que pueden asistir a la ayudantía pueden experimentar una penalización. Es por este motivo que en las gráficas superiores se consideran únicamente 46 estudiantes.

Los valores de *fitness* de los estudiantes son en su mayoría «Excelente» con un valor mayor a 0.9, teniendo algunos con un *fitness* entre 0.7 y 0.9, considerados como buenos y solo en un caso un *fitness* regular entre 0.4 y 0.7. Los tipos de penalizaciones corresponden exclusivamente a la asignación de ayudantías en días libres o a la generación de ventanas horarias, sin que ningún estudiante experimente más de una penalización simultánea. Esto sugiere que, si bien la solución *Baseline* presenta conflictos a nivel global, el impacto individual sobre los estudiantes es relativamente acotado, lo que la convierte en una buena solución inicial para ser refinada por el algoritmo *Simulated Annealing*.

Si bien el análisis del *Baseline* muestra que la solución inicial presenta un nivel de satisfacción individual aceptable para la mayoría de los estudiantes, la existencia de conflictos asociados a restricciones duras y la concentración de penalizaciones en determinados casos evidencian oportunidades claras de mejora. En este contexto, el uso del algoritmo SA permite refinar la asignación inicial con el objetivo de mejorar la calidad global de la solución sin deteriorar significativamente el bienestar individual. A continuación, en la Figura 25, se presenta un análisis detallado de una solución obtenida mediante el algoritmo *Simulated Annealing* para la asignatura INF295, utilizando las mismas métricas empleadas en el análisis del *Baseline*, lo que permite una comparación directa entre ambos enfoques.

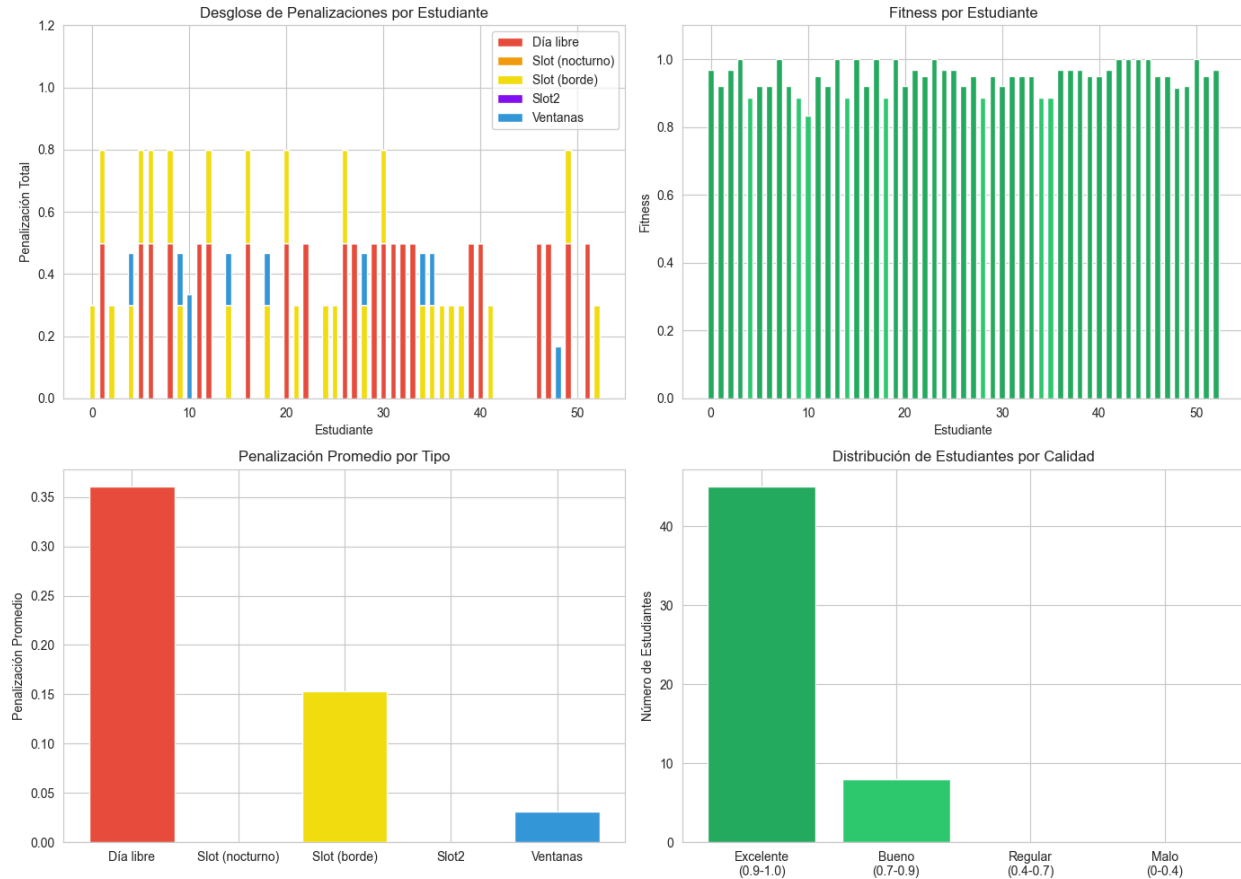


Figura 25: Gráficas de análisis de *Simulated Annealing*, INF295.  
Fuente: Elaboración propia.

La Figura 25 presenta el mismo conjunto de métricas utilizadas en el análisis del *Baseline*, lo que permite evaluar de manera directa y consistente los efectos del algoritmo *Simulated Annealing* sobre la asignación de ayudantías en la asignatura INF295. A diferencia del análisis previo, en este caso se consideran los 53 estudiantes inscritos en la asignatura, correspondientes al total de la población. Las métricas analizadas incluyen el desglose de penalizaciones por estudiante, el *fitness* porcentual individual, la penalización promedio por tipo de restricción blanda y la distribución del *fitness* por rangos de calidad.

El desglose de penalizaciones por estudiante, mostrado en la primera gráfica de la Figura 25, evidencia que un grupo reducido de cuatro estudiantes concentra las penalizaciones más elevadas. Estas se asocian principalmente a la asignación de ayudantías en bloques borde, correspondientes a los bloques 1-2 y 15-16, así como a la realización de ayudantías en días sin otras actividades académicas, lo que implica desplazamientos exclusivos para la asistencia a la ayudantía. No obstante, estas penalizaciones se concentran en casos puntuales y no se distribuyen de manera generalizada entre la población estudiantil.

La gráfica de penalización promedio por tipo de restricción blanda muestra que la solución obtenida presenta una baja incidencia de penalizaciones asociadas a bloques nocturnos, bloques adyacentes y generación de ventanas horarias. La restricción con mayor peso promedio

corresponde a la asignación de ayudantías en días libres. Sin embargo, este valor se mantiene bajo en relación con el total de estudiantes, lo que indica que dicha penalización afecta únicamente a un subconjunto reducido de la población y no compromete la calidad global de la solución.

Por otro lado, las gráficas de *fitness* individual y de distribución por rangos de calidad evidencian que la totalidad de los estudiantes alcanza un *fitness* superior a 0.7, concentrándose mayoritariamente en el rango de calidad «Excelente» (0.9 - 1.0). Este resultado indica que, a pesar de la presencia de penalizaciones individuales en casos específicos, la solución generada mediante *Simulated Annealing* logra mantener un alto nivel de satisfacción global, demostrando un desempeño sólido para la asignatura INF295.

Adicionalmente, se realiza el mismo análisis utilizando el solver propuesto, el cual, como se muestra en la Tabla 14, alcanza un *fitness* porcentual de 96.67%, lo que representa una mejora del 1.85% respecto a la solución obtenida por SA y del 15.92% en comparación con el *Baseline*. En la Figura 26 se presenta el análisis detallado de esta solución, considerando las mismas métricas previamente descritas, lo que permite una comparación directa entre ambos enfoques.

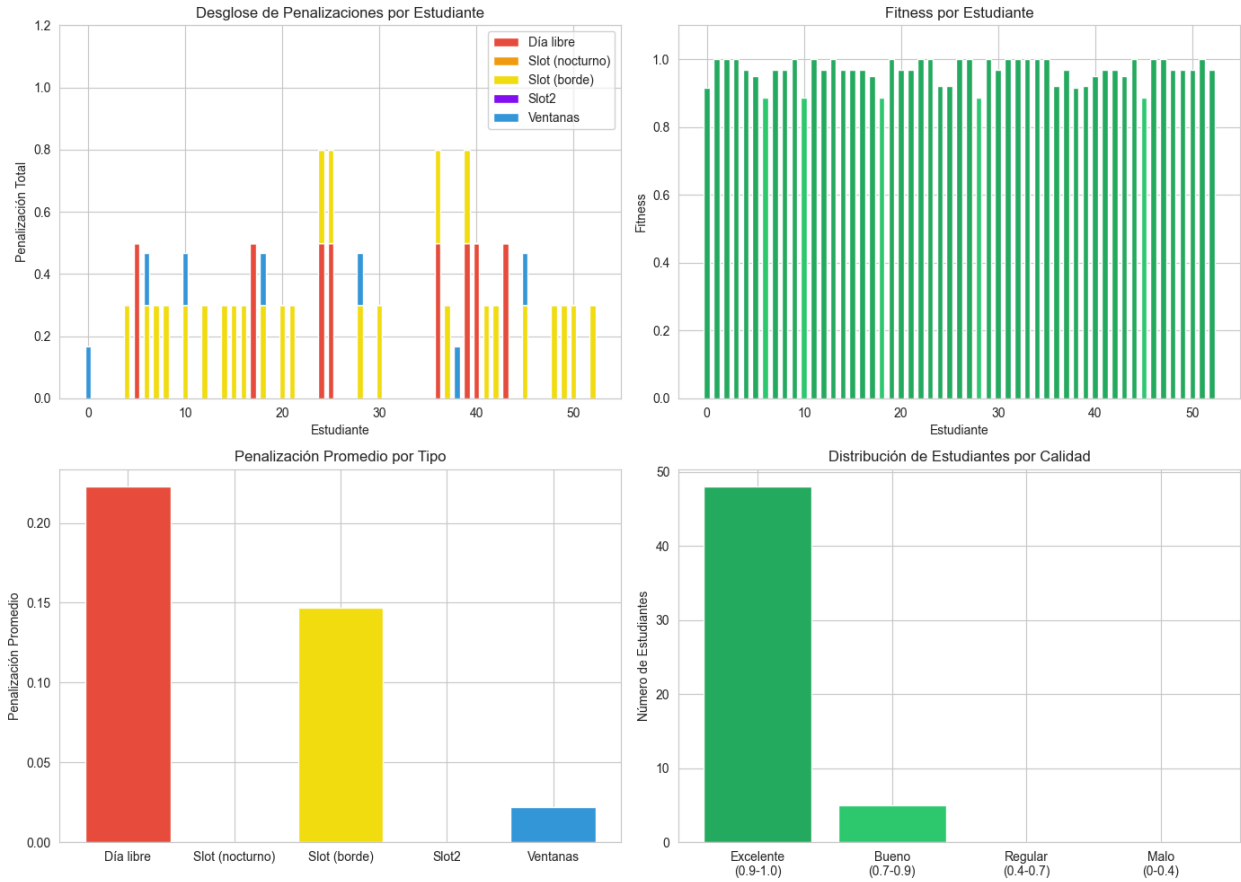


Figura 26: Gráficas de análisis de *Solver*, INF295.

Fuente: Elaboración propia.

El desglose de penalizaciones por estudiante correspondiente a la solución del *solver* muestra una distribución acotada y relativamente homogénea. Las penalizaciones más frecuentes se asocian a la asignación en días sin clases y a *slots* bordes, mientras que aquellas vinculadas a ventanas horarias y al uso del segundo bloque presentan una incidencia mínima. Este comportamiento sugiere que el *solver* tiende a generar soluciones compactas y consistentes, evitando configuraciones fragmentadas o con horarios poco deseables.

El análisis del *fitness* individual muestra que la gran mayoría de los estudiantes alcanza valores cercanos al máximo posible, concentrándose principalmente en el rango [0.9, 1.0]. No se observan casos de *fitness* bajo, lo que indica que el *solver* no maximiza la función objetivo a costa de perjudicar subconjuntos específicos de estudiantes, sino que mantiene un alto nivel de calidad individual de manera consistente.

En cuanto a la penalización promedio por tipo de restricción, se observa que las restricciones asociadas a días libres y *slots* bordes concentran la mayor contribución a la penalización total. Este resultado refleja principalmente limitaciones inherentes a la estructura base de los horarios de los estudiantes, más que deficiencias en el modelo de optimización. En contraste, las penalizaciones por ventanas horarias y *slots* nocturnos son prácticamente nulas, lo que confirma la efectividad del *solver* para evitar configuraciones horarias desfavorables.

Finalmente, la distribución de estudiantes por rangos de calidad refuerza los resultados anteriores, mostrando que prácticamente la totalidad de los estudiantes se clasifica en la categoría «Excelente», con un pequeño grupo en la categoría «Bueno» y ausencia total de soluciones «Regulares» o «Malas». Esto evidencia que el solver es capaz de producir soluciones de alta calidad de manera robusta, incluso cuando se consideran penalizaciones blandas que reflejan preferencias y condiciones reales del estudiantado.

En síntesis, el análisis detallado de la asignatura INF295 permite observar cómo los distintos enfoques abordados en este trabajo impactan no solo en métricas globales, sino también en la experiencia individual de los estudiantes. Mientras que la solución *Baseline* presenta un nivel de satisfacción aceptable pero con conflictos relevantes, el algoritmo *Simulated Annealing* logra eliminar dichos problemas manteniendo un alto nivel de calidad individual. Por su parte, el *solver* obtiene una mejora marginal adicional en el *fitness* global, produciendo una solución más homogénea y cercana al óptimo bajo el modelo formulado. Este caso de estudio ilustra de manera concreta los compromisos entre flexibilidad, optimalidad y esfuerzo computacional asociados a cada enfoque.

#### 4.5 DISCUSIÓN DE RESULTADOS

En este capítulo se presentaron y analizaron los resultados experimentales obtenidos a partir de la aplicación del algoritmo *Simulated Annealing* y del *solver* exacto CBC, modelado mediante programación lineal entera al problema de asignación de ayudantías, considerando tanto escenarios con ayudantías previamente asignadas como un análisis detallado mediante el caso de estudio representativo INF295.

A través de la comparación con la solución *Baseline*, se evidenció que los algoritmos propuestos son capaces de reducir significativamente, y en la mayoría de los casos de eliminar, los conflictos asociados a las restricciones duras, garantizando la factibilidad de las soluciones generadas. Asimismo, los resultados muestran que estos logran mejorar de manera consistente el porcentaje de *fitness*, reflejando un mayor grado de satisfacción de las restricciones blandas del problema, especialmente en instancias con alta conflictividad inicial.

El caso de estudio de la asignatura INF295 permitió profundizar en el comportamiento de los algoritmos a nivel individual, mostrando cómo la solución generada distribuye de forma más equilibrada las penalizaciones entre los estudiantes, manteniendo altos niveles de *fitness* individual y mejorando la calidad global de la asignación. Este análisis evidenció que, si bien la solución *Baseline* presenta un desempeño aceptable para una parte importante de los estudiantes, el algoritmo y el *solver* propuesto logran refinar dicha solución, eliminando conflictos y reduciendo penalizaciones sin comprometer el bienestar individual.

Los resultados experimentales permiten establecer una comparación integral entre el enfoque exacto basado en Programación Lineal Entera y el algoritmo metaheurístico *Simulated Annealing*, tanto en términos de calidad de solución como de comportamiento computacional. No obstante, es importante señalar que el análisis se realizó sobre un conjunto acotado de

asignaturas pertenecientes a una misma carrera, lo que futuras investigaciones podrían ampliar la evaluación a otros contextos institucionales.

Desde el punto de vista de calidad, se observa que SA logra aproximarse consistentemente a las soluciones obtenidas por el *solver* exacto, manteniendo un *gap* porcentual reducido en todas las instancias evaluadas. Esto sugiere que el espacio de soluciones del problema presenta regiones de alta calidad relativamente accesibles para un método de búsqueda estocástico adecuadamente parametrizado.

El análisis de variabilidad presentado en la Sección 4.3.3 refuerza esta conclusión, evidenciando que, pese a su carácter probabilístico, SA exhibe una convergencia estable hacia soluciones de calidad similar entre ejecuciones independientes. Este comportamiento resulta particularmente relevante en problemas reales donde la reproducibilidad y la confiabilidad son factores críticos.

En términos computacionales, los resultados muestran una diferencia estructural entre ambos algoritmos. Mientras que el *solver* exacto garantiza optimalidad bajo el modelo formulado, su desempeño temporal presenta mayor sensibilidad frente a la complejidad estructural de ciertas instancias. En contraste, SA mantiene tiempos de ejecución reducidos y una variabilidad acotada, lo que lo convierte en una alternativa especialmente atractiva en escenarios donde el tiempo de respuesta constituye una restricción operativa.

Desde una perspectiva metodológica, ambos enfoques cumplen roles complementarios. El *solver* actúa como referencia de calidad y herramienta de validación del modelo, permitiendo establecer un límite superior práctico para la función objetivo. Por otro lado, SA ofrece mayor flexibilidad ante modificaciones del problema, dado que la incorporación o ajuste de restricciones blandas puede realizarse directamente sobre la función de evaluación, sin requerir una reformulación matemática completa.

En conjunto, la evidencia experimental sugiere que el algoritmo *Simulated Annealing* constituye una solución eficiente, estable y escalable para el problema de asignación de ayudantías, especialmente en contextos donde la dinámica institucional exige adaptabilidad y tiempos de respuesta acotados. Haciendo del algoritmo SA una alternativa práctica y robusta para apoyar procesos institucionales de asignación de ayudantías, ofreciendo un equilibrio efectivo entre calidad de solución, estabilidad y eficiencia computacional.

## CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

El presente trabajo abordó el problema de asignación de horarios de ayudantía en la Universidad Técnica Federico Santa María desde la perspectiva del *Educational Timetabling Problem*, proponiendo un enfoque de optimización tanto con el algoritmo *Simulated Annealing* como el *solver COIN-OR CBC*. A partir de una modelación formal del problema, se diseñó una representación matricial capaz de incorporar simultáneamente restricciones duras, asociadas a la factibilidad de la asignación, y restricciones blandas, relacionadas con las preferencias y condiciones de bienestar de los estudiantes, permitiendo evaluar la calidad de las soluciones de manera integral.

Los resultados experimentales obtenidos al compararse con el *Baseline* evidencian que el algoritmo *Simulated Annealing* es capaz de generar soluciones factibles que eliminan o reducen significativamente los conflictos asociados a las restricciones duras, superando de manera consistente a la solución *Baseline* utilizada actualmente. Asimismo, se observó una mejora sostenida en el porcentaje de *fitness*, lo que refleja un mayor grado de cumplimiento de las restricciones blandas y, por ende, un aumento en el nivel de satisfacción del estudiantado respecto a los horarios asignados.

La comparación con el *solver* exacto CBC, formulado mediante Programación Lineal Entera, permitió validar la calidad de las soluciones obtenidas por la metaheurística y establecer un punto de referencia robusto. Si bien el *solver* exacto garantiza soluciones óptimas bajo el modelo propuesto, los resultados mostraron que su desempeño computacional se ve afectado de manera considerable. En contraste, SA presentó tiempos de ejecución estables y acotados, manteniendo una buena calidad de solución incluso en escenarios con alta densidad de restricciones.

El análisis detallado del caso de estudio INF295 permitió profundizar en el impacto del algoritmo a nivel individual, evidenciando una distribución más equilibrada de las penalizaciones entre los estudiantes y altos niveles de *fitness* individual. Estos resultados indican que los algoritmos propuestos no solo generan soluciones de alta calidad a nivel global, sino que también promueven una asignación de ayudantías más justa, evitando concentrar penalizaciones en un subconjunto reducido de estudiantes. Asimismo, la preferencia por bloques horarios satisfactorios para los estudiantes refuerza el enfoque centrado en el usuario del modelo propuesto, lo que resulta especialmente relevante en contextos académicos donde la percepción de equidad es un factor clave.

La diferencia en el costo computacional que se observó, resulta esperable, dado que el *solver* explora de forma exhaustiva el espacio de soluciones que satisfacen las restricciones duras, mientras que el algoritmo SA, mediante movimientos aleatorios, genera y recorre únicamente un vecindario acotado del espacio de búsqueda. No obstante, aun siendo un comportamiento esperado, la magnitud de la diferencia es significativa. En particular, para un escenario que considera solo 13 asignaturas y a los estudiantes de la carrera de Ingeniería Civil Informática de

sede Casa Central de la UTFSM, el *solver* presenta un tiempo de ejecución superior a tres veces el requerido por el algoritmo SA. En consecuencia, la utilización de este enfoque implicaría un aumento considerable del costo computacional, sin que la calidad de las soluciones obtenidas se vean lo suficientemente mejoradas como para justificar dicho sobrecosto.

Uno de los principales aportes de esta investigación corresponde al desarrollo de un modelo flexible para el algoritmo SA, ya que este algoritmo tiene una función de evaluación que puede adaptarse fácilmente a distintos criterios de satisfacción sin requerir una reformulación estructural del problema. Esta característica resulta especialmente relevante en contextos académicos reales, donde las preferencias estudiantiles, las políticas institucionales y las condiciones operativas pueden variar a lo largo del tiempo.

En conjunto, los resultados permiten concluir que el algoritmo SA constituye una alternativa eficiente, flexible y escalable para el problema de asignación de ayudantías, especialmente adecuada para contextos reales donde se requiere una rápida generación de soluciones de alta calidad y una fácil adaptación a cambios en las restricciones del problema.

Finalmente, si bien el *solver* exacto utilizado entrega soluciones de muy alta calidad y resulta fundamental como mecanismo de validación y referencia, su aplicabilidad práctica a nivel institucional se ve limitada por su mayor costo computacional y por la rigidez inherente a la reformulación del modelo ante cambios en las condiciones del problema. En este sentido, el algoritmo SA presenta una ventaja decisiva, ya que permite modificar de manera sencilla la función *fitness* para incorporar nuevas preferencias, políticas académicas o criterios de calidad, sin afectar la estructura general del algoritmo. Sumado a sus bajos tiempos de ejecución y a la obtención consistente de soluciones aceptables, estas características convierten a SA en una herramienta particularmente adecuada para su implementación a escala institucional, donde la flexibilidad, la rapidez de respuesta y la adaptabilidad a escenarios dinámicos resultan factores claves.

Desde una perspectiva metodológica, el desarrollo de este trabajo permitió evidenciar las ventajas prácticas del uso de metaheurísticas en una variante de planificación académica, particularmente cuando se enfrentan escenarios con restricciones y criterios de calidad difíciles de formalizar de manera estricta. A diferencia de los enfoques exactos, que requieren una formulación matemática precisa y cerrada, el enfoque basado en SA facilita la incorporación progresiva de conocimiento del dominio a través de la función *fitness*.

Esta característica resulta especialmente relevante en el contexto universitario, donde muchas restricciones importantes no son binarias ni absolutas, sino que responde a preferencias, hábitos o condiciones subjetivas del estudiantado. La posibilidad de modelar estos aspectos mediante penalizaciones suaves permite capturar de mejor forma la complejidad real del problema, evitando soluciones rígidas que, si bien son óptimas desde el punto de vista matemático, pueden resultar pocos satisfactorias en la práctica.

En relación con los objetivos planteados al inicio de esta memoria, los resultados obtenidos permiten concluir que estos fueron cumplidos de manera satisfactoria. Se logró modelar formalmente el problema de asignación de ayudantías, implementar y evaluar enfoques

basados en metaheurísticas y *solvers* exactos, y analizar su desempeño utilizando datos reales, validando la aplicabilidad del enfoque propuesto en un contexto académico completo.

## 5.2 TRABAJO FUTURO

Si bien el algoritmo propuesto demuestra un desempeño satisfactorio en los escenarios analizados, los resultados obtenidos evidencian tanto su potencial de escalabilidad como las limitaciones inherentes al alcance definido para este trabajo. En este sentido, la presente investigación abre diversas líneas de trabajo futuro orientadas a ampliar la cobertura del modelo, fortalecer su robustez y aumentar su aplicabilidad en contextos académicos reales.

Las extensiones propuestas surgen de manera natural a partir de los experimentos realizados y de las problemáticas observadas durante la implementación y validación del algoritmo. En particular, estas líneas de desarrollo apuntan a incorporar una mayor diversidad de actividades académicas, a considerar interacciones entre múltiples asignaturas y a integrar fuentes de información más representativas de la disponibilidad efectiva del estudiantado.

A continuación, se describen algunas direcciones de desarrollo que permitirían evolucionar el sistema hacia una herramienta de planificación académica más integral, capaz de abordar el problema desde una perspectiva institucional y centrada en el usuario.

### 5.2.1 SISTEMA MULTIPROPÓSITO

La versión actual del algoritmo se centra en la asignación óptima de ayudantías, maximizando la asistencia de los estudiantes y respetando sus preferencias horarias. Sin embargo, en un contexto académico real, una asignatura contempla múltiples actividades adicionales, tales como laboratorios, evaluaciones, talleres, controles y asignación de salas, las cuales también compiten por recursos temporales y espaciales.

Como trabajo futuro, se propone la extensión del algoritmo hacia un sistema multipropósito, capaz de integrar y coordinar distintos tipos de actividades académicas bajo un mismo modelo de optimización. Esta ampliación implica la incorporación de nuevas restricciones blandas y duras en la función *fitness*, así como la evaluación de relaciones temporales entre actividades. Por ejemplo, surge la interrogante de si resulta preferible programar talleres previos a los controles con fines de refuerzo, o bien separarlos temporalmente para evitar efectos negativos en la asistencia.

Para abordar la asignación de salas mediante el algoritmo *Simulated Annealing*, se requiere extender la representación de la solución, incorporando explícitamente el recurso espacial como una dimensión adicional del problema. En particular, el dominio asociado a los horarios de los estudiantes, actualmente modelado mediante la matriz  $E_{ijk} \in \{0, 1, 2\}$ , debería generalizarse para reflejar la coexistencia de múltiples propósitos y la asignación de salas específicas a cada actividad. Esto permitiría que los movimientos del algoritmo consideren no solo cambios temporales, sino también reasignaciones espaciales dentro de un vecindario definido.

Una vez definida esta representación extendida, sería posible incorporar nuevas penalizaciones asociadas a la capacidad de las salas, conflictos de uso, adecuación del tipo de sala a la actividad y preferencias institucionales, ajustando los pesos correspondientes en la función fitness.

Esta evolución del algoritmo permitiría abordar la planificación de una asignatura de manera más holística, considerando simultáneamente múltiples objetivos y restricciones, tales como la disponibilidad de salas, la carga académica del estudiante y la coordinación entre actividades teóricas y prácticas.

### **5.2.2 EVOLUCIÓN HACIA UN MODELO MULTI-ASIGNATURA**

Actualmente, el modelo resuelve el problema de asignación de ayudantías de forma local, considerando una única asignatura de manera aislada. Sin embargo, desde una perspectiva institucional, los principales conflictos de horario emergen a partir de la interacción entre múltiples asignaturas que comparten un mismo conjunto de estudiantes, lo que limita el impacto práctico del enfoque actual.

Como línea de trabajo futuro, se propone la evolución del algoritmo hacia un modelo multi-asignatura, capaz de coordinar simultáneamente las ayudantías de un departamento completo o incluso de varias unidades académicas. Para ello, se plantea la incorporación de una matriz de adyacencia curricular, que permita modelar las relaciones entre asignaturas en función de la superposición de estudiantes inscritos, identificando así dependencias y potenciales focos de conflicto.

No obstante, esta extensión introduce desafíos significativos tanto a nivel computacional como de implementación. En particular, el tamaño del espacio de búsqueda crecería de manera considerable, aumentando la complejidad asociada al diseño de la representación de soluciones, la definición de vecindarios eficientes y el ajuste de la función *fitness*. Asimismo, la incorporación de múltiples asignaturas obliga a gestionar prioridades institucionales, posibles conflictos entre criterios locales y globales, y a establecer un balance adecuado entre equidad individual y eficiencia sistémica.

Desde el punto de vista del desarrollo, este enfoque requiere un diseño modular y altamente escalable del algoritmo, que permita incorporar nuevas asignaturas y restricciones sin comprometer la mantenibilidad del sistema. En este sentido, la transición hacia un modelo multi-asignatura representa no solo un desafío algorítmico, sino también un problema de ingeniería de *software*, cuya resolución resulta clave para una eventual adopción del modelo en contextos reales. Este enfoque permitiría minimizar conflictos de manera global, considerando la carga académica completa de cada estudiante y no únicamente el impacto sobre una asignatura aislada, ampliando así el alcance y la aplicabilidad práctica de la solución propuesta.

En consecuencia, si bien las extensiones propuestas hacia un modelo multipropósito y multiasignatura permitirían aumentar significativamente el alcance y la aplicabilidad práctica del algoritmo, su implementación excede los objetivos y el alcance de este trabajo. Estas líneas de trabajo futuro implican un aumento sustancial en la complejidad del modelo, tanto a nivel algorítmico como de ingeniería de *software*, requiriendo un diseño escalable, modular y

cuidadosamente validado. No obstante, los resultados obtenidos en los casos analizados demuestran que el enfoque propuesto constituye una base sólida sobre la cual es posible construir soluciones más generales, capaces de abordar problemáticas de planificación académica de mayor escala.

### **5.2.3 SISTEMA DE INFORMACIÓN HÍBRIDO**

Un hallazgo relevante de esta investigación fue la discrepancia existente entre la información oficial registrada por la institución académica y la disponibilidad real de los estudiantes. Mientras que los datos institucionales reflejan los horarios formales de clases, la recolección directa de información en la asignatura INF285 evidenció la existencia de compromisos personales y actividades extraprogramáticas que inciden de manera significativa en la capacidad real de asistencia del estudiantado.

En este contexto, se propone como línea de trabajo futuro el desarrollo de un sistema de información híbrido, en el cual el algoritmo de optimización se alimente automáticamente de la base de datos institucional, pero que, adicionalmente, permita a los estudiantes editar y personalizar su disponibilidad a través de una interfaz. Esta integración permitiría incorporar información dinámica y contextual que actualmente no es considerada en los sistemas oficiales de planificación académica.

La implementación de este enfoque implica desafíos relevantes en términos de diseño de sistemas, validación de datos y resguardo de la consistencia de la información. No obstante, su incorporación permitiría generar horarios que representen de manera más fiel la realidad del estudiantado, aumentando la aceptación de las soluciones propuestas y fortaleciendo la percepción de equidad del proceso de asignación. Desde la perspectiva del estudiante, este tipo de sistema otorgaría un rol más activo en la planificación académica, transformando el proceso de asignación de ayudantías en una herramienta de apoyo real a la trayectoria formativa.

En general, esta línea de trabajo futuro refuerza la orientación centrada en el usuario del modelo propuesto, y plantea una evolución natural del algoritmo hacia soluciones que no solo optimizan métricas formales, sino que también consideran las condiciones reales en las que los estudiantes desarrollan su vida académica.

Las líneas de trabajo futuro propuestas en esta subsección reflejan el potencial de extensión y adaptación del modelo desarrollado, así como los desafíos inherentes a su aplicación en contextos académicos reales. Estas extensiones no solo suponen un aumento en la complejidad algorítmica, sino que también requieren una visión sistémica que integre criterios técnicos, institucionales y humanos. En este sentido, el presente trabajo establece una base conceptual y metodológica sólida, sobre la cual es posible construir soluciones de mayor alcance, manteniendo siempre como eje central la mejora de la experiencia académica del estudiantado.

### **5.3 REFLEXIÓN FINAL**

El desarrollo de esta memoria representó un proceso de aprendizaje que trascendió la implementación de un algoritmo de optimización, permitiendo comprender en profundidad la

complejidad real asociada a los problemas de planificación académica. A lo largo del trabajo, fue posible constatar que, si bien estos problemas pueden formularse desde un punto de vista matemático, su impacto se manifiesta directamente en la experiencia cotidiana de los estudiantes y en la dinámica institucional.

Uno de los principales desafíos enfrentados durante esta investigación fue el equilibrio entre la rigurosidad técnica del modelo y la necesidad de representar de manera adecuada las preferencias, condiciones y restricciones del entorno académico real. En este sentido, el uso de metaheurísticas permitió adoptar una aproximación flexible, capaz de incorporar criterios de calidad difíciles de formalizar de forma estricta, y de adaptarse progresivamente a las particularidades del problema abordado.

Asimismo, el trabajo evidenció la importancia de considerar el rol del estudiante como un actor central dentro de los procesos de asignación académica. La observación directa de los casos de estudio permitió comprender que soluciones matemáticamente óptimas no siempre se traducen en soluciones percibidas como justas o satisfactorias, reforzando la necesidad de enfoques centrados en el usuario y sensibles a la diversidad de realidades del estudiantado.

Desde una perspectiva personal y profesional, esta investigación consolidó competencias tanto en el ámbito de la optimización y el diseño algorítmico, como en el análisis crítico de sistemas complejos con múltiples actores y objetivos en conflicto. La experiencia adquirida durante el desarrollo de este trabajo sienta una base sólida para abordar futuros desafíos en los que se requiera combinar herramientas computacionales con una comprensión profunda del contexto en el que serán aplicadas.

Finalmente, se espera que esta memoria contribuya modestamente a la discusión sobre el uso de técnicas de optimización en la planificación académica, y que sirva como punto de partida para el desarrollo de soluciones que, además de ser eficientes desde el punto de vista computacional, aporten de manera concreta a la mejora de la experiencia formativa de los estudiantes.

## BIBLIOGRAFÍA

- [1] Universidad Técnica Federico Santa María, “Reglamento de Ayudantías,” Departamento de Matemática, pp. 1, 2005. [online]. Disponible en: [https://matematica.usm.cl/wp-content/uploads/2020/04/DR016\\_2005\\_Reglamento-Ayudantias.pdf](https://matematica.usm.cl/wp-content/uploads/2020/04/DR016_2005_Reglamento-Ayudantias.pdf).
- [2] M. Granada Echeverri, J. J. Santa Chávez, “Optimización Combinatoria -de la teoría a la práctica-”, Universidad Libre, Colombia, 2017. [online]. Disponible en: [https://www.researchgate.net/publication/331196434\\_Optimizacion\\_combinatoria\\_-\\_de\\_la\\_teor%C3%ADa\\_a\\_la\\_practica](https://www.researchgate.net/publication/331196434_Optimizacion_combinatoria_-_de_la_teor%C3%ADa_a_la_practica)
- [3] S. C. Brailsford, C.N. Potts, B. M. Smith, “Constraint satisfaction problems: Algorithms and applications”, *European Journal of Operational Research*, vol. 119, pp. 557-581, Diciembre, 1999. [online]. Disponible en: <https://cepac.cheme.cmu.edu/pasilectures/henning/EJOR-BrailsfordSmith.pdf>
- [4] Sugiarto, Mohammad Idhom, Roggo Alit, Akhmad Fauzi, “Implementation of Constraint Satisfaction Problem Methods on Course Scheduling in High School”, *Atlantis Highlights in Engineering (AHE)*, vol. 1, 2018. [online]. Disponible en: [https://www.researchgate.net/publication/330963318\\_Implementation\\_of\\_Constraint\\_Satisfaction\\_Problem\\_Methods\\_on\\_Course\\_Scheduling\\_in\\_High\\_School/fulltext/5c5d833f92851c48a9c1ffff/Implementation-of-Constraint-Satisfaction-Problem-Methods-on-Course-Scheduling-in-High-School.pdf](https://www.researchgate.net/publication/330963318_Implementation_of_Constraint_Satisfaction_Problem_Methods_on_Course_Scheduling_in_High_School/fulltext/5c5d833f92851c48a9c1ffff/Implementation-of-Constraint-Satisfaction-Problem-Methods-on-Course-Scheduling-in-High-School.pdf)
- [5] Mehdi M. “Parallel Hybrid Optimization Methods For Permutation Based Problems”, *Distributed, Parallel, and Cluster Computing [cs.DC]*. Université des Sciences et Technologie de Lille, Octubre 2011. [online]. Disponible en: [https://www.researchgate.net/publication/281015427\\_PARALLEL\\_HYBRID\\_OPTIMIZATION\\_METHODS\\_FOR\\_PERMUTATION\\_BASED\\_PROBLEMS](https://www.researchgate.net/publication/281015427_PARALLEL_HYBRID_OPTIMIZATION_METHODS_FOR_PERMUTATION_BASED_PROBLEMS)
- [6] B. Melián Batista, F. Glover, “Introducción a la Búsqueda Tabú”, *Optimización Heurística y Redes Neuronales*, pp. 105-142, 1996. [online]. Disponible en: [https://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS\\_Spanish%20w%20Belen%2811-9-06%29.pdf](https://leeds-faculty.colorado.edu/glover/fred%20pubs/329%20-%20Introduccion%20a%20la%20Busqueda%20Tabu%20TS_Spanish%20w%20Belen%2811-9-06%29.pdf)
- [7] V. Granville, M. Krivanek, J. Rasson, “Simulated annealing: A proof of convergence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 11, pp. 652–656, 30 Junio 1994. [online]. Disponible en: <https://ieeexplore.ieee.org/document/295910>
- [8] J. P. Pascual, I. Chiazzo, "Asignación de horarios utilizando algoritmos evolutivos" 2011. [online]. Disponible en: [https://www.academia.edu/16090250/Algoritmos\\_evolutivos\\_Asignaci%C3%B3n\\_de\\_horarios\\_Time\\_Tabling\\_problem\\_](https://www.academia.edu/16090250/Algoritmos_evolutivos_Asignaci%C3%B3n_de_horarios_Time_Tabling_problem_)

- [9] UniTime.org, "University Timetabling, Comprehensive Academic Scheduling Solutions", Febrero, 2025. [online]. Disponible en: <https://www.unitime.org/>
- [10] T. Müller, R. Barták, H. Rudova, "Iterative Forward Search Algorithm: Combining Local Search Maintaining Arc Consistency and a Conflict-Based Statistics", Septiembre, 2004. [online]. Disponible en: [https://www.researchgate.net/publication/221633250\\_Iterative\\_Forward\\_Search\\_Algorithm\\_Combining\\_Local\\_Search\\_with\\_Maintaining\\_Arc\\_Consistency\\_and\\_a\\_Conflict-Based\\_Statistics](https://www.researchgate.net/publication/221633250_Iterative_Forward_Search_Algorithm_Combining_Local_Search_with_Maintaining_Arc_Consistency_and_a_Conflict-Based_Statistics)
- [11] C. C.Gotlieb, "The Construction of Class-Teacher Time-Tables." *IFIP Congress*, 1962, pp. 73-77.
- [12] A. Schaerf, "A Survey of Automated Timetabling", Dipartimento di Informatica e Sistemistica Università di Roma "La Sapienza", pp. 2-43, Enero, 1996. [online]. Disponible en: ["https://www.researchgate.net/publication/2371435\\_A\\_Survey\\_of\\_Automated\\_Timetabling"](https://www.researchgate.net/publication/2371435_A_Survey_of_Automated_Timetabling).
- [13] M. W. S. Almeida, J. P. S. Medeiros and P. R. Oliveira, "Solving the Academic Timetable Problem Thinking on Student Needs," *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 673-676, Diciembre, 2015. [online]. Disponible en: ["https://ieeexplore.ieee.org/abstract/document/7424396"](https://ieeexplore.ieee.org/abstract/document/7424396).
- [14] K. Sylejmani, E. Gashi, A. Ymeri, "Simulated annealing with penalization for university course timetabling", *Journal of Scheduling*, pp. 497–517, Julio 2022. [online]. Disponible en: [https://www.researchgate.net/publication/362137880\\_Simulated\\_annealing\\_with\\_penalization\\_for\\_university\\_course\\_timetabling](https://www.researchgate.net/publication/362137880_Simulated_annealing_with_penalization_for_university_course_timetabling)
- [15] R. A. Rutenbar, "Simulated annealing algorithms: an overview," *IEEE Circuits and Devices Magazine*, pp. 19-26, Enero, 1989. [online] Disponible en: <https://ieeexplore.ieee.org/abstract/document/17235>
- [16] D. Van Bulck and D. Goossens, "The international timetabling competition on sports timetabling (ITC2021)," *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, vol. 308, no. 3, pp. 1249–1267, 2023. [online] Disponible en: <https://biblio.ugent.be/publication/01GK63HTH1EY74KDYTVJVBDR30>
- [17] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf y T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," *Comput. Oper. Res.*, vol. 65, pp. 83–92, Enero, 2016. [online] Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S0305054815001690>.
- [18] C. Arancibia, "*Repositorio del proyecto: Un algoritmo para la planificación de horarios de ayudantía universitaria*", GitHub, 2025. [online] Disponible en: <https://github.com/Camilu-png/student-assistant-scheduling>
- [19] Ober I., "A variant of the high-school timetabling problem and a software solution for it based on integer linear programming", *Proceedings of the 11th International Conference on*

Practice and Theory of Automated Timetabling (PATAT-2016), pp. 283-294, Agosto 2016.  
[online]. Disponible en: <https://www.irit.fr/~Julian.Ober/docs/paper26-PATAT.pdf>