



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Asignación de estudiantes a cursos con turnos considerando preferencias y envidia

Informe de Proyecto de Grado presentado por

Rodrigo Boffa y Mateo Romero

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería de Producción de Facultad de Ingeniería de la Universidad de la
República

Supervisores

Carlos Testuri, Pedro Piñeyro y Libertad Tansini

Montevideo, 6 de marzo de 2025



Asignación de estudiantes a cursos con turnos considerando preferencias y envidia por Rodrigo Boffa y Mateo Romero tiene licencia [CC Atribución 4.0](https://creativecommons.org/licenses/by/4.0/).

Agradecimientos

En primer lugar, agradecemos a nuestros tutores, Carlos Testuri, Pedro Piñeyro y Libertad Tansini, por su acompañamiento y orientación durante el desarrollo del proyecto. Su experiencia y compromiso han sido determinantes en la realización de este trabajo. Asimismo, reconocemos la implicación y la dedicación que aportaron en cada etapa del proyecto.

En segundo lugar, queremos agradecer a nuestros compañeros de estudio de la carrera, especialmente a Mateo Parpal, Franco Giosa, y Federico González, por acompañarnos en este trayecto y hacer de ello una experiencia memorable y gratificante.

Por último, expresamos nuestro agradecimiento a nuestras familias por el apoyo incondicional recibido no solo durante esta etapa, sino que también a lo largo de toda la carrera.

Resumen

Este informe presenta el estudio realizado sobre el problema de asignación de estudiantes a cursos con cierta cantidad de turnos y cupos, considerando las preferencias de los estudiantes y las superposiciones de turnos. Este problema se enmarca en el contexto de la Universidad de la República, donde es común que algunos cursos se encuentren sobrecargados, mientras que otros quedan con cupos disponibles. Como parte del trabajo, se realizó una revisión de literatura sobre modelos cuantitativos en la gestión de la enseñanza, centrándose en la asignación de estudiantes a distintas actividades académicas como cursos, seminarios y talleres. En particular, se exploró la relación entre equidad y eficiencia, y se analizaron mecanismos que promueven la equidad en las asignaciones. Se utiliza el enfoque de equidad basado en el concepto de envidia, entendida como la percepción que tiene un estudiante cuando otro estudiante recibe una asignación más favorable. Para abordar el problema, se proponen dos modelos, denominados “Fase 1” y “Fase 2”. En el modelo de la Fase 1, se maximiza la satisfacción de los estudiantes considerando sus preferencias. En el modelo de la Fase 2, se minimiza la envidia, asegurando que la satisfacción alcanzada en el modelo de la Fase 1 no se vea comprometida. Para evaluar el comportamiento de los modelos en distintos escenarios, se generaron configuraciones de datos con variaciones en los valores de los parámetros de entrada, para analizar el impacto de estos cambios en las soluciones obtenidas. Los resultados se analizaron mediante indicadores que miden la calidad de las asignaciones generadas, permitiendo la comparación entre configuraciones. A partir de los resultados obtenidos, se concluye que los modelos propuestos solucionan el problema abordado, y que, si bien la resolución del modelo de la Fase 2 es computacionalmente más compleja que la del modelo de la Fase 1, en ciertos casos logra obtener soluciones más equitativas, manteniendo la satisfacción global.

Palabras clave: Asignación de Estudiantes a Cursos, Asignación con Preferencias, Equidad, Envidia, Optimización, Programación Entera

Índice general

1. Introducción	1
2. Marco Teórico	3
2.1. Problemas de asignación con preferencias	3
2.2. Equidad y envidia en asignaciones	5
2.3. Otros acercamientos a problemas de asignación	7
3. Definición y modelado del problema	9
3.1. Definición del problema	9
3.2. Formulación de los modelos matemáticos	11
3.3. Validación de los modelos matemáticos	15
4. Experimentación numérica	25
4.1. Generación de datos	26
4.1.1. Elementos variables	26
4.1.2. Generación del caso base y configuraciones	28
4.1.3. Generador de datos	31
4.2. Indicadores	31
4.3. Resultados	33
4.3.1. Resultados del análisis computacional	34
4.3.2. Resultados del análisis de sensibilidad	35
5. Conclusiones y Trabajo Futuro	59
Referencias	63
A. Apéndice	65
A.1. Estado del arte	65
A.2. Modelo unificado	65
A.3. Cantidad máxima de superposiciones	67
A.4. Funcionamiento del generador de datos	67
A.5. Archivo para obtener los indicadores	72
A.6. Envidia por listas de espera	75

Capítulo 1

Introducción

La Universidad de la República (UdelaR), fundada en 1849, es la principal institución de educación superior en Uruguay, con más de 140.000 estudiantes. Su oferta académica incluye más de 90 carreras de grado y 300 programas de posgrado. Esta propuesta educativa se distribuye en 15 facultades, 3 institutos, 6 escuelas, 3 centros universitarios regionales y un hospital universitario. En varias de estas carreras, algunos de los turnos de los cursos están desiertos, mientras que la gran mayoría se encuentran sobrecargados. Esto se debe a factores como la limitada capacidad de las aulas, el presupuesto restringido, y las preferencias de los estudiantes por horarios o profesores específicos. Todas las inscripciones a los cursos, independientemente de la carrera, se gestionan a través del Sistema de Gestión Administrativa de la Enseñanza (SGAE). Este sistema centralizado también permite a los estudiantes consultar los calendarios de cursos y exámenes, solicitar escolaridad, entre otros servicios.

En muchas facultades, algunos cursos cuentan con cupos limitados; en estos casos se utilizan diferentes criterios para la asignación de quienes aspiran a realizar un curso con estas características. Uno de los más comunes es el criterio de “orden de llegada”, donde los primeros estudiantes en inscribirse, tras la apertura de inscripciones en una fecha y hora determinadas, obtienen los cupos disponibles. Esta situación genera un trato poco equitativo y, sobre todo, poco transparente. De acuerdo con lo informado por el servicio responsable del sistema, este mecanismo provoca una sobrecarga en el sistema, generando picos que en varias oportunidades terminan dejando a toda la UdelaR sin acceso a uno de sus sistemas neurálgicos. Por ello, las autoridades universitarias han reconocido la necesidad de implementar mecanismos de inscripción más equitativos y transparentes para los cursos con cupos limitados. Estos mecanismos deben basarse en criterios claros y predefinidos, y garantizar un acceso equitativo a todos los estudiantes, independientemente de su velocidad de conexión y de la hora del día en que se inscriben, evitando así una concurrencia excesiva en determinados horarios. En consecuencia, se determinó que las inscripciones a cursos con cupos limitados se deberán realizar utilizando herramientas y algoritmos específicos

para asignar los cupos de manera equitativa, pero que en ningún caso podrán depender del orden de llegada concurrente del estudiante que aspira a inscribirse.

En este trabajo se proponen dos modelos resueltos en fase, logrando así considerar tanto las preferencias individuales de los estudiantes como la equidad en la asignación. Para considerar las preferencias de un estudiante, este deberá asignarle valores de preferencia a los turnos de los cursos a los cuales desea asistir, en donde valores más bajos refieren a una mayor prioridad. Por otra parte, la equidad se mide a través de la envidia global de la asignación, basándose en que los estudiantes que quedan asignados a sus turnos no preferidos envidiarán a los que sí quedaron asignados a esos turnos. Una explicación más detallada del funcionamiento de las preferencias y la envidia se puede ver en 3.1. En la Fase 1, el objetivo es maximizar la satisfacción basada en las preferencias indicadas, minimizando el valor total de preferencias. En la Fase 2, se minimiza la envidia total entre estudiantes, asegurando que el nivel de satisfacción alcanzado en la Fase 1 se mantenga. Para analizar la eficacia de los modelos propuestos, se generaron conjuntos de datos de entrada con distintas configuraciones que simulan diversas circunstancias reales. Estas configuraciones permiten evaluar cómo se comportan los modelos frente a variaciones en factores clave como el número de estudiantes, la capacidad de los cursos y la cantidad de turnos disponibles. Además, se definieron una serie de indicadores que permiten medir y comparar la calidad de las asignaciones generadas en cada configuración. Los resultados obtenidos permitieron evaluar cómo cambian los niveles de satisfacción y envidia al modificar los valores de los parámetros de entrada. En ciertos casos, principalmente en los que existen superposiciones, se obtuvieron mejoras en los valores del nivel de envidia al aplicar la Fase 2, sin comprometer la satisfacción inicial.

El resto del documento se organiza de la siguiente manera: en el Capítulo 2 se presenta el marco teórico y los conceptos relacionados con la satisfacción de preferencias y la equidad en la asignación, así como el concepto de envidia utilizado en el modelo de la Fase 2. En el Capítulo 3 se detalla la formulación matemática de los modelos de asignación, incluyendo las restricciones, criterios de equidad considerados y la validación de los modelos. El Capítulo 4 está dedicado a la experimentación numérica, donde se presentan las distintas configuraciones de datos de entrada y los indicadores utilizados para evaluar el comportamiento de los modelos. Finalmente, el Capítulo 5 presenta las conclusiones y propuestas de trabajo futuro sobre la base de los resultados obtenidos en el proyecto.

Capítulo 2

Marco Teórico

En este capítulo se establecen los fundamentos teóricos que sustentan el desarrollo del presente trabajo. Se aborda el problema de la asignación de estudiantes a cursos como un problema clásico de asignación donde se consideran las preferencias de los estudiantes. Además, se presenta el concepto de equidad asociado a este problema y se describe el concepto de envidia en el marco de la asignación de estudiantes a cursos.

2.1. Problemas de asignación con preferencias

En general, los problemas de asignación buscan responder la pregunta de cómo asignar ciertos objetos (trabajadores, estudiantes, etc.) a otros objetos (máquinas, tareas, etc.). La estructura combinatoria subyacente no es más que una correspondencia entre dos conjuntos finitos. En el problema de optimización, se busca la mejor asignación, es decir, se debe optimizar alguna función objetivo que dependa de dicha asignación. En particular, el problema de asignación unilateral con preferencias, también conocido como emparejamiento uno a muchos, consiste en asignar objetos (como cursos) a agentes (como estudiantes) teniendo en cuenta sus preferencias individuales, pero sin la mediación de dinero. En el contexto educativo, los estudiantes expresan preferencias ordinales que deben ser consideradas, mientras que los cursos presentan restricciones como capacidades máximas y superposiciones de horarios. El objetivo principal es encontrar una asignación que maximice la satisfacción global de los estudiantes, respetando estas limitaciones.

Los problemas más simples de asignación en el ámbito educativo se centran únicamente en restricciones de capacidad. [Proll \(1972\)](#) propone un modelo de programación lineal para asignar estudiantes a proyectos. En este problema, los estudiantes seleccionan un subconjunto de proyectos disponibles y los ordenan según sus preferencias, mientras que se busca minimizar el máximo valor del orden de los proyectos para todos los estudiantes. De manera similar, [Willoughby](#)

y Zappe (2006) presentan un modelo para asignar estudiantes de primer año a seminarios universitarios. En este caso, los estudiantes generan listas ordenadas de preferencias, y el modelo minimiza el rango promedio de estas preferencias asignadas a los seminarios, tratando de satisfacer las preferencias más altas de los estudiantes. Las restricciones del problema aseguran que cada estudiante sea asignado a un único seminario, y que se cumplan las restricciones de capacidad de cada seminario.

Los problemas de asignación de estudiantes a cursos, en los que se deben tener en cuenta las preferencias individuales de los estudiantes, se han abordado utilizando varios modelos y técnicas. A modo de ejemplo, en el artículo de Beroš y Meter (2015) se presenta un modelo de programación entera para la asignación de estudiantes a cursos electivos. Este modelo busca maximizar la satisfacción total de los estudiantes en función de sus preferencias, respetando las restricciones de capacidad de los cursos y los recursos disponibles. Bichler y Merting (2021) abordan el problema de asignación como un problema de asignación combinatoria y proponen el uso de mecanismos aleatorios, que permiten asignar a los estudiantes a combinaciones de cursos de una manera probabilística. Nogareda y Camacho (2017) proponen dos enfoques diferentes para abordar el problema de asignación. El primer enfoque es un problema de satisfacción de restricciones, que se centra en encontrar una solución que cumpla con todas las restricciones establecidas, como la capacidad de los cursos y las preferencias de los estudiantes. El segundo enfoque utiliza optimización por colonia de hormigas.

Al abordar problemas de asignación, donde varios objetivos pueden entrar en conflicto, el enfoque multiobjetivo se convierte en una herramienta importante. Como señalan Gralla, Goentzel, y Fine (2014), los compromisos entre múltiples objetivos son inevitables al diseñar operaciones o asignar recursos escasos. En estas situaciones, comprender estos compromisos es fundamental para tomar buenas decisiones, especialmente cuando los objetivos están en conflicto o cuando hay incertidumbre sobre su importancia relativa. En este trabajo, se adopta un esquema jerárquico, donde primero se optimiza un objetivo principal, en este caso la satisfacción de los estudiantes, y luego se refina la solución para considerar la equidad.

2.2. Equidad y envidia en asignaciones

Al asignar cursos a estudiantes, uno de los objetivos principales puede ser maximizar su satisfacción considerando las preferencias individuales. Sin embargo, al considerar solo este objetivo, es posible obtener asignaciones donde algunos estudiantes se vean fuertemente perjudicados en relación con sus preferencias respecto a los demás estudiantes. Como señalan [Chen y Hooker \(2023\)](#), los modelos de optimización típicamente buscan maximizar el beneficio total o minimizar el costo total. Sin embargo, la equidad es un elemento importante en muchas decisiones prácticas y es mucho menos obvio cómo expresarla matemáticamente. La equidad puede ser entendida de muchas maneras; no hay una manera generalizada de representarla.

Existen diversas estrategias para incorporar la equidad en los problemas de asignación. Por ejemplo, [Beroš y Meter \(2015\)](#) introducen un nivel mínimo de satisfacción para cada estudiante, asegurando que todos reciban al menos una asignación que no sea completamente desfavorable. Este enfoque busca un equilibrio entre la equidad y la eficiencia, evitando que algunos estudiantes queden totalmente insatisfechos con su asignación. De forma similar, [van den Broek, Hurkens, y Woeginger \(2009\)](#) amplían este enfoque al considerar listas ordenadas de preferencias de los estudiantes y penalizar asignaciones a cursos con posiciones bajas en dichas listas. En el trabajo de [Chen y Hooker \(2023\)](#), se exploran varios criterios que pueden ser incluidos en los modelos de optimización para abordar la equidad. Entre estos criterios se encuentran:

- **Maxmin:** Este criterio se enfoca en maximizar la satisfacción del estudiante que se encuentra en la peor situación. Aplicado a la asignación de cursos, garantiza que el estudiante menos privilegiado en términos de preferencias reciba la mejor asignación posible dentro de sus limitaciones.
- **Maximización lexicográfica:** Este enfoque extiende el criterio *maxmin* al mejorar iterativamente la satisfacción de los estudiantes en orden de los menos privilegiados a los más privilegiados. Busca mejorar la equidad de manera jerárquica, priorizando primero a los estudiantes que están en las peores condiciones.
- **Combinaciones convexas:** Este criterio combina diferentes objetivos de equidad y eficiencia en un solo modelo, permitiendo un ajuste para equilibrar estos objetivos según el contexto específico de la asignación.

En los problemas de asignación, existen diversas maneras de incorporar el concepto de equidad, algunas de las cuales fueron mencionadas anteriormente. En este trabajo, se decide enfocar la equidad desde el concepto de envidia. En este contexto, la envidia surge cuando un estudiante prefiere la asignación que ha recibido otro estudiante sobre la suya propia, lo que genera una percepción de injusticia en el proceso de asignación. En situaciones donde no se involucra el dinero y todos los recursos deben ser asignados, no siempre es posible garantizar

una asignación completamente libre de envidia (Netzer, Meisels, y Zivan, 2016). Ante esta situación, en lugar de eliminar completamente la envidia, el objetivo es minimizarla, buscando una asignación que sea percibida como más justa y equitativa por todos los estudiantes.

Existen varios criterios para reducir la cantidad de envidia en una asignación. Una opción es minimizar el número de estudiantes envidiosos, es decir, aquellos que prefieren la asignación de otro. Otra alternativa es reducir la suma total de envidia en la asignación, lo que se conoce como la minimización de la envidia utilitaria. Además, es posible optar por minimizar la envidia del estudiante más envidioso, asegurando que ningún estudiante sufra un alto grado de insatisfacción en comparación con los demás. Estos enfoques ofrecen diferentes maneras de gestionar la equidad en sistemas donde los recursos son limitados y es importante considerar las preferencias de los estudiantes. En la asignación de estudiantes a cursos, puede ser necesario equilibrar la satisfacción individual con la equidad. Aunque maximizar el bienestar puede ser eficiente, esto no siempre asegura una distribución justa. La consideración de la equidad permite que el proceso de asignación sea percibido como más justo por todos los estudiantes, reduciendo las desigualdades y mejorando la percepción de justicia.

2.3. Otros acercamientos a problemas de asignación

Además de los temas abordados en las secciones precedentes, que incluyen preferencias y equidad en los problemas de asignación de estudiantes a cursos, existen otros enfoques y problemas tratados en la literatura académica. Estos incluyen restricciones específicas relacionadas con capacidades, horarios y características de los estudiantes, así como criterios de estabilidad en las asignaciones. La siguiente selección de artículos es un resumen de un análisis más detallado del estado del arte, el cual se encuentra desarrollado en el Apéndice [A.1](#).

Un enfoque fundamental en los problemas de asignación educativa son las restricciones de capacidad. Este tipo de problemas se centran en garantizar asignaciones factibles, respetando límites físicos o administrativos. Por ejemplo, [Manlove, Milne, y Olaosebikan \(2018\)](#) estudian la asignación de estudiantes a proyectos con preferencias tanto de estudiantes como de profesores. En su modelo, se deben respetar las capacidades máximas de los proyectos y de supervisión de los profesores, buscando emparejamientos estables que consideren las prioridades de ambas partes.

En los problemas de asignación que consideran la existencia de múltiples instancias para una misma actividad, se plantean soluciones que abordan las complejidades específicas de estos escenarios. [Agustín-Blas, Salcedo-Sanz, Ortiz-García, Pérez-Bellido, y Portilla-Figueras \(2008\)](#) presentan la asignación de estudiantes a grupos de laboratorios, respetando tanto las preferencias de los estudiantes como las de los profesores, y minimizando la insatisfacción. El modelo propuesto también incluye criterios para balancear la carga entre los grupos y cumplir con las restricciones de capacidad. De forma similar, [Beroš y Meter \(2015\)](#) introducen un modelo de programación entera que considera restricciones como la necesidad de garantizar una satisfacción mínima por estudiante y un balance adecuado entre los cursos. Este enfoque busca evitar asignaciones desfavorables para algunos estudiantes, distribuyendo las preferencias de manera equitativa. Por su parte, [van den Broek y cols. \(2009\)](#) diseñan un modelo que limita la asignación de estudiantes a posiciones menos favorables en sus listas de preferencia, incorporando restricciones adicionales para atender preasignaciones y garantizar que los cursos prioritarios sean cubiertos.

Otro enfoque posible es cuando los problemas de asignación incluyen la planificación de horarios como una restricción ya definida. [Domenech y Lusa \(2016\)](#) proponen un modelo que balancea la carga de trabajo de los profesores y maximiza sus preferencias. Este enfoque considera horarios preestablecidos y utiliza un modelo de programación lineal mixta. Por otro lado, [Heitmann y Brüggemann \(2014\)](#) abordan la asignación de estudiantes a cursos en múltiples instancias, asegurando que no haya conflictos entre los horarios y que se optimicen las preferencias individuales de los estudiantes. Entre los problemas de asignación

donde los horarios se definen con la resolución del problema, se puede nombrar el trabajo de [Varone y Schindl \(2013\)](#), donde se desarrolla un modelo de programación que combina la asignación de estudiantes a cursos y la programación de los horarios de dichos cursos. Este enfoque incluye restricciones de capacidad mínima y máxima, priorización de cursos principales frente a secundarios y restricciones de simultaneidad de horarios.

Dentro de los problemas de asignación que consideran características específicas de los estudiantes, se puede señalar el artículo de [Schneider, Hirtreiter, y Morgenstern \(2009\)](#), donde se trata la asignación de estudiantes a universidades teniendo en cuenta su progreso académico y sus preferencias sobre las instituciones disponibles. En este caso se busca maximizar la satisfacción de los estudiantes, priorizando la asignación de los de mayor rendimiento académico. Por su parte, [Chartier, Ellison, y Langville \(2014\)](#) introducen un modelo que integra la diversidad como un factor clave en la conformación de grupos, considerando género y etnia como elementos centrales para garantizar una asignación balanceada y equitativa en términos de diversidad.

En cuanto a los problemas con criterios de estabilidad, [Chiarandini, Fagerberg, y Gualandi \(2019\)](#) exploran asignaciones de estudiantes a proyectos bajo restricciones de capacidad, maximizando la satisfacción colectiva y reduciendo la envidia de los participantes. También se introducen varias métricas de estabilidad, como la ausencia de envidia y la optimalidad de Pareto, para evaluar la calidad de las asignaciones. [Darmann, Döcker, Dorn, y Schneckenburger \(2022\)](#) analizan la complejidad computacional de asignaciones grupales libres de envidia y estables, proponiendo modelos que equilibran las preferencias individuales con restricciones de capacidad.

Capítulo 3

Definición y modelado del problema

En este capítulo se presentan los modelos de asignación de estudiantes a cursos elaborados, considerando las restricciones y limitaciones existentes. El problema de asignación se resuelve en dos fases. En la Fase 1, se maximiza la satisfacción de los estudiantes, sujeta a las restricciones establecidas. En la Fase 2, se minimiza la envidia entre los estudiantes, permitiendo ajustes en las asignaciones siempre que no se modifique el nivel de satisfacción alcanzado en la fase anterior. Este enfoque busca minimizar la percepción de injusticia en el proceso de asignación.

3.1. Definición del problema

El problema abordado surge ante la necesidad de asignar un conjunto de estudiantes a turnos de cursos, respetando múltiples restricciones. Entre ellas destacan los límites de capacidad en cada turno y la existencia de superposiciones horarias que impiden a un mismo estudiante asistir a dos turnos que coincidan en el tiempo. Además, se desea que las preferencias de los estudiantes sean consideradas al momento de realizar la asignación. Para resolver este problema, se desarrollaron dos modelos matemáticos, que se resuelven en forma secuencial. En la Fase 1 se maximiza la satisfacción global, mientras que en la Fase 2 se reduce la envidia (que se manifiesta cuando un estudiante preferiría la asignación de otro frente a la suya), manteniendo el nivel de satisfacción obtenido en la primera fase. Los elementos principales del problema se describen a continuación. En primer lugar, se cuenta con un conjunto de estudiantes que expresan sus preferencias ordinales sobre los turnos de los cursos disponibles. Dichas preferencias se expresan en valores numéricos, donde un valor de i indica mayor preferencia que un valor de j , si $i < j$. Por otra parte, se dispone de un conjunto de cursos, los cuales pueden ofrecer uno o más turnos con capacidades limitadas. Estas capacidades determinan el número máximo de estudiantes que

se puede asignar a cada turno específico. Adicionalmente, cada curso contempla una lista de espera en la cual se asignan estudiantes que no pueden ser ubicados en ninguno de sus turnos preferidos. De esta manera, si dentro de un curso quedan muchos estudiantes asignados a su lista de espera, se podría evaluar aumentar la capacidad de alguno de sus turnos, o agregar más turnos, de modo que se pueda asignar a dichos estudiantes. Además, si estudiantes asignados a turnos de ese curso se dan de baja, se podrían asignar aquellos estudiantes que quedaron en su lista de espera.

En la Fase 1, el objetivo principal es maximizar la satisfacción de los estudiantes, lo que se traduce en minimizar la suma de las preferencias expresadas, referido como el valor de preferencias a lo largo del informe. El límite de capacidad de cada turno se impone como restricción dura, ya que no puede asignarse más de una cantidad específica de estudiantes a un turno determinado. Asimismo, se garantiza que cada estudiante no se asigne a más de un turno por curso, y que no incurra en superposiciones horarias. En caso de que no sea posible ubicar a un estudiante en ninguno de los turnos de un curso al cual desea inscribirse, el modelo contempla su asignación en la lista de espera para dicho curso. Una vez determinado el máximo nivel de satisfacción por la Fase 1, la Fase 2 se centra en minimizar la envidia total entre los estudiantes. El modelo de esta fase se enfoca en reducirla en la mayor medida posible, sin empeorar el nivel de satisfacción obtenido en la Fase 1. Por esta razón, el modelo de la Fase 2 incorpora una nueva restricción para garantizar este comportamiento, manteniendo, además, las de la Fase 1.

La envidia, en el contexto de este trabajo, se define de la siguiente manera: un estudiante i , al que se le asignó el turno k del curso c , envidia a otro estudiante j , asignado al turno l del mismo curso, si ambos expresan preferencias para dicho curso, con turnos $\{1, \dots, k, l, \dots, m\}$, el estudiante i expresa una preferencia p_{ick} al turno k y una preferencia p_{icl} al turno l , y se cumple que $p_{ick} \geq p_{icl}$. Esto indica que el estudiante i hubiera preferido ser asignado al turno l , que le fue otorgado al estudiante j . El valor de la envidia se calcula como la diferencia $p_{ick} - p_{icl}$. Si esta diferencia es positiva, el estudiante i envidia al estudiante j , ya que el turno l asignado a j tiene una mayor preferencia para i que su propio turno asignado k .

Esta definición de envidia se basa en la empleada por Chiarandini y cols. (2019), quienes aplican el concepto de envidia para la asignación de estudiantes a proyectos, de modo que cada grupo de estudiantes compare su proyecto asignado con el de otros. A diferencia de su enfoque, que también explora la posibilidad de asignaciones libres de envidia, este trabajo se centra en medir y minimizar dicha envidia sin imponer su completa eliminación, manteniendo el nivel óptimo de satisfacción.

El enfoque en dos fases (maximizar la satisfacción y, luego, minimizar la envidia sin sacrificar la satisfacción obtenida previamente) se justifica ante la necesidad de equilibrar satisfacción y equidad. Por un lado, es fundamental

que los estudiantes reciban turnos que reflejen sus preferencias. Por otro lado, si algunos perciben grandes desigualdades, por ejemplo, si quedan muy lejos de sus opciones deseadas mientras otros obtienen sus asignaciones preferidas, el proceso de asignación se considera poco equitativo. Al introducir la envidia como indicador de equidad, se incorpora un mecanismo que garantiza que la solución final sea no solo óptima en términos de satisfacción, sino también más justa.

3.2. Formulación de los modelos matemáticos

En esta sección se describen los modelos matemáticos desarrollados para abordar el problema de asignación de estudiantes a cursos. Los modelos integran en la Fase 1, las preferencias individuales de los estudiantes y, en la Fase 2, además, un criterio de equidad, con el objetivo de maximizar la satisfacción de los estudiantes y minimizar la envidia en el proceso de asignación. A continuación, se presenta la nomenclatura utilizada para los modelos.

Conjuntos

- E : conjunto de estudiantes.
- C : conjunto de cursos.
- C_e : conjunto de cursos para los cuales el estudiante $e \in E$ expresa preferencias.
- T_c : conjunto de turnos disponibles para cada curso $c \in C$.
- T : conjunto de todos los turnos disponibles. Se define como $T = \bigcup_{c \in C} T_c$
- S_{ct} : conjunto que representa los pares curso-turno que se superponen con el curso $c \in C$ en el turno $t \in T_c$, con $(c, t) \in S_{ct}$.

Parámetros

- p_{ect} : preferencia del estudiante $e \in E$ en el curso $c \in C$ por el turno $t \in T_c$.
- r_e : máximo valor de preferencia que el estudiante $e \in E$ marca por algún turno de algún curso, más 1. Se utiliza para penalizar las asignaciones a lista de espera. Se define formalmente como:

$$r_e = 1 + \max_{c \in C_e, t \in T_c : p_{ect} < u} p_{ect}$$

- q_{ct} : capacidad máxima en el curso $c \in C$ del turno $t \in T_c$.
- u : valor de preferencia asignado a los turnos de cursos que no son preferidos por los estudiantes.

- k : valor de preferencia considerado demasiado elevado. Las asignaciones a preferencias mayores o iguales a k se penalizarán.
- n : valor utilizado para penalizar las preferencias mayores o iguales a k .

Variables de decisión

- x_{ect} : variable binaria que toma el valor de 1 si el estudiante $e \in E$ es asignado al curso c en el turno $t \in T_c$, y 0 en caso contrario.
- w_{ec} : variable binaria que toma el valor 1 si el estudiante $e \in E$ queda en la lista de espera para el curso $c \in C$, y 0 en caso contrario.
- v_{efc} : representa el valor de envidia del estudiante $e \in E$ hacia el estudiante $f \in E$ por el curso $c \in C$, con $f \neq e$.

Maximización de la satisfacción global en la Fase 1

En el modelo propuesto para la Fase 1, se maximiza la satisfacción global de los estudiantes, considerando las restricciones descritas en la Sección 3.1.

$$\begin{aligned}
 \text{(Fase 1) : } \min & \sum_{e \in E} \sum_{c \in C_e} \sum_{t \in T_c: p_{ect} < k} p_{ect} x_{ect} + \sum_{e \in E} \sum_{c \in C_e} \sum_{t \in T_c: p_{ect} \geq k} n p_{ect} x_{ect} \\
 & + \sum_{e \in E} \sum_{c \in C_e} n r_e w_{ec}
 \end{aligned} \tag{3.1}$$

sujeto a :

$$\sum_{t \in T_c: p_{ect} < u} x_{ect} + w_{ec} = 1, \quad e \in E, c \in C_e \tag{3.2}$$

$$\sum_{e \in E} x_{ect} \leq q_{ct}, \quad c \in C, t \in T_c \tag{3.3}$$

$$\sum_{(c,t) \in S_{ij}: p_{ect} < u} x_{ect} \leq 1, \quad e \in E, i \in C_e, j \in T_i \tag{3.4}$$

$$\sum_{t \in T_c} x_{ect} + w_{ec} = 0, \quad e \in E, c \in C \setminus C_e \tag{3.5}$$

$$x_{ect} \in \{0, 1\}, \quad e \in E, c \in C_e, t \in T_c \tag{3.6}$$

$$w_{ec} \in \{0, 1\}, \quad e \in E, c \in C_e \tag{3.7}$$

La función objetivo (3.1) maximiza la satisfacción global de los estudiantes. Esto lo hace minimizando el valor total de preferencias, considerando que las asignaciones a preferencias mayores o iguales a k serán penalizadas por un factor de n . Las asignaciones a lista de espera también se penalizan con el multiplicador n , de forma tal que siempre cueste menos asignar a un estudiante a un turno por el que haya marcado preferencia que asignarlo a la lista de espera de

un curso. Las preferencias inferiores a k no reciben esta penalización adicional. Esto permite que el modelo tenga en cuenta de manera diferenciada las preferencias más altas y más bajas, promoviendo una asignación más justa, reflejando un tratamiento más riguroso de las asignaciones que generan mayor descontento.

Por otra parte, las restricciones aseguran que las asignaciones de estudiantes respeten las limitaciones del sistema, como la capacidad de los turnos y la unicidad de asignación para cada curso. Las restricciones (3.2) imponen que cada estudiante debe ser asignado a un único turno de un curso en el que haya expresado preferencias, o quedar en la lista de espera si no es posible asignarlo a ningún turno disponible. La cantidad de estudiantes asignados a un turno no puede exceder la capacidad del mismo, esto se logra con (3.3). Con (3.4) se considera que un estudiante no puede ser asignado simultáneamente a dos turnos que se superpongan, mientras que en (3.5) se impone que un estudiante no debe ser asignado a un curso o turno en el cual no ha expresado preferencias. Por último, (3.6) y (3.7) son las restricciones de dominio de las variables de decisión.

Minimización de la envidia global en la Fase 2

En el modelo propuesto para la Fase 2, se minimiza la envidia entre los estudiantes, respetando el nivel de satisfacción obtenido en la Fase 1. Además de las restricciones (3.2) - (3.7) que aplican para el modelo de la Fase 1, se introducen nuevas condiciones para minimizar la envidia sin reducir el nivel de satisfacción alcanzado previamente.

$$(\text{Fase 2}) : \min \sum_{e \in E} \sum_{f \in E: f \neq e} \sum_{c \in (C_e \cap C_f)} v_{efc} \quad (3.8)$$

sujeto a :

restricciones (3.2) - (3.7)

$$(w_{ec} - w_{fc})r_e + \sum_{t \in T_c: p_{ect} < u \vee p_{fct} < u} (x_{ect} - x_{fct})p_{ect} \leq v_{efc},$$

$$e \in E, f \in E, c \in C_e \cap C_f : e \neq f \quad (3.9)$$

$$\sum_{e \in E} \sum_{c \in C_e} \sum_{t \in T_c} p_{ect} x_{ect} + \sum_{e \in E} \sum_{c \in C_e} r_e w_{ec} \leq Opt_1 \quad (3.10)$$

$$v_{efc} \geq 0 \quad e \in E, f \in E : f <> e, c \in (C_e \cap C_f) \quad (3.11)$$

Las restricciones (3.9) definen formalmente la noción de envidia, respetando la definición dada en la Sección 3.1, considerando además que un estudiante puede quedar asignado a la lista de espera de un curso. Si suponemos que el valor óptimo de la Fase 1 es Opt_1 , entonces para garantizar que el nivel de satisfacción general se mantenga en la Fase 2, se debe imponer la restricción (3.10).

Este planteamiento formal de los dos modelos refleja cómo se estructuran las fases jerárquicas de la optimización: primero se maximiza la satisfacción de

los estudiantes, y luego, respetando esa optimización, se minimiza la envidia para lograr una asignación justa y equitativa de los turnos. Las restricciones aplicadas aseguran que las limitaciones del sistema se respeten en ambas fases, garantizando la factibilidad de las soluciones.

Unificación de ambas fases

A partir de los modelos anteriores (Fase 1 y Fase 2), se elaboró uno nuevo, combinándolos, para facilitar la implementación. Se utilizó un ponderador $a \in [0, 1]$ para la minimización de los valores de preferencias, y $(1 - a)$ para la minimización de la envidia. La función objetivo quedó entonces definida de la siguiente manera:

$$\begin{aligned} \min \quad & a \left(\sum_{e \in E} \sum_{c \in C_e} \sum_{t \in T_c: p_{ect} < k} p_{ect} x_{ect} + \sum_{e \in E} \sum_{c \in C_e} \sum_{t \in T_c: p_{ect} \geq k} n p_{ect} x_{ect} \right. \\ & \left. + \sum_{e \in E} \sum_{c \in C_e} n r_e w_{ec} \right) + (1 - a) \left(\sum_{e \in E} \sum_{f \in E: f \neq e} \sum_{c \in (C_e \cap C_f)} v_{efc} \right) \end{aligned}$$

Notar como con $a = 1$, la función objetivo es análoga a (3.1), y con $a = 0$, la función objetivo es análoga a (3.8). Introduciendo este parámetro a en los archivos de datos se puede decidir si minimizar el valor de preferencias o la envidia, dependiendo de si se le asigna el valor de 1 ó 0. Similarmente, se utilizó el parámetro a para activar y desactivar las restricciones (3.9) y (3.10) en base a si se busca minimizar el valor de preferencias o la envidia. El modelo unificado se puede visualizar en el Apéndice A.2.

A pesar de unificar ambas fases, el comportamiento del modelo unificado frente a un conjunto de datos con el parámetro de $a = 1$ será el mismo que el de la Fase 1, y con $a = 0$ será igual que la Fase 2, por lo que se seguirá refiriendo a ellas en el correr del informe.

3.3. Validación de los modelos matemáticos

Luego de construir los modelos, se procedió a validarlos; para ello, estos fueron sometidos a distintos casos de prueba con el fin de evaluar si sus comportamientos eran los esperados o si presentaban errores que requirieran modificaciones. Se elaboraron 10 casos de validación, con el objetivo de abarcar varios escenarios, como por ejemplo, cuando la capacidad es una limitante, cuando dos cursos se superponen, cuando un estudiante no expresa preferencias por ningún turno de ningún curso, entre otras. Antes de resolver los modelos con los casos de prueba, se detallaron las soluciones esperadas para cada uno de ellos, para luego verificar si las obtenidas eran las mismas o si había discrepancias. De esta manera, se identificó un error en la formulación inicial, y también distintos comportamientos dependiendo de cómo se definía el parámetro r , utilizado para penalizar asignaciones a listas de espera.

Caso 1: Capacidad insuficiente

El primer caso de prueba definido contempla un curso con un turno, con capacidad de 4, y 5 estudiantes, todos ellos marcando preferencia 1 por ese turno del curso. Se esperaba que en la Fase 1 uno de los estudiantes quedara asignado a lista de espera, mientras que los otros 4 quedaran asignados al turno en cuestión. El valor de r inicialmente dependía tanto del estudiante como del curso, y se definió de la siguiente manera:

$$r_{ec} = 1 + \max_{t \in T_c : p_{ect} < u} p_{ect} \quad (3.12)$$

Por lo tanto, se esperaba que el valor de preferencias fuera de 6 (4 asignaciones a preferencia 1, y una asignación a lista de espera, con un valor de 2), mientras que la envidia total fuera de 4 (el estudiante asignado a espera envidiaría con un valor de 1 a cada uno de los 4 estudiantes que quedaron asignados al turno). Efectivamente fue lo que sucedió. En cuanto a la Fase 2, se esperaba obtener la misma asignación, y que por lo tanto la envidia global fuera la hallada anteriormente. Esto fue lo que sucedió. Es importante destacar que si no se activa la restricción (3.10), que exige que el valor de preferencias sea menor o igual al valor óptimo hallado anteriormente, se podría lograr una envidia global de 0, al asignar a todos los estudiantes a lista de espera, así nadie envidia a nadie. Esta solución es claramente peor que la obtenida previamente, resaltando la importancia de la restricción de la Fase 2 de mantener el valor de preferencias en el valor óptimo obtenido en la Fase 1.

Caso 2: Turnos superpuestos

El segundo caso considerado consiste en 2 estudiantes y 2 cursos, uno de ellos con 2 turnos, mientras que el otro con tan solo 1, el cual se superpone con uno de los turnos del otro curso. Los valores de preferencias de los estudiantes por los turnos de los cursos se detallan en la Tabla 3.1.

preferencias	$c1t1$	$c1t2$	$c2t1$
$e1$	1	2	u
$e2$	2	u	1

Tabla 3.1: Valores de preferencias para el Caso 2 de validación

Aquí, ei hace referencia al estudiante i , y $citj$ al turno j del curso i . Por lo que en este caso, el estudiante $e1$ expresa preferencia 1 por el primer turno del primer curso ($c1t1$), y preferencia 2 por su segundo turno ($c1t2$), y no está interesado en el segundo curso, ya que $c2t1 = u$, mientras que el estudiante $e2$ expresa preferencia 1 por el primer turno del segundo curso ($c2t1$), y luego preferencia 2 por el primer turno del primer curso ($c1t1$), y no le interesa el turno 2 de ese curso, ya que $c1t2 = u$. De esta forma, con el valor de r definido en (3.12), se tiene que $r_{e1,c1} = 3$, $r_{e2,c1} = 3$, y $r_{e2,c2} = 2$ (con ci haciendo referencia al curso i).

Para la Fase 1, se esperaba que al estudiante $e1$ se le asignara $c1t1$, con un valor de 1, ya que lo prefiere frente a $c1t2$, y que no se asignara a ningún turno ni a lista de espera del curso $c2$ ya que no expresa preferencias por ningún turno de ese curso; y que al estudiante $e2$ se le asignara $c2t1$, con un valor de 1, y que quedara en lista de espera para $c1$, con un valor de 3. Esto es porque, si bien expresa preferencia por $c1t1$, dicho turno se superpone con $c2t1$ (que prefiere sobre $c1t1$), y, como no expresa preferencia por $c1t2$, debería ser asignado a lista de espera ya que no existiría ninguna asignación posible a un turno de dicho curso (si queda asignado a $c2t1$). El valor total de preferencias sería entonces de 5.

Sin embargo, la solución obtenida no fue la esperada, ya que, si bien las asignaciones del estudiante $e1$ se dieron de manera esperada, el estudiante $e2$ quedó asignado a $c1t1$, con un valor de 2, y en lista de espera para el curso $c2$, con un valor de 2. Efectivamente, el valor global de preferencias de esta solución también fue de 5, pero es claramente una peor solución que la esperada, ya que el estudiante $e2$ prefiere $c2t1$ sobre $c1t1$. El problema aquí es que el valor de r depende no solo del estudiante, sino que también del curso, por lo que el valor de r del estudiante $e2$ por el curso $c2$ es menor al valor de r del mismo estudiante por el curso $c1$ ($2 = r[e2, c2] < r[e2, c1] = 3$), lo que contrarrestó el valor mayor de la asignación al turno del curso menos preferido, $c1t1$. Debido a esto, se modificó la definición del parámetro r , para que solo dependiese del

estudiante, y no del curso:

$$r_e = 1 + \max_{c \in C_e, t \in T_c : p_{ect} < u} p_{ect} \quad (3.13)$$

Con el nuevo valor de r , que solo depende del estudiante, efectivamente se cumple lo esperado. El estudiante $e1$ se asigna a $c1t1$, con un valor de 1, y el estudiante $e2$ se asigna a $c2t1$, con un valor de 1, y queda en lista de espera para $c1$, con un valor de 3. Cabe destacar que con esta modificación, la solución que antes era óptima tiene un valor una unidad mayor que la nueva solución óptima, ya que asignar el estudiante $e2$ a $c1t1$ valdría 1 más, pero quedar en lista de espera para $c2$ ahora vale lo mismo que quedar en lista de espera para $c1$ (antes valía 1 menos, por lo que se balanceaba). Es importante mencionar que la nueva definición de r no tiene efecto alguno sobre el primer caso, ya que en él solo había un curso.

En cuanto a la envidia, se esperaba que el estudiante $e1$ tuviera un valor de envidia de 0, ya que quedaría asignado a su preferencia 1 por el curso $c1$, y no quedaría asignado a ningún turno ni a la lista de espera por el curso $c2$. Para el estudiante $e2$, se esperaba que tuviera un valor de envidia de 1, ya que quedaría asignado a espera por el curso $c1$, mientras que el estudiante $e1$ quedaría asignado a la preferencia 2 del estudiante $e2$ por el curso $c1$. Luego de realizar el arreglo del valor de r , efectivamente fue lo que sucedió. Además, se esperaba que para la Fase 2 la asignación fuera la misma y, por lo tanto, que la envidia global se mantuviera, que fue lo que sucedió.

Caso 3: Un estudiante no expresa preferencias por un curso

Otro caso que se consideró fue el de evitar asignaciones si un estudiante no expresa preferencias por ningún turno de un curso. Lo esperado en ese caso es que no quede asignado a ningún turno de dicho curso, ni a su lista de espera. Inicialmente, se podría esperar que esto suceda sin tener que implementar ninguna restricción adicional, ya que asignar estudiantes a turnos o a listas de espera tiene asociado un valor de preferencias, que el modelo de la Fase 1 busca minimizar. Es por ello que, a priori, sería razonable suponer que nunca se asignaría a ningún estudiante a turnos o a la lista de espera de cursos por los que ellos no expresan preferencias. Pero, al analizar la función objetivo, se puede observar que para cada estudiante solo se consideran los cursos por los cuales dicho estudiante expresa preferencias. Esto se hace para reducir la cantidad de operaciones a realizar por el *solver*¹, buscando que sea lo más eficiente posible. Por esta razón, se podría asignar (o no) a un estudiante a un turno de un curso por el cual no expresa preferencias, o a su lista de espera, y no afectaría a la solución óptima (ya que no se consideraría en la sumatoria). Debido a esto, es necesario incluir una restricción adicional que evite realizar dichas asignaciones. El caso anterior consideraba esto, ya que uno de los estudiantes no expresaba

¹Software diseñado para encontrar soluciones óptimas o factibles a problemas matemáticos de optimización

preferencias por el único turno de uno de los cursos; pero en este caso de prueba, se simplifica el conjunto de datos (se considera solo un estudiante) y, además, se utiliza un valor absurdo (-10) en la preferencia del turno del curso que el estudiante no desea ser asignado. Los valores de preferencias del estudiante por los turnos de los cursos se detallan en la Tabla 3.2.

preferencias	$c1t1$	$c1t2$	$c2t1$
$e1$	1	2	-10

Tabla 3.2: Valores de preferencias para el Caso 3 de validación

Si bien los turnos a los que un estudiante no quiere ser asignado se deberían marcar con preferencia u , aquí se utiliza el valor de -10 para resaltar que incluso en el caso de que mejoraría la solución óptima si se pudiera considerar ese turno (si se cambia el modelo para que la función óptima considere todos los cursos), no se asignaría a ese turno ya que el estudiante no le interesa ningún turno de ese curso. Se esperaba que el estudiante quedara asignado a $c1t1$, y que no quedara asignado a ningún otro turno ni a espera de ningún curso, y que la envidia fuera 0. Efectivamente fue lo que sucedió. Luego se quitó la restricción que evitaba este tipo de asignaciones y se pudo apreciar que el estudiante también quedaba asignado a $c2t1$, logrando así un valor objetivo de -9, mostrando la importancia de dicha restricción.

Caso 4: Validación de la penalización de asignaciones a espera

Para validar el comportamiento de la familia de parámetros utilizados para penalizar las asignaciones a listas de espera, r_e , que ya había sido modificada previamente debido a comportamientos incoherentes con su formulación inicial, se procedió a definir un conjunto de datos con 2 estudiantes y 2 cursos, cada uno con 3 turnos. Ambos estudiantes desean ser asignados a ambos cursos, pero un estudiante marca preferencias por todos los turnos de ambos cursos, mientras que otro estudiante marca preferencias por solo un turno de cada curso. Se considera una capacidad de 1 para cada turno de cada curso, así solo un estudiante podría quedar dentro, dejando al otro estudiante fuera. Aquí se busca mostrar que el estudiante que marca más preferencias ($e1$) se ve favorecido, ya que su valor de r_e será mayor que el valor de r_e del otro estudiante, ($e2$), por lo que asignar a $e1$ a espera tendría un valor mayor que asignar a $e2$. Los valores de preferencias por los turnos de los cursos se detallan en la Tabla 3.3.

preferencias	$c1t1$	$c1t2$	$c1t3$	$c2t1$	$c2t2$	$c2t3$
$e1$	1	3	4	5	2	6
$e2$	1	u	u	u	2	u

Tabla 3.3: Valores de preferencias para el Caso 4 de validación

Se esperaba que el estudiante $e1$ quedara asignado a $c1t1$ y $c2t2$, con un valor de 3, y que el estudiante $e2$ quedara en espera para los cursos $c1$ y $c2$, con

un valor de 6, obteniendo así un valor total de preferencias de 9. Además, se esperaba que la envidia total fuera de 3, ya que el estudiante $e2$ envidiaría al estudiante $e1$ con un valor de 2 por el curso $c1$, y con un valor de 1 por el curso $c2$. Efectivamente, fue lo que sucedió. En este caso, se favorece al que expresa una mayor cantidad de preferencias.

En cambio, al modificar la definición de r_e , pasando a ser una constante (sin depender del estudiante), se esperaba que el comportamiento fuera distinto. Considerando el mismo caso de prueba, con la siguiente modificación: $r_e[e1] = r_e[e2] = 10$, se esperaba que el estudiante $e2$ fuera favorecido, ya que al marcar preferencias por tan solo un turno de cada curso, si no quedaba asignado a estos, quedaría asignado a la lista de espera de dichos cursos, con un valor elevado, mientras que el estudiante $e1$ podría ser asignado a otros turnos menos preferidos, ya que marcó preferencias por todos, lo que tendría un valor levemente mayor, pero permitiría poder asignar al estudiante $e2$ a los turnos que prefiere, reduciendo significativamente el valor global. Notar cómo con este nuevo conjunto de datos, la asignación obtenida anteriormente tendría un valor objetivo de $1 + 2 + 10 + 10 = 23$. Pero, si se favoreciera a $e2$, quedaría asignado a $c1t1$ con un valor de 1, y a $c2t2$ con un valor de 2, mientras que $e1$ quedaría asignado a $c1t2$ con un valor de 3, y $c2t1$ con un valor de 5, por lo que el valor total de preferencias sería de 11. Con esta solución, la envidia total sería de 5 (el estudiante $e1$ envidiaría a $e2$ con un valor de $3 - 1 = 2$ por $c1$, y con un valor de $5 - 2 = 3$ por $c2$). La solución obtenida por el modelo fue la esperada.

Elegir una de las definiciones vistas anteriormente para este parámetro no es sencillo, ya que cada una implica compensaciones. Por un lado, priorizar a los estudiantes que marcan más preferencias podría resultar en que aquellos que marcan pocas terminen en listas de espera. Esto podría considerarse injusto, ya que es posible, y esperable, que haya estudiantes con una disponibilidad de horarios reducida, y que por lo tanto solo les sean útiles pocos turnos, mientras que otros estudiantes tendrán una mayor disponibilidad y, por lo tanto, podrán marcar preferencias por una gran cantidad de turnos, desfavoreciendo así a los estudiantes con baja disponibilidad de horarios. Por otro lado, si se favorece a quienes seleccionan menos preferencias, aunque esto podría reducir el número de estudiantes en listas de espera, algunos estudiantes podrían aprovechar la situación y marcar preferencia por solo un turno por cada curso a los que quieren asistir, a pesar de tener disponibilidad para varios turnos. Esto aumentaría la probabilidad de quedar asignados en sus turnos más preferidos, en detrimento de los estudiantes que, de manera más honesta, marcaron preferencias por todos los turnos a los que podrían asistir.

Tras considerar las ventajas y desventajas de ambos comportamientos, se decidió mantener la definición de r dependiente de cada estudiante, como se especifica en (3.13). De este modo, se reconoció que aquellos que indiquen una mayor cantidad de preferencias probablemente serán favorecidos frente a quienes indiquen menos. Con esto, se busca evitar que el funcionamiento del modelo

de la Fase 1 permita a estudiantes obtener ventajas personales al no marcar preferencias por turnos que realmente les podrían servir. De todas formas, es posible modificar los archivos de datos y utilizar un mismo valor para todos los r_e , siempre y cuando el mismo sea mayor que el máximo valor de preferencia asignado entre todos los estudiantes, cursos y turnos, y el modelo lo resolverá de manera acorde.

Caso 5: Preferencias expresadas por curso

Hasta ahora, las preferencias asignadas por los estudiantes no diferenciaban entre los distintos cursos disponibles. Es decir, se marcaban preferencias globalmente, considerando todos los turnos de todos los cursos, sin repetir valores de preferencias. Un enfoque alternativo sería discriminar entre cursos. Esto implicaría comenzar por un curso y asignar valores de preferencias a sus turnos, empezando desde 1, y luego se pasaría al siguiente curso, y se volverían a marcar preferencias por sus turnos, empezando nuevamente desde 1, y así sucesivamente. Se esperaba que los modelos se comportaran adecuadamente frente a este nuevo enfoque, sin requerir modificaciones. Se elaboró un conjunto de datos de prueba con 2 estudiantes, y 3 cursos con 2 turnos cada uno. Cada estudiante expresa preferencias para cada turno de cada curso, pero para cada curso puede asignar preferencias por separado, tal como fue explicado anteriormente. De esta manera, los valores de preferencia asignados se detallan en la Tabla 3.4.

preferencias	$c1t1$	$c1t2$	$c2t1$	$c2t2$	$c3t1$	$c3t2$
$e1$	1	2	1	2	2	1
$e2$	1	2	2	1	1	2

Tabla 3.4: Valores de preferencias para Caso 5 de validación

Dado que este caso no considera limitantes de capacidad ni superposiciones de turnos, se esperaba que los estudiantes fueran asignados a los turnos cuyos valores de preferencia son 1. Es decir, se esperaba que $e1$ fuera asignado a $c1t1$, $c2t1$, y $c3t2$, y que $e2$ fuera asignado a $c1t1$, $c2t2$, y $c3t1$, con un valor objetivo de 6. Además, se esperaba que la envidia global fuera 0, ya que al ser asignados a sus preferencias 1, no deberían envidiarse entre ellos. Para ambas fases, la solución obtenida fue la descrita anteriormente, por lo que los modelos se comportaron correctamente frente a este nuevo planteamiento.

Caso 6: Penalización de preferencias demasiado altas

Para penalizar preferencias que se consideren demasiado altas, se introdujeron 2 parámetros en los modelos, k y n . El valor de k es el valor considerado “demasiado alto”, por lo que asignaciones a preferencias mayores o iguales a k serán penalizadas. La penalización está dada por el valor de n , que se utiliza en la función objetivo, multiplicando a las preferencias mayores o iguales a k . Se desarrolló un conjunto de datos con 3 estudiantes, y 1 curso con 3 turnos, todos ellos con capacidad 1. Las preferencias expresadas por los estudiantes se detallan en la Tabla 3.5.

preferencias	$c1t1$	$c1t2$	$c1t3$
$e1$	1	4	4
$e2$	4	1	4
$e3$	3	3	5

Tabla 3.5: Valores de preferencias para el Caso 6 de validación

Si bien es un caso extraño ya que la asignación de preferencias no comienza desde 1, y a veces se repiten preferencias, ya se vio anteriormente que los modelos propuestos son capaces de procesar valores de preferencias que no siguen cierta lógica. Solamente se debe procurar que se asignen valores menores a u a los turnos que se prefieren e iguales o mayores a u a los turnos que no se prefieren, por lo que este caso es válido. Este caso se diseñó de forma tal de que los modelos se comportasen distinto al introducir los parámetros k y n . Al no penalizar preferencias altas ($n = 1$ y/o $k > 5$), se esperaba que la solución óptima fuera asignar el estudiante $e1$ a $c1t1$, el estudiante $e2$ a $c1t2$, y el estudiante $e3$ a $c1t3$, con un valor total de preferencias de 7, y una envidia de 4. Efectivamente, esta fue la solución óptima obtenida. Por otra parte, al considerar $k = 5$ y $n = 2$, se esperaba que $e3$ no quedara asignado a su preferencia 5, como sucedió anteriormente, sino que a alguna de sus preferencias 3 (es indistinto cuál), mientras que $e1$ y $e2$ quedaran asignados uno en su preferencia 1, y el otro en su preferencia 4 (dependiendo de qué asignación se realice para $e3$). Aquí el valor total de preferencias sería 8, y la envidia total sería de 3, ya que, dependiendo de cuál de las 2 soluciones esperadas posibles se obtuviera, $e1$ o $e2$ envidiaría a $e3$ con un valor de 3, mientras que los demás estudiantes no envidiarían a nadie. En ambas fases, la solución obtenida fue una de las esperadas, mostrando así la utilidad de estos parámetros.

Caso 7: Un estudiante no expresa preferencias

Luego, se consideró un caso con un estudiante, y 2 cursos con 2 turnos cada uno, y el estudiante no expresa preferencias por ningún turno de ningún curso. Aquí se esperaba que el estudiante no quedara asignado a ningún turno, ni a ninguna lista de espera, y que el valor de preferencias y la envidia fueran ambos 0, que fue lo que efectivamente sucedió al resolver tanto la Fase 1 como la Fase 2.

Caso 8: Todos los turnos con capacidad 1

Por otra parte, se consideró un caso con 3 estudiantes, y un curso con 3 turnos, todos con capacidad 1. Un estudiante expresa preferencia uno por todos los turnos, mientras que los demás estudiantes expresan preferencias variadas, como se detallan en la Tabla 3.6.

preferencias	$c1t1$	$c1t2$	$c1t3$
$e1$	1	1	1
$e2$	1	2	3
$e3$	2	3	1

Tabla 3.6: Valores de preferencias para el Caso 8 de validación

Se esperaba que para la Fase 1 el estudiante $e1$ quedara asignado al turno $c1t2$, y que $e2$ quedara asignado a $c1t1$, mientras que $e3$ quedara en $c1t3$, lo que implicaría una asignación a preferencias 1 para todos los estudiantes. Por lo tanto, el valor de la asignación sería de 3, y la envidia de 0. Por esta razón, se esperaba obtener la misma solución óptima en la Fase 2 (ya que la envidia global sería 0 en la Fase 1). Los resultados obtenidos fueron consistentes con los esperados.

Caso 9: Todos los turnos superpuestos

Otro caso límite considerado fue uno en el que todos los turnos de todos los cursos se superponían. Se definió un caso con 5 estudiantes, y 3 cursos con un turno cada uno, en donde la capacidad no era una limitante para el problema. Las preferencias marcadas por los estudiantes hacia los turnos se detallan en la Tabla 3.7.

preferencias	$c1t1$	$c1t2$	$c1t3$
$e1$	3	2	1
$e2$	1	2	u
$e3$	1	u	u
$e4$	2	u	1
$e5$	u	1	u

Tabla 3.7: Valores de preferencias para el Caso 9 de validación

Se esperaba que los estudiantes fueran asignados al turno del curso que marcaban con preferencia 1, y quedaran en espera para el resto de los cursos, si expresaban preferencias por ellos. La solución óptima coincidió con lo esperado.

Caso 10: Todos los turnos superpuestos y con capacidad 1

Por último, se consideró el mismo caso que el anterior, solo que se modificó la capacidad de los turnos a 1, por lo que la solución óptima obtenida en el caso anterior no sería factible para este nuevo problema, ya que en ella se habían asignado dos estudiantes ($e2$ y $e3$) a $c1t1$, y otros dos ($e1$ y $e4$) a $c3t1$, y ahora la capacidad de dichos turnos es tan solo de 1. Como se vio anteriormente, con el valor de la familia de parámetros r_e dependiendo de cada estudiante, según la ecuación (3.13), el modelo de la Fase 1 parecería priorizar a los estudiantes que expresan más preferencias. Es por ello que se esperaba que ahora el modelo priorizara a $e2$ frente a $e3$, y a $e1$ frente a $e4$, obteniendo la siguiente asignación: $e1$ sería asignado a $c3t1$, y quedaría en espera para el resto de los cursos; $e2$ sería asignado a $c1t1$ y quedaría en espera por el curso $c2$; $e3$ quedaría en espera por $c1$, $e4$ quedaría en espera por $c1$ y $c3$; y $e5$ sería asignado a $c2t1$. Esto es lo que sucedió al resolver los modelos con este conjunto de datos, tanto para la Fase 1 como para la Fase 2.

Capítulo 4

Experimentación numérica

Una vez validado el modelo de la Fase 1 y el de la Fase 2, se procedió a generar y resolver un caso base, y luego realizar un análisis de sensibilidad. Esto ayuda a entender cómo se comportan los modelos frente a una variedad de escenarios, permitiendo identificar qué cambios en los parámetros afectan de qué forma a las soluciones obtenidas a partir de los modelos.

Se definió y resolvió un caso base, y a partir de ello se realizó el análisis de sensibilidad. Para realizar dicho análisis, fue necesario definir conjuntos de datos, variando los distintos parámetros de entrada que los modelos requieren, y junto a ello, definir ciertos indicadores, de manera de poder comparar las soluciones al utilizar distintos conjuntos de datos. En las siguientes secciones se presentarán los conjuntos de datos utilizados, y se detallará cómo fueron generados (Sección [4.1](#)); luego se detallará la selección de indicadores para evaluar el comportamiento de los modelos (Sección [4.2](#)), y finalmente se analizarán las soluciones obtenidas con los distintos conjuntos de datos a partir de los indicadores (Sección [4.3](#)).

4.1. Generación de datos

El método para definir los conjuntos de datos a generar fue el siguiente: se tomó un caso base y se modificaron, uno por uno, los parámetros y conjuntos de entrada, manteniendo el resto fijos. De esta forma, se visualizaría cómo los modelos de la Fase 1 y Fase 2 responden a variaciones individuales. Dicho esto, el primer paso fue analizar estos modelos, para así identificar a los conjuntos y parámetros de entrada que podían ser modificados, y cuya variación resultase interesante considerando tanto el problema en cuestión como el comportamiento de los modelos. Luego, se podría elegir un valor base para ellos, y así generar el conjunto de datos base.

4.1.1. Elementos variables

Variaciones de la cantidad de estudiantes y cursos

Analizar cómo se comportarían los modelos frente a cambios en el tamaño del conjunto de estudiantes, y a cambios en el tamaño del conjunto de cursos, sería interesante principalmente desde un punto de vista computacional. Es decir, sería interesante analizar los tiempos de resolución que se obtendrían con distintos *solvers* al variar los tamaños de dichos conjuntos por separado, y así poder ver si los modelos propuestos podrían resolver casos reales en tiempos razonables.

Para que tenga sentido analizar la variación del tamaño del conjunto de estudiantes desde un punto de vista computacional, también tendría que variar la capacidad de los turnos de los cursos de la misma manera, es decir, si la cantidad de estudiantes se triplica, también debería triplicarse la capacidad de los turnos. De esta forma, solo se esperaría que varíe el tiempo de resolución del *solver*, y no otros indicadores, como podrían ser la envidia media de un estudiante, o la preferencia media asignada a los estudiantes. La variación del tamaño del conjunto de cursos no tendría que ser acompañada con la variación de otro parámetro, ya que variar únicamente dicho tamaño ya sería suficiente para tan solo escalar el problema sin modificar los indicadores mencionados anteriormente.

Variaciones de la cantidad y capacidad de turnos

Por otro lado, se podría variar la cantidad de turnos que tiene cada curso, variando así la cantidad de opciones que tendría un estudiante para participar en un curso, y por ello sería esperable que las soluciones varíen. Por ejemplo, si la cantidad de turnos disponibles para un curso es mayor, los estudiantes tendrán más turnos para elegir y, por lo tanto, se diversificarán las preferencias, lo que probablemente implicaría mejores asignaciones, tanto desde el punto de vista de las preferencias como de la envidia. Además, se podría variar la capacidad de los turnos disponibles para los cursos, lo que claramente afectaría los indicadores de las soluciones obtenidas relacionados con valores de preferencias y

envidias. Es fácil ver que un aumento en la capacidad de los turnos de los cursos implicaría que más estudiantes podrían ser asignados a sus turnos preferidos que si la capacidad de los turnos fuera reducida, y gracias a ello se obtendrían asignaciones con menores valores de preferencias y envidias.

Variaciones de la cantidad de superposiciones, y las preferencias elevadas

Otro valor que resultaría interesante modificar es el de la cantidad de superposiciones de turnos que un conjunto de datos tiene. Mientras mayor sea la cantidad de superposiciones existentes en un conjunto de datos, más se dificultaría la asignación de estudiantes a turnos respetando sus preferencias, por lo que se esperaría obtener peores asignaciones que si la cantidad de superposiciones fuera menor. Como ya se vio anteriormente en la sección de validación de los modelos, utilizar distintos valores de k (valor considerado como preferencia demasiado alta) y n (multiplicador asociado con las preferencias consideradas demasiado altas) repercute en las soluciones obtenidas y, por lo tanto, sería interesante analizar dichas soluciones frente a la variación de estos parámetros.

Variaciones de las preferencias de los estudiantes

Por último, dado que la generación de datos fue realizada de forma automatizada, y para definir las preferencias de los estudiantes se utilizó un parámetro m , resultaría interesante observar cómo se comportan las soluciones al variar dicho parámetro. En términos sencillos, el parámetro m hace referencia a que el turno i es m veces más probable de ser preferido frente al turno $i + 1$. Es por ello que distintos valores de m generarían distintos conjuntos de datos, que en principio parecería que podrían afectar en ciertos aspectos a las soluciones óptimas obtenidas. Por ejemplo, un valor de $m = 1$ implicaría que las preferencias de cada estudiante se generen equitativamente entre los distintos turnos de un curso. En cambio, valores elevados de m implicarían una generación de preferencias en la que la mayoría preferiría el turno 1, luego el turno 2, y así sucesivamente.

4.1.2. Generación del caso base y configuraciones

Ya habiendo identificado los conjuntos y parámetros que se variarían para generar los distintos conjuntos de datos, se procedió a definir los valores iniciales para estos. La cantidad de estudiantes de la configuración base sería de 30, lo suficientemente grande como para poder observar tendencias globales de las soluciones al variar otros parámetros, pero no demasiado grande como para que los modelos tarden demasiado en obtener dichas soluciones. Similarmente, se configuró en 3 el valor base para la cantidad de cursos, y para la cantidad de turnos de cada curso. La capacidad de cada turno de cada curso se estableció en 9, por lo que los cursos tendrían una capacidad total de 27 estudiantes, y, dado que hay 30 estudiantes en la configuración base, 3 estudiantes necesariamente deberían ser asignados a la lista de espera de cada curso. Se decidió que la configuración base no tendría superposiciones entre turnos, y que no se penalizarían preferencias altas, ajustando el valor de k a 10, y el de n a 1. El valor de m se estableció en 1.5, para generar datos de preferencias en donde para cada curso el turno 1 sea el más popular, luego el 2, y luego el 3.

Se obtuvieron distintas configuraciones al variar por separado la cantidad de estudiantes (junto con la capacidad de los turnos), la cantidad de cursos, la cantidad de turnos por curso y la capacidad de los turnos. Estas configuraciones se clasificaron en las siguientes categorías: “**variaciones de E** ”, “**variaciones de C** ”, “**variaciones de T_c** ”, y “**variaciones de q** ”. Dentro de cada categoría, la variación siguió el mismo patrón: la primera configuración consideró 1/3 del valor base del elemento variable, la segunda el doble del valor base, la tercera el cuádruple, la cuarta 10 veces el valor base, y la quinta 50 veces el valor base (para las configuraciones de las variaciones de E , se consideraron como elementos variables tanto la cantidad de estudiantes como la capacidad de los turnos). Tomando estos valores, se logró considerar casos pequeños, regulares y numerosos, para así poder analizar el comportamiento de los modelos y las soluciones obtenidas para distintas realidades.

Para considerar las distintas configuraciones para las variaciones de la cantidad de superposiciones de turnos, primero se consideró la cantidad máxima de pares de turnos que se podrían superponer considerando la cantidad de cursos y turnos por curso de la configuración base. Su cálculo se detalla en el Apéndice A.3, en donde se puede observar que el valor es 36. En base a esto, se definieron las siguientes configuraciones para la categoría de “**variaciones de $|S_{ij}|$** ”: que haya 5, 10, 15, 20 y 30 superposiciones (manteniendo el resto de los elementos variables con los valores de la configuración base).

Para generar las configuraciones dentro de la categoría “**variaciones de k y n** ”, se estableció el valor de k en 3, y se varió el valor de n entre 1.5, 2, 3, y 5. Notar cómo no tendría sentido establecer el valor de k en 1, 2, o cualquier valor mayor a 3, ya que penalizar preferencias mayores o iguales a 1, penalizaría a todas las asignaciones; penalizar preferencias mayores o iguales a 2 intentaría

favorecer asignaciones a preferencias 1, que es lo que ya se intenta hacer en la Fase 1; y penalizar preferencias mayores a 3, no afectaría en estos conjuntos de prueba ya que la máxima preferencia que expresan los estudiantes es 3, al haber solo 3 turnos para cada curso. Por otro lado, para generar las configuraciones dentro de la categoría “**variaciones de m** ”, se varió dicho parámetro, tomando los siguientes valores: 1, 2, 4 y 10.

Además, se consideraron variaciones conjuntas para ciertos pares de elementos particulares. Un par de elementos que resultó interesante variar en conjunto fue el de la cantidad de estudiantes y la cantidad de cursos. Se generaron configuraciones en las cuales la cantidad de estudiantes aumentaba mientras la de cursos disminuía. A esta categoría se le llamó “**variaciones $E(+)$ $C(-)$** ”, haciendo referencia a que la cantidad de estudiantes aumenta mientras la cantidad de cursos disminuye. Esta configuración busca analizar si sería más complicado para el *solver* resolver problemas con muchos estudiantes pero pocos cursos, o al revés, o si no afectaría demasiado. También se generaron configuraciones en donde la cantidad de estudiantes y la de cursos aumentaban a la vez, pertenecientes a la categoría “**variaciones $E(+)$ $C(+)$** ”, para ver si se observaba un aumento considerable en los tiempos de resolución con datos generados a partir de estas configuraciones. Por último, se consideró variar el par de elementos de la cantidad de turnos y la capacidad de los turnos a la vez, buscando analizar lo que sucedería al aumentar uno mientras disminuye el otro, obteniendo así ciertas configuraciones pertenecientes a la categoría “**variaciones $q(+)$ $t(-)$** ”. Esto resultaría interesante para discutir sobre si en un caso real sería mejor introducir más turnos o aumentar la capacidad de los existentes.

Un resumen de las distintas configuraciones consideradas se puede ver en la Figura 4.1, en donde se resaltan en rojo los valores de los elementos variables que son distintos a los valores que toman en la configuración base.

Pueba	E	C	T_c	q	S_ij	k	n	m
Base	30	3	3	9	0	10	1	1.5
Variaciones de E	10	3	3	3	0	10	1	1.5
	60	3	3	18	0	10	1	1.5
	120	3	3	36	0	10	1	1.5
	300	3	3	90	0	10	1	1.5
	1500	3	3	450	0	10	1	1.5
Variaciones de C	30	1	3	9	0	10	1	1.5
	30	6	3	9	0	10	1	1.5
	30	12	3	9	0	10	1	1.5
	30	30	3	9	0	10	1	1.5
	30	150	3	9	0	10	1	1.5
Variaciones de T_c	30	3	1	9	0	10	1	1.5
	30	3	6	9	0	10	1	1.5
	30	3	12	9	0	10	1	1.5
	30	3	30	9	0	10	1	1.5
	30	3	150	9	0	10	1	1.5
Variaciones de #S_ij	30	3	3	9	5	10	1	1.5
	30	3	3	9	10	10	1	1.5
	30	3	3	9	15	10	1	1.5
	30	3	3	9	20	10	1	1.5
	30	3	3	9	30	10	1	1.5
Variaciones de q	30	3	3	3	0	10	1	1.5
	30	3	3	18	0	10	1	1.5
	30	3	3	36	0	10	1	1.5
	30	3	3	90	0	10	1	1.5
	30	3	3	450	0	10	1	1.5
Variaciones de k y n	30	3	3	9	0	3	1.5	1.5
	30	3	3	9	0	3	2	1.5
	30	3	3	9	0	3	3	1.5
	30	3	3	9	0	3	5	1.5
Variaciones de m	30	3	3	9	0	10	1	1
	30	3	3	9	0	10	1	2
	30	3	3	9	0	10	1	4
	30	3	3	9	0	10	1	10
E(-,+) C(+,-)	10	150	3	3	0	10	1	1.5
	30	30	3	9	0	10	1	1.5
	60	12	3	18	0	10	1	1.5
	120	6	3	36	0	10	1	1.5
	300	3	3	90	0	10	1	1.5
	1500	1	3	450	0	10	1	1.5
E(-,+) C(-,+)	10	1	3	3	0	10	1	1.5
	60	6	3	18	0	10	1	1.5
	120	12	3	36	0	10	1	1.5
	300	30	3	90	0	10	1	1.5
	1500	150	3	450	0	10	1	1.5
q(-,+) t(+,-)	30	3	150	3	0	10	1	1.5
	30	3	30	9	0	10	1	1.5
	30	3	12	18	0	10	1	1.5
	30	3	6	36	0	10	1	1.5
	30	3	3	90	0	10	1	1.5
	30	3	1	450	0	10	1	1.5

Figura 4.1: Resumen de las distintas configuraciones del análisis de sensibilidad

4.1.3. Generador de datos

Para cada una de las configuraciones detalladas anteriormente, se generaron 7 conjuntos de datos a partir de un generador de datos diseñado en Visual Studio Code¹, utilizando el lenguaje de programación Python², y consultando con ChatGPT³ para desarrollarlo. Dado que se consideraron 51 configuraciones distintas, se generaron en total 357 conjuntos de datos para la experimentación numérica.

El generador comienza definiendo los conjuntos y parámetros editables, que son los ya mencionados anteriormente referidos como los elementos variables de las configuraciones. Estos son la cantidad de estudiantes, cursos, y turnos por curso, la capacidad de los turnos, la cantidad de pares de superposiciones, el valor de preferencia considerado demasiado elevado, y el multiplicador asociado a las preferencias elevadas. Además, se agregan otros 2 parámetros editables por el usuario: el ponderador de la función objetivo, que hace referencia a si se busca minimizar las preferencias o la envidia, y el nombre que se le quiere dar al archivo que se generará al correr el código. En base a los valores que el usuario detalla dentro de esta sección editable, al correr el código, se generará un archivo que contiene un conjunto de datos que corresponde con la configuración detallada. El código utilizado en el generador de datos se encuentra en el Apéndice A.4.

4.2. Indicadores

Luego de generar los conjuntos de datos a utilizar para el análisis de sensibilidad, se definieron los indicadores a utilizar para analizar las soluciones obtenidas con cada configuración. La medición cuantitativa de las soluciones a través de estos indicadores permitiría analizar cualitativamente el comportamiento de los modelos frente a las variaciones de cada elemento variable.

Para las configuraciones dentro de variaciones de E , y C , solamente se utilizó el tiempo de resolución del *solver* como indicador, y en caso de no obtener una solución óptima, se tomó la mejor solución encontrada, y el *mipgap*⁴ obtenido por el *solver*. De esta manera, sería posible comparar distintos *solvers* y así elegir uno con el cual seguir adelante con el resto de las configuraciones elaboradas para la experimentación numérica.

¹<https://code.visualstudio.com/>

²<https://www.python.org/>

³<https://chat.openai.com>

⁴Indicador que representa el porcentaje de diferencia entre el valor de la solución óptima del problema relajado (sin restricciones enteras), y el valor de la mejor solución factible encontrada

Los indicadores utilizados para el resto de las configuraciones fueron los siguientes:

- Valor total de preferencias
- Preferencia mínima asignada a un estudiante
- Preferencia máxima asignada a un estudiante
- Preferencia media asignada a los estudiantes
- Distribución de preferencias (Curtosis⁵)
- Cantidad total de asignaciones a lista de espera
- Cantidad mínima de asignaciones a lista de espera para un estudiante
- Cantidad máxima de asignaciones a lista de espera para un estudiante
- Cantidad promedio de asignaciones a lista de espera por estudiante
- Valor total de envidias
- Envidia mínima de un estudiante
- Envidia máxima de un estudiante
- Envidia media por estudiante
- Distribución de envidias (Curtosis)

El uso de estos indicadores también permitirá comparar el funcionamiento de la Fase 1 (cuando se está minimizando el valor de preferencias) y la Fase 2 (cuando se está minimizando el valor de envidias). Si bien el valor total de preferencias debería mantenerse en ambas instancias, los demás indicadores podrían, e idealmente deberían, variar. Esto es porque en la Fase 2 se busca mantener el valor total de preferencias, pero encontrar una distribución de preferencias distinta a la obtenida en la Fase 1, que mejore (disminuya) el valor total de envidias.

⁵Medida estadística que describe la forma de la distribución de un conjunto de datos. En este informe, cuando se habla de “Curtosis” se hace referencia al exceso de curtosis. En este contexto, un valor de Curtosis igual a 0 significa que la distribución es Gaussiana (Normal). Cuanto más positivo sea el valor de Curtosis, más “puntiaguda” será la forma de la distribución. Cuanto más negativo sea el valor de Curtosis, lo opuesto sucede; más “plana” será la forma de la distribución.

4.3. Resultados

En esta sección, se analizarán los valores de los indicadores obtenidos al resolver los modelos con los distintos conjuntos de datos generados. Se utilizó el lenguaje AMPL⁶, junto con el editor AMPL IDE, que permite utilizar tanto *solvers* gratuitos como comerciales. Se elaboró un archivo *.run* para resolver los modelos junto con los distintos conjuntos de datos elaborados y extraer los valores de los indicadores a partir de las soluciones obtenidas. Este archivo se puede visualizar en el Apéndice A.5. Para la mayoría de los casos, se utilizó el *solver* CPLEX⁷, pero en algunos se utilizó también el *solver* Gurobi⁸.

Si bien se generaron 7 conjuntos para cada una de las 51 configuraciones consideradas, en la mayoría de las configuraciones se utilizaron solo 5. El *solver* frecuentemente demoraba un tiempo considerable en hallar una solución óptima para algunos casos, o incluso llegaba al tiempo límite definido para la prueba sin haberla encontrado, por lo que analizar todos los casos elaborados habría requerido demasiado tiempo.

A continuación se detallarán los resultados obtenidos dentro de cada categoría considerada. Como se mencionó en la Sección 4.2, las variaciones de la cantidad de estudiantes y cursos tienen como propósito elegir el *solver* a utilizar en las siguientes pruebas, a partir de los indicadores de tiempos de resolución y *mipgaps* obtenidos por cada *solver*, por lo que sus resultados serán analizados en la Sección 4.3.1, “Resultados del análisis computacional”. Por otra parte, el resto de las variaciones (las de cantidad de turnos por curso, capacidad de los turnos, superposiciones, penalizadores de preferencias altas, generador de preferencias, y las mixtas de capacidad y cantidad de turnos) se analizará en la Sección 4.3.2, “Resultados del análisis de sensibilidad”. Para cada categoría, se analizará el comportamiento de cada indicador al aumentar/disminuir el valor del elemento variable considerado en dicha categoría, diferenciando entre los resultados obtenidos en la Fase 1 (minimizando el valor de preferencias) y la Fase 2 (minimizando el valor de envidias).

⁶<https://ampl.com>

⁷<https://www.ibm.com/products/ilog-cplex-optimization-studio>

⁸<https://www.gurobi.com>

4.3.1. Resultados del análisis computacional

Las variaciones de la cantidad de estudiantes y las de la cantidad de cursos se analizarán en conjunto ya que su principal función era decidir qué *solver* utilizar para el resto de las pruebas, y no analizar las soluciones óptimas obtenidas con los conjuntos de datos dentro de estas categorías. Los *solvers* utilizados en esta etapa fueron Gurobi y CPLEX, que son de los *solvers* comerciales más conocidos y utilizados globalmente.

Se observó que ambos *solvers* resolvieron la Fase 1 en menos de un segundo para la gran mayoría de los casos de estas categorías. No obstante, en la configuración con 1500 estudiantes, ambos *solvers* tardaron entre 10 y 15 segundos para obtener soluciones óptimas para cada instancia de dicha configuración. No se observaron diferencias significativas entre los *solvers*.

Por otra parte, para resolver la Fase 2, se observó que en la mayoría de los casos ambos *solvers* no lograban obtener una solución óptima en un tiempo razonable. Al considerar configuraciones con 60 estudiantes o más, ninguno logró obtenerla en menos de 300 segundos; incluso se aumentó el tiempo límite para los casos con 300 estudiantes, que pasó a ser 600 segundos, y aún más para los casos con 1500 estudiantes, pasando a 90 minutos, pero no fue suficiente. A pesar de esto, para los 7 casos con 60 estudiantes, en promedio se notó que CPLEX obtenía mejores *mipgaps* que Gurobi (38,43 % contra 50,52 %). Para el único caso que se consideró de 1500 estudiantes, Gurobi no halló solución factible en 6 horas (no respetando el tiempo límite), mientras que CPLEX sí la halló, aunque era considerablemente peor que la que se halló en la Fase 1.

CPLEX también tuvo un rendimiento superior a Gurobi en la Fase 2 para los casos con 1 y 6 cursos. Con tan solo un curso, CPLEX logró hallar una solución óptima en 1.05 segundos en promedio, mientras que Gurobi demoró 4,38 segundos; más del cuádruple. Cuando se consideraban 6 turnos, CPLEX logró obtener *mipgaps* de 21,07 % en promedio, contra 39,29 % por parte de Gurobi. Sin embargo, para el único caso considerado con 150 cursos, con un tiempo límite de 90 minutos, CPLEX obtuvo un *mipgap* de 59,71 %, mientras que Gurobi obtuvo un *mipgap* de 48,81 %. Lo que es aún más interesante es que, en menos de 5 minutos, Gurobi ya había alcanzado el mejor resultado obtenido por CPLEX.

Tras considerar todo esto, se decidió seguir adelante con el *solver* de CPLEX, ya que en la mayoría de los casos se comportó similarmente a Gurobi, pero en algunos lo superó. Si bien Gurobi podría llegar a obtener mejores resultados que CPLEX con algunos casos más grandes (como lo hizo con 150 cursos), las configuraciones restantes a analizar consideran casos pequeños, de 30 estudiantes y 3 cursos. Es importante aclarar que, si bien se generaron los datos para las variaciones mixtas de E y C , estos no fueron resueltos, ya que servían principalmente para analizar a los *solvers* y así elegir cuál utilizar para los casos restantes, y el análisis previo ya fue suficiente para ello.

4.3.2. Resultados del análisis de sensibilidad

En esta sección se presenta el análisis de sensibilidad realizado sobre el resto de las categorías consideradas: la cantidad de turnos disponibles por curso (T_c), la capacidad de los turnos (q), la superposición (S), los parámetros de penalización de asignaciones a preferencias elevadas (k y n), la variación en la distribución de preferencias (m), y la variación mixta de capacidad y cantidad de turnos ($q(+)$, $t(-)$).

El procedimiento general seguido para cada una de estas categorías consiste en dos etapas; primero, tanto para la Fase 1 como para la Fase 2, se analizará cómo la variación de los valores del elemento variable influye sobre los indicadores clave, definidos en la Sección 4.2, como el valor de preferencias, la envidia, la cantidad de asignaciones a lista de espera, entre otros. En segundo lugar, se compararán los resultados obtenidos para los indicadores entre las dos fases, evaluando específicamente los indicadores referidos a la envidia. De esta forma, será posible analizar si los niveles de equidad alcanzados mejoran en la Fase 2, si se obtienen asignaciones con el mismo valor de preferencias, pero un menor valor de envidias.

Variaciones de la cantidad de turnos

En esta sección, se analiza cómo influye el aumento del número de turnos disponibles por curso (T_c) en los principales indicadores de los modelos. Se presenta una comparación entre diferentes configuraciones de cantidad de turnos (1, 3, 6, 12, 30, y 150 turnos), evaluando su impacto sobre los indicadores definidos. En total, se analizan 14 indicadores, cuyos resultados para la Fase 1 se resumen en la Tabla 4.1, donde se presentan los valores correspondientes a cada una de las configuraciones de T_c . Posteriormente, se presenta el análisis de la Fase 2, resumiendo los valores obtenidos para los indicadores en la Tabla 4.2, que muestra el impacto de minimizar la envidia en las asignaciones, comparando cómo estos resultados se diferencian de los obtenidos en la Fase 1.

En la **Fase 1**, al aumentar la cantidad de turnos disponibles para cada curso, se observó lo siguiente:

- **Valor total de preferencias**

El valor de preferencias disminuye a medida que aumenta T_c , pero esta reducción ocurre hasta $T_c = 12$. Por ejemplo, el valor de preferencias promedio pasa de 126,4 con 3 turnos a 94,8 con 12 turnos, como se muestra en la Tabla 4.1. Sin embargo, al seguir incrementando el número de turnos más allá de 12, el valor de preferencias se estabiliza y no se observan mejoras adicionales.

- **Preferencia mínima, máxima y media asignada**

Las preferencias mínima y máxima se mantienen casi sin cambios a medida que se incrementa el número de turnos, mostrando poca variación a lo largo

de los diferentes escenarios. En contraste, la preferencia media disminuye levemente a partir de $T_c = 3$, lo que refleja una leve mejora en la asignación general de los estudiantes hacia opciones más cercanas a sus preferencias ideales. Con $T_c = 1$, la preferencia media asignada es 1, lo que, en un principio, parecería mejor que cuando hay más turnos (por ejemplo, con 3 turnos la preferencia media asignada es 1,12, y cuando hay 12 turnos, la preferencia media asignada es 1,06), pero lo que sucede aquí es que al haber tan solo un turno por curso, solo se asignan a los estudiantes que habían expresado preferencia 1 por ese turno, y luego la gran mayoría de estudiantes quedan asignados a la lista de espera de ese curso, que no se considera para el cálculo de las preferencias medias asignadas.

■ **Curtosis de preferencias**

La medida de Curtosis de la distribución de preferencias es muy susceptible a los datos, pero parecería empeorar hasta $T_c = 6$, y luego mejorar para $T_c = 12$, y variar sin seguir ningún patrón aparente al continuar aumentando la cantidad de turnos. Por esta razón, no se puede concluir nada a partir de esta medida para las variaciones de T_c .

■ **Asignaciones a espera**

A medida que aumenta T_c , la cantidad total de asignaciones a espera disminuye, lo que indica que se logra distribuir a una mayor cantidad de estudiantes entre los turnos disponibles, lo cual tiene sentido, ya que al aumentar la cantidad de turnos por curso, se aumenta la capacidad global de cada curso (si bien la capacidad de cada turno se mantiene), por lo que es posible asignar a más estudiantes a cada curso. Esto se refleja también en la reducción de la cantidad máxima de asignaciones a espera para un estudiante, y en la cantidad promedio de asignaciones a espera por estudiante.

■ **Valor total de envidias y envidia media asignada**

El valor total de envidias disminuye al incrementar el número de turnos, a partir de $T_c = 3$. Con $T_c = 1$, hay muchos estudiantes envidiosos, pero hay pocos a quienes envidiar, ya que solo hay un turno y menos personas quedan asignadas. Con $T_c = 3$, menos personas experimentan envidia, pero el número de personas a quienes envidiar aumenta, dado que más estudiantes están siendo asignados. Estas diferencias entre dichas configuraciones se compensan y es por ello que el valor total de envidia promedio para $T_c = 1$ es similar al de $T_c = 3$. Al continuar incrementando los turnos, ya no hay asignaciones a lista de espera, lo que reduce considerablemente el valor total de envidias. Un comportamiento análogo se puede observar para la envidia media por estudiante, ya que la envidia media se puede calcular dividiendo el valor total de envidia entre la cantidad de estudiantes, que no varía en las configuraciones de las variaciones de T_c .

■ **Envidia mínima y máxima asignada**

La envidia mínima de un estudiante se mantiene en 0 para todas las configuraciones, ya que siempre fue posible asignar a al menos una persona en

sus turnos de preferencia 1 para los cursos considerados, logrando que no envidien a nadie. Por otra parte, se puede observar que la envidia máxima de un estudiante aumenta al pasar de $T_c = 1$ a $T_c = 3$. Esto se da porque tanto con $T_c = 1$ como con $T_c = 3$ quedará al menos un estudiante en lista de espera para algún curso, y un estudiante asignado a lista de espera de un curso envidiará a todos los que no fueron asignados a la lista de espera de ese curso, y para la configuración de $T_c = 1$ esta cantidad es 9, mientras que para el caso de $T_c = 3$, la cantidad es 27. Por esta razón, se observa que la envidia máxima de un estudiante prácticamente se triplica al pasar de $T_c = 1$ a $T_c = 3$. Para el resto de las configuraciones, ya es posible asignar a todos los estudiantes a algún turno de cada curso, por lo que las únicas envidias que existen son las de los estudiantes que fueron asignados a sus preferencias 2 y 3, mientras que otras personas fueron asignadas a las preferencias 1 de los estudiantes envidiosos. Esto reduce significativamente las envidias máximas de los estudiantes, por lo que para las configuraciones desde $T_c = 6$ en adelante las envidias máximas varían levemente, pero son bastante menores que en las configuraciones anteriores.

■ **Tiempos de resolución**

Si bien el tiempo que le toma al *solver* resolver estas configuraciones aumenta con la cantidad de turnos, los valores se mantienen dentro de las fracciones de un segundo.

Al comparar los indicadores de la Fase 1 con la **Fase 2**, no se observan diferencias sustanciales para una misma configuración (una cantidad de turnos dada). Las soluciones obtenidas en la Fase 2 solo difieren en la distribución de envidias, pero el valor total de envidia se mantiene igual que en la Fase 1 para las distintas configuraciones. En otras palabras, en las Fases 1 y 2 se hallan soluciones equivalentes, que son óptimas para ambas fases. Las distintas distribuciones de envidias halladas implican diferencias no relevantes en las envidias máximas por estudiante y en el valor de Curtosis comparando entre las 2 fases.

Por otra parte, los tiempos de resolución del *solver* en la Fase 2 fueron mucho mayores, y aumentaron hasta $T_c = 6$ y luego disminuyeron. En la configuración de $T_c = 6$, el *solver* demoró en promedio 185,67 segundos en resolver los 5 casos de esta configuración (no logrando verificar la optimalidad de las soluciones encontradas en 2 de los casos en el tiempo límite otorgado de 300 segundos).

	$T_c = 1$	$T_c = 3$	$T_c = 6$	$T_c = 12$	$T_c = 30$	$T_c = 150$
Valor Preferencias	153,0	126,4	100	94,8	95,0	95,2
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	1,0	2,0	2,0	2,0	1,8	2,0
Preferencia media	1,0	1,12	1,11	1,05	1,06	1,06
Curtosis preferencias	∞	5,71	4,96	22,13	11,92	18,22
Cant. asig. a espera	63,0	9,0	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	3,0	1,6	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	2,1	0,3	0,0	0,0	0,0	0,0
Valor envidias	567,0	570,6	90,0	43,2	45,0	46,8
Envidia mín. de un estudiante	0,0	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	27,0	90,0	16,2	12,6	10,8	10,8
Envidia media de un estudiante	18,9	19,0	3,0	1,4	1,5	1,6
Curtosis envidias	-1,24	0,65	1,30	6,57	3,63	5,13
Tiempo de resolución (s)	0,00	0,01	0,02	0,02	0,03	0,08

Tabla 4.1: Valores de los indicadores para el modelo de Fase 1 con variación de T_c

	$T_c = 1$	$T_c = 3$	$T_c = 6$	$T_c = 12$	$T_c = 30$	$T_c = 150$
Valor Preferencias	153,0	126,4	100	94,8	95,0	95,2
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	1,0	2,0	2,0	2,0	1,8	2,0
Preferencia media	1,0	1,12	1,11	1,05	1,06	1,06
Curtosis preferencias	∞	5,71	4,96	22,13	11,92	18,22
Cant. asig. a espera	63,0	9,0	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	3,0	2,4	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	2,1	0,3	0,0	0,0	0,0	0,0
Valor envidias	567,0	570,6	90,0	43,2	45,0	46,8
Envidia mín. de un estudiante	0,0	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	27,0	122,4	14,4	10,8	14,4	10,8
Envidia media de un estudiante	18,9	19,0	3,0	1,4	1,5	1,6
Curtosis envidias	-1,24	3,33	0,51	5,48	7,29	4,10
Tiempo de resolución (s)	1,58	15,14	185,67	86,32	62,98	61,51

Tabla 4.2: Valores de los indicadores para el modelo de Fase 2 con variación de T_c

Se presentan dos gráficas que ilustran cómo varían el valor de preferencias y el valor de envidias conforme se incrementa T_c . En la Figura 4.2, se observa que el valor de preferencias disminuye hasta $T_c = 12$, estabilizándose a partir de ese valor. Esto indica que, aunque aumentar el número de turnos mejora la asignación, después de un cierto punto, el impacto es marginal. La estabilización del valor de preferencias sugiere que el aumento en T_c más allá de 12 no proporciona mejoras significativas en la asignación considerando las preferencias. Por otro lado, la Figura 4.3 muestra cómo el valor de envidia disminuye a medida que se incrementa el número de turnos. Al igual que con el valor de preferencias, la envidia disminuye considerablemente hasta $T_c = 12$, donde las diferencias en los niveles de envidia se vuelven menos pronunciadas. Esto refleja que un mayor número de turnos reduce las disparidades en las asignaciones, haciendo que los estudiantes envidien menos las asignaciones de otros, lo que favorece una distribución más equitativa.

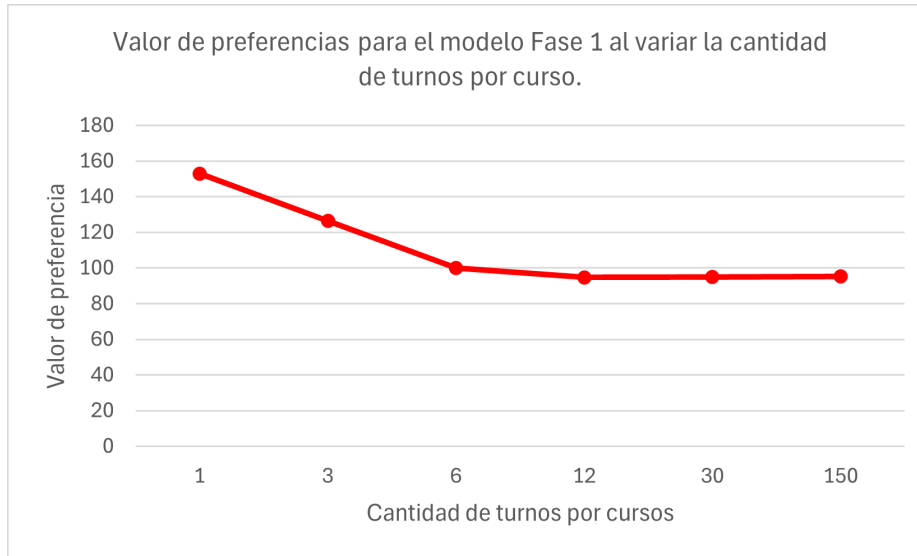


Figura 4.2: Valor de preferencias para el modelo Fase 1 al variar la cantidad de turnos por curso.

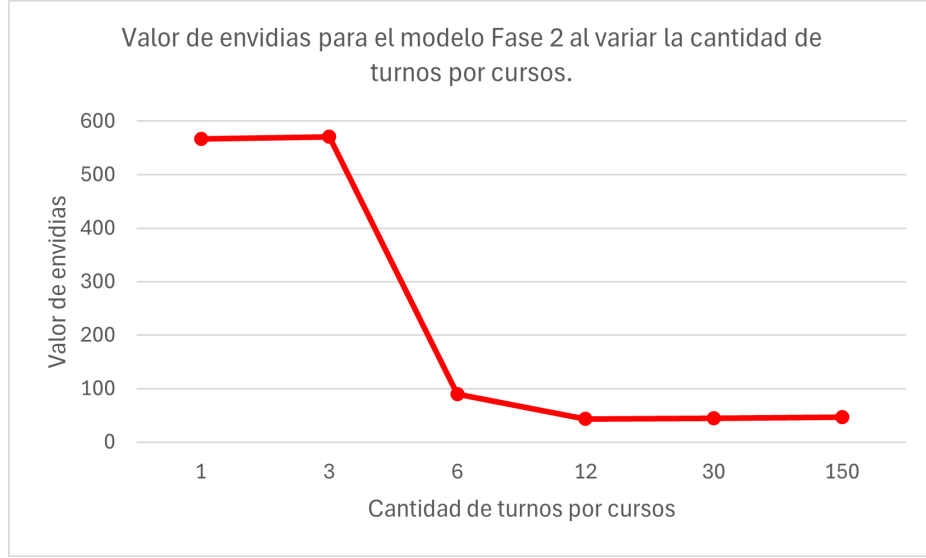


Figura 4.3: Valor de envidias para el modelo Fase 2 al variar la cantidad de turnos por curso.

En conclusión, la variación en la cantidad de turnos disponibles por curso tiene un impacto significativo tanto en el valor de preferencias como en el valor de envidias, pero solo hasta cierto punto. Para estos datos en particular, a partir de $T_c = 12$, no se observan mejoras sustanciales en ninguno de los dos indicadores. Cabe destacar que los datos fueron generados de forma automatizada a partir de un generador de datos, como se vio en la Sección 4.1, por lo que no se pueden derivar las conclusiones obtenidas hacia casos reales en donde se aumente la cantidad de turnos disponibles de cursos. Si bien se espera que al aumentar este valor, tanto el valor de preferencias como el de envidias disminuyan, es probable que lo hagan de forma distinta. De hecho, la envidia podría tender a un valor global de 0, y el valor de preferencias podría ser menor que el observado en las configuraciones de T_c analizadas.

Variaciones de la capacidad de los turnos

En esta sección, se analiza cómo la variación en la capacidad de los turnos (q) influye sobre los principales indicadores de los modelos. La capacidad de los turnos determina cuántos estudiantes pueden ser asignados a cada turno, lo cual afecta directamente tanto a los valores de preferencia de las asignaciones de los estudiantes como a los niveles de envidia y a la cantidad de asignación a espera. Los resultados para cada configuración al resolver la Fase 1 se resumen en la Tabla 4.3, donde se presentan los valores correspondientes a cada nivel de q . Posteriormente, se analiza el comportamiento de la Fase 2, resumiendo los valores de los indicadores clave en la Tabla 4.4, y comparando los resultados de esta fase frente a la Fase 1.

En la **Fase 1**, al aumentar la capacidad de turnos disponibles para cada curso, se observó lo siguiente:

- **Valor total de preferencias**

El valor de preferencias disminuye conforme aumenta q . Para $q = 3$, el valor total promedio entre las 5 instancias analizadas es 279,0. Este valor se reduce a 126,4 en $q = 9$ y finalmente se estabiliza en 90 para $q = 18$ y valores superiores. Esto tiene sentido, ya que al aumentar la capacidad de los turnos es posible asignar a más estudiantes a sus turnos más preferidos; a partir de cierto punto, se pueden asignar a todos a sus turnos preferidos, alcanzando el valor de preferencias mínimo posible, que para estas configuraciones es 90.

- **Preferencia mínima, máxima y media asignada**

La preferencia mínima se mantiene constante en 1 en todas las configuraciones, lo que indica que siempre hay estudiantes que son asignados a su primera opción. La preferencia máxima disminuye de 2 en $q = 9$ a 1 a partir de $q = 18$, lo que significa que, con una mayor capacidad, todos los estudiantes son asignados a su primera preferencia. De manera similar, la preferencia media sigue este patrón, tomando un valor de 1,12 en $q = 9$, y estabilizándose en 1,0 a partir de $q = 18$, lo que refleja una asignación más equitativa a partir de una mayor capacidad. Cuando la capacidad de los turnos es muy reducida, los pocos estudiantes que quedan asignados en los turnos disponibles son algunos de los que marcaron preferencia 1 por dichos turnos, por lo que en la configuración de $q = 3$, tanto la preferencia máxima asignada, como la media, son 1 (similar a lo que ocurría en la configuración de $T_c = 1$).

■ **Asignaciones a espera**

A medida que aumenta la capacidad de los turnos, tanto la cantidad total de asignaciones a espera como el promedio de asignaciones a espera por estudiante disminuyen. Para $q = 3$, hay 63 estudiantes en espera, con un promedio de 2,1 asignaciones a espera por estudiante. Sin embargo, este número cae rápidamente a 9 estudiantes en $q = 9$ y se reduce a 0 a partir de $q = 18$. Un comportamiento similar se puede apreciar para los indicadores de la cantidad máxima y mínima de asignaciones a espera para un estudiante. Todo esto indica que con una capacidad suficiente, se logra asignar a todos los estudiantes sin necesidad de listas de espera.

■ **Indicadores de envidia**

El valor total de envidias disminuye a medida que aumenta la capacidad de los turnos, reduciéndose a 0 a partir de $q = 18$. La envidia mínima de un estudiante se mantiene en 0 para todas las configuraciones, excepto en $q = 3$. Esto indica que en al menos una instancia de esa configuración, no hubo ningún estudiante libre de envidia, lo cual es concordante con que, para la configuración en cuestión, la cantidad mínima de asignaciones a espera para un estudiante fue en promedio 0,2. Esto quiere decir que en una de las 5 instancias analizadas de esta configuración, todos los estudiantes fueron asignados a la lista de espera de al menos un curso, envidiando a los que quedaron asignados a los turnos. La envidia máxima es más alta en $q = 9$, ya que en este caso hay pocos estudiantes en espera que envidian a los muchos que han sido asignados, similar a lo que ocurre en las configuraciones de las variaciones de T_c . La envidia máxima toma el valor de 0 a partir de $q = 18$. De manera similar, la envidia media sigue el mismo patrón, disminuyendo de 19,02 en $q = 9$ a 0 en los casos con mayor capacidad.

	$q = 3$	$q = 9$	$q = 18$	$q = 36$	$q = 90$	$q = 450$
Valor Preferencias	279,0	126,4	90,0	90,0	90,0	90,0
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	1,0	2,0	1,0	1,0	1,0	1,0
Preferencia media	1,0	1,12	1,0	1,0	1,0	1,0
Curtosis preferencias	∞	5,71	∞	∞	∞	∞
Cant. asig. a espera	63,0	9,0	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,2	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	3,0	1,6	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	2,1	0,3	0,0	0,0	0,0	0,0
Valor envidias	1134,0	570,6	0,0	0,0	0,0	0,0
Envidia mín. de un estudiante	3,6	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	54,0	90,0	0,0	0,0	0,0	0,0
Envidia media de un estudiante	37,8	19,02	0,0	0,0	0,0	0,0
Curtosis envidias	-0,46	0,65	∞	∞	∞	∞
Tiempo de resolución (s)	0,01	0,01	0,02	0,00	0,00	0,00

Tabla 4.3: Valores de los indicadores para el modelo de Fase 1 con variación de q

	$q = 3$	$q = 9$	$q = 18$	$q = 36$	$q = 90$	$q = 450$
Valor Preferencias	279,0	126,4	90,0	90,0	90,0	90,0
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	1,0	2,0	1,0	1,0	1,0	1,0
Preferencia media	1,0	1,12	1,0	1,0	1,0	1,0
Curtosis preferencias	∞	5,71	∞	∞	∞	∞
Cant. asig. a espera	63,0	9,0	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,6	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	3,0	2,4	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	2,1	0,3	0,0	0,0	0,0	0,0
Valor envidias	1134,0	570,6	0,0	0,0	0,0	0,0
Envidia mín. de un estudiante	10,8	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	54,0	122,4	0,0	0,0	0,0	0,0
Envidia media de un estudiante	37,8	19,02	0,0	0,0	0,0	0,0
Curtosis envidias	-0,84	3,33	∞	∞	∞	∞
Tiempo de resolución (s)	128,13	15,14	0,00	0,01	0,01	0,01

Tabla 4.4: Valores de los indicadores para el modelo de Fase 2 con variación de q

Al comparar los resultados obtenidos en la Fase 1 frente a la **Fase 2**, no se observan diferencias significativas entre los dos enfoques, más que en los tiempos de resolución. Al igual que en el análisis de las variaciones de T_c , el valor de preferencias y de envidias se mantuvo entre ambas fases, manteniendo la misma distribución de preferencias, pero una distribución de envidias distinta, lo que genera ciertas diferencias en algunos indicadores de envidia. Las soluciones óptimas obtenidas en la Fase 1 también serían óptimas en la Fase 2, y viceversa. Esto sugiere que para esta generación de datos, solamente la variación de la capacidad no es suficiente para observar mejoras en los indicadores de envidia al utilizar la Fase 2.

En la Figura 4.4 se muestra cómo varía el valor de las preferencias al aumentar la capacidad de los turnos. Como se puede observar, el valor de preferencias disminuye a medida que q incrementa, estabilizándose a partir de cierto valor, lo que indica que una mayor capacidad de los turnos permite satisfacer mejor las preferencias de los estudiantes.

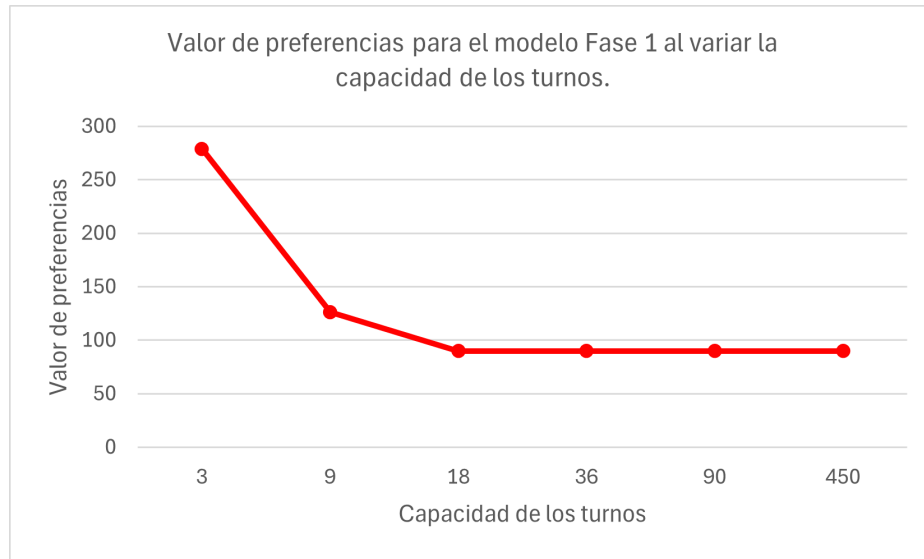


Figura 4.4: Valor de preferencias para el modelo Fase 1 al variar la capacidad de los turnos

Variaciones de la cantidad de turnos superpuestos

En esta sección, se analiza cómo la variación en la cantidad de superposiciones (S) influye en los principales indicadores de los modelos. La superposición de un par de turnos implica que ningún estudiante pueda ser asignado a ambos turnos, lo cual puede afectar la asignación de estudiantes. Se presenta una comparación entre diferentes configuraciones de cantidad de pares de superposiciones (5, 10, 15, 20 y 30 superposiciones), evaluando su impacto sobre indicadores clave como el valor de preferencias, el valor de envidias y la cantidad de asignaciones a lista de espera.

Los resultados para cada configuración se resumen en la Tabla 4.5 para el caso de la Fase 1 y en la Tabla 4.6 para el caso de la Fase 2, donde se detallan los valores de cada indicador, correspondientes a cada nivel de S . El análisis sigue el mismo procedimiento que en la sección anterior: primero, se evalúa el impacto de la variación de S sobre los indicadores clave del modelo de la Fase 1, y luego se comparan los resultados obtenidos con los de la Fase 2, prestando especial atención a los indicadores relacionados con la envidia y la equidad.

En la **Fase 1**, al aumentar la cantidad de pares de turnos superpuestos, se observó lo siguiente:

- **Valor total de preferencias**

El valor de preferencias aumenta conforme se incrementa el número de superposiciones. Este aumento es esperable, ya que al permitir más superposiciones, se introducen más restricciones, lo que implica que más personas queden en lista de espera. Además, las asignaciones se vuelven más difíciles de gestionar debido a las restricciones adicionales, lo que en algunos casos incrementa la cantidad de asignaciones a preferencias mayores (por ejemplo, al pasar de $S = 0$ a $S = 15$), en donde se nota que la preferencia media asignada por estudiante aumenta.

- **Preferencia mínima, máxima y media asignada**

Las preferencias mínimas se mantienen en 1 para todos los casos, lo que indica que siempre hay estudiantes que son asignados a su primera opción. Por otro lado, a partir de $S = 5$, las preferencias máximas disminuyen hasta 1, lo cual es esperable, ya que al haber menos posibilidades de asignación, el modelo tiende a asignar únicamente a las preferencias más bajas. De manera similar, la preferencia media también disminuye. Como ya se mencionó anteriormente, al pasar de $S = 0$ a $S = 15$, la preferencia media aumenta (al igual que la preferencia máxima asignada), esto probablemente se da porque aún es posible asignar a la mayoría de los estudiantes (ya que la cantidad de pares de turnos superpuestos es baja), pero estas superposiciones dificultan las asignaciones a preferencias 1. Por ejemplo, considerando un caso en el que dos turnos de dos cursos distintos se superpongan, y que un estudiante haya marcado preferencia 1 para dichos turnos, sería imposible asignarlo a su preferencia 1 para ambos cursos, que

en el caso de $S = 0$ podría suceder.

■ **Asignaciones a espera**

Todos los indicadores considerados referentes a asignaciones a espera (cantidad total de asignaciones a espera, cantidad mínima y máxima de asignaciones a espera para un estudiante, y cantidad promedio de asignaciones a espera por estudiante) aumentan a medida que se incrementa el número de superposiciones. Este aumento es esperado, ya que al permitir más superposiciones, también se incrementa la complejidad de las asignaciones, lo que genera más estudiantes en lista de espera y mayores dificultades para completar las asignaciones de manera directa.

■ **Valor total de envidias, y envidia mínima y media asignada**

Tanto la envidia global, como la envidia mínima de un estudiante, y la envidia media de los estudiantes, aumentan hasta $S = 15$, y luego comienzan a descender. Una posible interpretación para este comportamiento es que la envidia global es mayor cuando la cantidad de asignaciones a espera, y la cantidad de asignaciones a turnos es similar. En estas configuraciones, al haber 30 estudiantes y 3 cursos, y todos los estudiantes quieren ser asignados a todos los cursos, la cantidad total de asignaciones será $30 \times 3 = 90$, y se observa que en $S = 15$, que hay en promedio 43,8 asignaciones a espera (por lo que hay $90 - 43,8 = 46,2$ asignaciones a turnos), la envidia es mayor. Esto es porque los estudiantes asignados a lista de espera de un curso no se envidian entre ellos (por ese curso), pero sí envidian a los asignados a los turnos de ese curso, por lo que el mayor número de pares envidiosos se daría cuando la cantidad de asignaciones a espera es la misma que la de asignaciones a turnos. Este resultado se deriva de la desigualdad de la media aritmética y geométrica, como se muestra en el Apéndice A.6.

- **Envidia máxima asignada** La envidia máxima de un estudiante se comporta distinto al resto de los indicadores de envidia considerados. Este indicador disminuye al aumentar la cantidad de pares de turnos superpuestos, en vez de aumentar hasta cierto punto y luego disminuir, como en el resto de los indicadores de envidia. Esto va de la mano con que la cantidad de asignaciones a espera va aumentando, por lo que en las primeras configuraciones, hay pocos estudiantes que envidian a muchos, por lo que sus envidias serán mayores, si bien el impacto en la envidia global no será tan drástico.

Al analizar los resultados de la **Fase 2**, se observan diferencias importantes en varios indicadores clave. Las principales diferencias entre ambos enfoques fueron:

■ **Valor total de envidias y envidia media asignada**

El valor de envidias, y la envidia media de un estudiante muestran diferencias significativas entre las dos fases, al comparar entre las mismas configuraciones. En la Fase 2, este valor es considerablemente más bajo que en la Fase 1, especialmente para valores altos de S . Por ejemplo, para

$S = 30$, el valor total de envidias promediado entre las 5 instancias analizadas es de 391,4 en la Fase 2 y la envidia media es de 13,05, mientras que en la Fase 1, el valor total de envidias es de 843,6, y la envidia media de 28,12. Esta diferencia evidencia el impacto positivo de la Fase 2 sobre las soluciones obtenidas.

- **Envidia mínima y máxima asignada**

Se observan mejores valores de envidia mínima en la Fase 2, permaneciendo en valores cercanos a 5 para los casos de $S = 10$ en adelante, mientras que en la Fase 1, la envidia mínima variaba considerablemente para los casos de $S = 10$ en adelante, y siempre tomaba valores mayores que en la Fase 2. La envidia máxima, sin embargo, no varía considerablemente entre las 2 fases, pero generalmente es menor en la Fase 2.

- **Curtosis de envidias**

Si bien se esperaba que obtener soluciones con menores envidias implicara mejores valores de Curtosis para las distribuciones de envidia (valores más elevados), los resultados obtenidos no se comportan de esta manera. En algunos casos se observan mejores valores, pero en otros no. Por ejemplo, con $S = 10$, la envidia global se reduce en la Fase 2, pero la Curtosis de la distribución de envidias es menor que en la Fase 1. Por esta razón, no se puede inferir que una solución es mejor que otra observando solamente este indicador. Este indicador toma importancia cuando el valor total de preferencias y de envidias de 2 soluciones son iguales, pero difieren los valores de Curtosis de las distribuciones de envidias de ambas soluciones; ahí sí se podría argumentar que la mejor solución es la que tiene un mayor valor de Curtosis, ya que esto implica que las envidias están más concentradas en torno al valor promedio y que hay menos valores extremos, que puede interpretarse como una mayor equidad en la asignación.

En resumen, aunque ambos modelos siguen un comportamiento similar en cuanto a la cantidad de asignaciones a espera y el valor de preferencias, el modelo de minimización de envidia logra una mejor distribución de las preferencias y una reducción significativa en los niveles de envidia entre los estudiantes, logrando reducir el valor total de envidias. Estos resultados destacan la ventaja de utilizar el enfoque de minimización de envidia para conseguir asignaciones más equitativas, especialmente a medida que se permite un mayor número de superposiciones.

	$S = 0$	$S = 5$	$S = 10$	$S = 15$	$S = 20$	$S = 30$
Valor Preferencias	126,4	142,4	200,6	227,0	264,8	270,0
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	2,0	3,0	2,4	1,8	1,2	1,0
Preferencia media	1,12	1,17	1,14	1,12	1,01	1,0
Curtosis preferencias	5,71	5,99	4,06	3,16	39,0	∞
Cant. asig. a espera	9,0	13,0	34,2	43,8	58,2	60,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,6	1,0	1,8	2,0
Cant. máx de asignaciones a espera para un estudiante	1,6	1,4	1,8	2,0	2,0	2,0
Cant. promedio de asignaciones a espera para un estudiante	0,3	0,43	1,14	1,46	1,94	2,0
Valor envidias	570,6	785,2	1196,4	1308,4	1120,8	843,6
Envidia mín. de un estudiante	0,0	0,0	13,0	22,0	21,2	13,6
Envidia máx. de un estudiante	90,0	81,8	77,2	71,0	56,8	57,4
Envidia media de un estudiante	19,02	26,07	39,88	43,63	37,36	28,12
Curtosis envidia	0,65	-0,85	-0,49	-1,33	-0,59	-0,54
Tiempo de resolución (s)	0,01	0,01	0,05	0,02	0,02	0,02

Tabla 4.5: Valores de los indicadores para el modelo de Fase 1 con variación de S

	$S = 0$	$S = 5$	$S = 10$	$S = 15$	$S = 20$	$S = 30$
Valor Preferencias	126,4	142,4	200,6	227,0	264,8	270,0
Preferencia mínima	1,0	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	2,0	2,4	2,6	1,8	1,2	1,0
Preferencia media	1,12	1,19	1,14	1,12	1,01	1,0
Curtosis preferencias	5,71	3,49	8,29	3,16	39,0	∞
Cant. asig. a espera	9,0	12,6	34,2	43,8	58,2	60,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,6	1,0	1,8	2,0
Cant. máx de asignaciones a espera para un estudiante	2,4	1,4	1,8	2,0	2,0	2,0
Cant. promedio de asignaciones a espera para un estudiante	0,3	0,42	1,14	1,46	1,94	2,0
Valor envidias	570,6	755,4	913,8	915,0	565,2	391,4
Envidia mín. de un estudiante	0,0	0,0	4,8	5,6	5,6	5,0
Envidia máx. de un estudiante	122,4	78,0	67,0	60,8	52,4	55,4
Envidia media de un estudiante	19,02	25,18	30,46	30,5	18,84	13,05
Curtosis envidia	3,33	-0,82	-1,4	-0,94	0,34	3,56
Tiempo de resolución (s)	15,14	58,64	12,64	64,33	63,05	39,87

Tabla 4.6: Valores de los indicadores para el modelo de Fase 2 con variación de S

Para finalizar el análisis de la variación en el número de superposiciones (S), se presentan dos gráficas que ilustran los resultados más relevantes. En la Figura 4.5 se muestra cómo el valor de preferencias aumenta significativamente a medida que aumenta S , lo que refleja la mayor dificultad para satisfacer las preferencias cuando se permiten más superposiciones. En la Figura 4.6 se observa el comportamiento del valor total de envidias en función de S , tanto para la Fase 1 como para la Fase 2. En esta gráfica, se destaca que el modelo de minimización de envidia consigue valores considerablemente más bajos del valor total de envidias, especialmente para valores altos de S , mostrando que este enfoque logra una mejor equidad en las asignaciones.

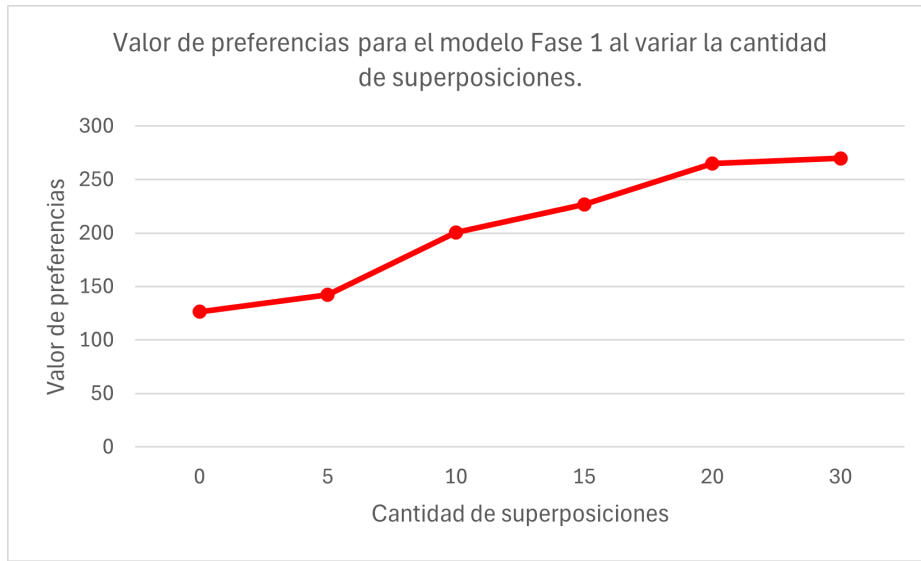


Figura 4.5: Valor de preferencias para el modelo Fase 1 al variar las superposiciones

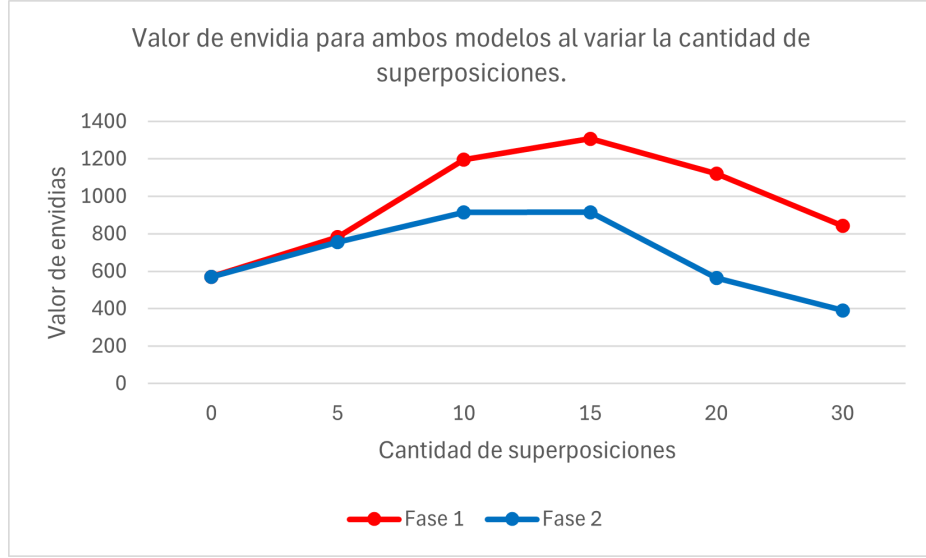


Figura 4.6: Valor de envidias para los modelos Fase 1 y Fase 2 al variar la cantidad de superposiciones

Variaciones de los penalizadores de preferencias altas

Este análisis tiene como propósito analizar cómo se modifican las soluciones óptimas al modificar los parámetros k y m , penalizando preferencias altas. Para los casos considerados, se buscaba penalizar las asignaciones a preferencias 3, pero, al notar que en el caso base, que no penaliza asignaciones a preferencias 3, en ninguna de las instancias se asignaron preferencias 3, penalizar estas asignaciones no afectaría en nada en las configuraciones posteriores, ya que inicialmente no se están asignando estas preferencias. Luego de obtener los resultados de los indicadores de las primeras dos configuraciones ($k = 10$, $n = 1$ y $k = 3$, $n = 1, 5$) se concluyó que el análisis de estas variaciones no brindaría ningún aporte interesante. En la Sección 5, se detallarán posibles soluciones para este problema.

Variaciones del generador de preferencias

En esta sección, se analiza cómo influye la variación del parámetro m , utilizado para generar las preferencias en el generador de datos elaborado, en los principales indicadores de los modelos. El parámetro m define la probabilidad de que un turno específico sea preferido por un estudiante frente al siguiente turno, y su variación afecta directamente la distribución de preferencias entre los estudiantes. Se presenta una comparación entre diferentes configuraciones de m (1, 1.5, 2, 4, 10), evaluando su impacto sobre los indicadores clave, tales

como el valor de preferencias, la preferencia mínima y máxima asignada y la cantidad de asignaciones. Los resultados se resumen en la Tabla 4.7, donde se presentan los valores obtenidos de dichos indicadores para cada una de las configuraciones de m . Posteriormente, se presenta el análisis de la Fase 2. En este caso, los valores de los indicadores clave se resumen en la Tabla 4.8.

En la **Fase 1**, al aumentar el valor de m , se observó lo siguiente:

- **Valor total de preferencias**

El valor de preferencias aumenta conforme el valor de m se incrementa. Esto tiene sentido, ya que cada vez es más probable que los estudiantes elijan el primer turno con preferencia 1, el segundo con preferencia 2, y el tercero con preferencia 3. No obstante, no hay suficiente capacidad para que todos los estudiantes sean asignados a sus primeros turnos preferidos. Como resultado, será cada vez más necesario asignar preferencias 2 y 3, lo que incrementa el valor total de preferencias.

- **Preferencia mínima, máxima y media asignada**

La preferencia mínima se mantiene en 1 para todos los valores de m , lo que indica que siempre hay al menos un estudiante asignado a su primera opción. La preferencia máxima aumenta con m , reflejando que más estudiantes son asignados a opciones menos preferidas. La preferencia media también sigue esta tendencia, lo que sugiere una menor satisfacción general a medida que m crece, debido a la asignación de preferencias más altas.

- **Curtosis de preferencias**

El valor de Curtosis para las distribuciones de preferencias asignadas disminuye al aumentar m , lo que implica que cada vez se asignan menos preferencias 1, y más preferencias 2 y 3.

- **Asignaciones a espera**

Los indicadores relacionados con las asignaciones a espera se mantienen constantes, o experimentan variaciones leves, ya que la cantidad de estudiantes, cursos, turnos, y capacidad se mantienen constantes en estas configuraciones.

- **Indicadores de envidia**

Tanto el valor de envidias como la envidia media asignada aumentan con m , lo cual es razonable ya que, si bien la cantidad total de asignaciones a espera se mantiene constante, la cantidad de asignaciones a preferencias 2 o 3 aumenta junto con m , lo que genera que los estudiantes asignados a estas preferencias mayores experimenten envidia hacia los estudiantes que quedan asignados en sus turnos preferidos. Por otra parte, la envidia mínima se mantiene en 0 para todas las configuraciones, y la envidia máxima varía entre 90 y 117, pero no se logra deducir un comportamiento claro con el aumento de m .

Al comparar estos resultados con los de la **Fase 2**, no se observan mejoras en el valor total de envidias, excepto para la configuración de $m = 4$. Aquí, se logró reducir levemente la envidia global en dos de las cinco instancias analizadas. En estas instancias, se observa que la distribución de preferencias se modificó con respecto a la Fase 1, teniendo menos asignaciones a preferencias 1 y 3, y más asignaciones a preferencias 2, manteniendo el valor total de preferencias. En una de las instancias, la envidia global pasó de 864 a 846, y en la otra de 846 a 819, por lo que las diferencias no son tan importantes (menores al 5 %).

Por otra parte, como ha sucedido en los análisis previos, el tiempo de resolución en la Fase 2 es mucho mayor que en la Fase 1, para todas las configuraciones de m . En la Fase 1, los tiempos de resolución varían entre 0,01 y 0,03 segundos, mientras que en la Fase 2, varían entre 15,14 y 300 segundos (el tiempo límite considerado para estas pruebas), empeorando desde $m = 1,5$ en adelante. Con $m = 10$, no se logró hallar ninguna solución óptima en menos del tiempo límite, para ninguna de las 5 instancias analizadas. Los valores óptimos obtenidos al variar m para las Fases 1 y 2 se muestran en las Figuras 4.7 y 4.8, respectivamente.

	$m = 1$	$m = 1,5$	$m = 2$	$m = 4$	$m = 10$
Valor Preferencias	121,6	126,4	138,8	157,0	183,8
Preferencia mínima	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	2,0	2,0	2,4	3,0	3,0
Preferencia media	1,06	1,12	1,27	1,49	1,82
Curtosis preferencias	25,43	5,71	-0,05	-0,26	-1,42
Cant. asig. a espera	9,0	9,0	9,0	9,0	9,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	2,0	1,6	1,8	2,0	1,6
Cant. promedio de asignaciones a espera para un estudiante	0,3	0,3	0,3	0,3	0,3
Valor envidias	527,4	570,6	685,8	898,2	1276,2
Envidia mín. de un estudiante	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	108,0	90,0	100,8	117,0	117,0
Envidia media de un estudiante	17,58	19,02	22,86	29,94	42,54
Curtosis envidia	2,25	0,65	0,96	1,52	-0,65
Tiempo de resolución (s)	0,02	0,01	0,01	0,02	0,03

Tabla 4.7: Valores de los indicadores para el modelo de Fase 1 con variación de m

	$m = 1$	$m = 1,5$	$m = 2$	$m = 4$	$m = 10$
Valor Preferencias	121,6	126,4	138,8	157,0	183,8
Preferencia mínima	1,0	1,0	1,0	1,0	1,0
Preferencia máxima	2,0	2,0	2,4	3,0	3,0
Preferencia media	1,06	1,12	1,27	1,49	1,82
Curtosis preferencias	25,43	5,71	-0,05	-0,54	-1,42
Cant. asig. a espera	9,0	9,0	9,0	9,0	9,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	1,8	2,4	2,0	2,0	2,2
Cant. promedio de asignaciones a espera para un estudiante	0,3	0,3	0,3	0,3	0,3
Valor envidias	527,4	570,6	685,8	898,2	1276,2
Envidia mín. de un estudiante	0,0	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	97,2	122,4	113,4	126,0	136,8
Envidia media de un estudiante	17,58	19,02	22,86	29,94	42,38
Curtosis envidia	1,46	3,33	2,62	2,6	0,89
Tiempo de resolución (s)	25,84	15,14	138,07	180,41	300,0

Tabla 4.8: Valores de los indicadores para el modelo de Fase 2 con variación de m

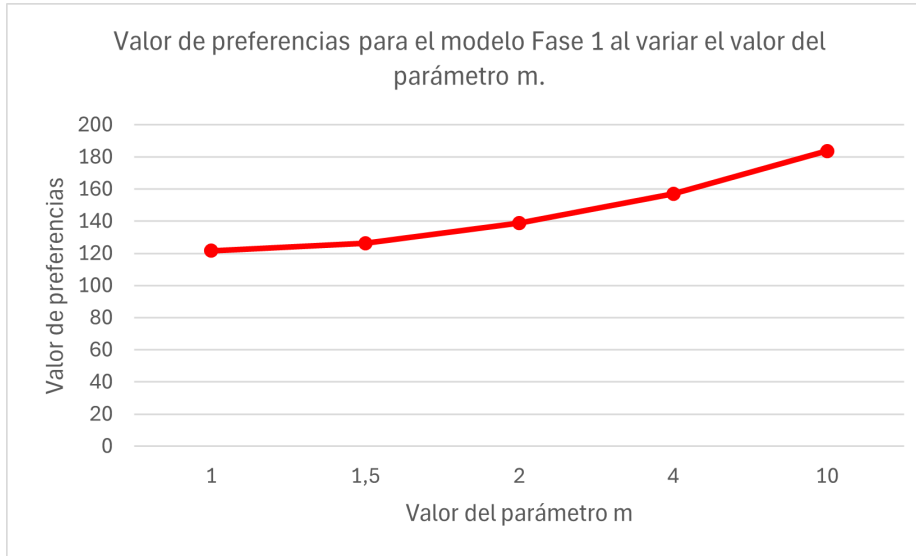


Figura 4.7: Valor de preferencias para el modelo Fase 1 al variar el valor del parámetro m

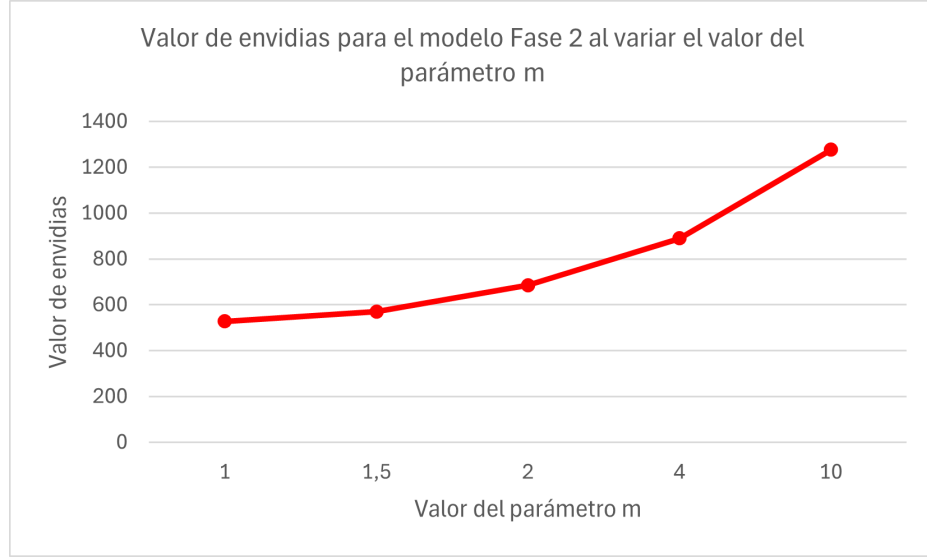


Figura 4.8: Valor de envidias para el modelo Fase 2 al variar el valor del parámetro m

Variación mixta de capacidad y cantidad de turnos

Por último, se analizaron los casos en donde la capacidad de los turnos aumenta mientras la cantidad de turnos disminuye, utilizando los valores de q propuestos en las variaciones de q , y los de T_c propuestos en las variaciones de T_c . Al igual que en el resto de los casos, primero se analizan los indicadores obtenidos en la Fase 1, resumiendo los valores obtenidos para dichos indicadores en la Tabla 4.9. Luego, se comparan estos valores con los obtenidos en la Fase 2, cuyo resumen se encuentra en la Tabla 4.10. Cabe destacar que no se analizaron todas las configuraciones generadas para esta categoría, ya que al notar que a partir de cierto punto los resultados serían iguales, seguir con el análisis no aportaría nada adicional.

En la **Fase 1**, al aumentar la capacidad de los turnos, mientras se reduce la cantidad de turnos, se observó lo siguiente:

- **Indicadores de preferencias**

Tanto el valor total de preferencias, como la preferencia máxima y media asignadas disminuyen, llegando a valores inmejorables. El valor de preferencias comienza en 178,4 en $q = 3$, $T_c = 150$, y se reduce hasta llegar al valor óptimo de 90 (asignando a todos los estudiantes a sus preferencias 1) en $q = 18$, $T_c = 12$, manteniéndose para futuras configuraciones, lo que se puede observar en la Figura 4.9. Similarmente, la preferencia máxima asignada pasa de ser 5,6 en $q = 3$, $T_c = 150$, a tomar el valor de 1

en $q = 18$, $T_c = 12$, mientras que la preferencia media pasa de 1,98 a 1 en las configuraciones mencionadas. Por otro lado, la preferencia mínima asignada se mantiene en 1 para todas las configuraciones.

■ **Asignaciones a espera**

En las configuraciones consideradas, siempre hay suficiente capacidad para poder asignar a todos los estudiantes a todos los cursos, por lo que los valores de los indicadores de las asignaciones a espera toman todos el valor de 0.

■ **Indicadores de envidia**

A los indicadores de envidia les ocurre algo similar que a los de preferencias; a partir de $q = 18$, $T_c = 12$, comienzan a tomar valores inmejorables (todos comienzan a valer 0). El valor total de envidias comienza en 473,4 mientras que la envidia media por estudiante en 15,78 y la envidia máxima de un estudiante en 61,2. La envidia mínima de un estudiante se mantiene en 0 para todas las configuraciones.

■ **Tiempos de resolución**

Los tiempos de resolución del *solver* también disminuyen al avanzar en las configuraciones. Al *solver* le toma en promedio 0,08 segundos resolver las instancias consideradas en la configuración de $q = 3$, $T_c = 150$, mientras que en $q = 36$, $T_c = 6$, los tiempos de resolución pasan a ser prácticamente 0 segundos.

En la **Fase 2**, los valores obtenidos para la mayoría de los indicadores son similares a los de la Fase 1, excepto los tiempos de resolución de las primeras 2 configuraciones. Los tiempos de resolución, si bien son mayores en la Fase 2 (como ya ha pasado en los demás análisis), tienen un comportamiento distinto frente a la variación mixta de q y T_c que en la Fase 1. Como ya fue mencionado, en la Fase 1 el tiempo de resolución disminuía al aumentar q y disminuir T_c , pero en la Fase 2, se puede observar que en $q = 3$, $T_c = 150$, el tiempo de resolución del *solver* es en promedio 3,2 segundos, y al pasar a $q = 9$, $T_c = 30$, este valor aumenta a 93,45 segundos. En el resto de las configuraciones analizadas, los tiempos de resolución de la Fase 2 pasan a ser similares a los de la Fase 1.

Por otra parte, se logra reducir el valor total de envidias para cuatro de las cinco instancias consideradas en la primera configuración, de $q = 3$, $T_c = 150$, tomando un valor promedio de 465, cuando este valor era 473,4 en la Fase 1. Como se puede observar, al igual que en el análisis de las variaciones de m , estas mejoras de envidia no son significativas, ya que son mejorías de menos del 2 %, por lo que es necesario analizar si el tiempo extra requerido por el *solver* para resolver la Fase 2 vale la pena. Como los tiempos de resolución promedio para esta configuración fueron de 3,2 segundos, si bien las mejoras de envidia no son significativas, no es costoso mejorarlas. Se puede observar cómo varía el valor total de envidias en la Fase 2 al variar la capacidad y cantidad de turnos de forma mixta en la Figura 4.10.

	$T_c = 150, q = 3$	$T_c = 30, q = 9$	$T_c = 12, q = 18$	$T_c = 6, q = 36$
Valor Preferencias	178,4	96,4	90,0	90,0
Preferencia mínima	1,0	1,0	1,0	1,0
Preferencia máxima	5,6	2,0	1,0	1,0
Preferencia media	1,98	1,07	1,00	1,00
Curtosis preferencias	0,54	15,67	∞	∞
Cant. asig. a espera	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Valor envidias	473,4	57,6	0,0	0,0
Envidia mín. de un estudiante	0,0	0,0	0,0	0,0
Envidia máx. de un estudiante	61,2	14,4	0,0	0,0
Envidia media de un estudiante	15,78	1,92	0,00	0,00
Curtosis envidias	1,66	4,55	∞	∞
Tiempo de resolución (s)	0,08	0,04	0,03	0,00

Tabla 4.9: Valores de los indicadores para el modelo de Fase 1 con variación mixta de T_c y q

	$T_c = 150, q = 3$	$T_c = 30, q = 9$	$T_c = 12, q = 18$	$T_c = 6, q = 36$
Valor Preferencias	178,4	96,4	90,0	90,0
Preferencia mínima	1,0	1,0	1,0	1,0
Preferencia máxima	5,4	2,0	1,0	1,0
Preferencia media	1,98	1,07	1,00	1,00
Curtosis preferencias	0,32	15,67	∞	∞
Cant. asig. a espera	0,0	0,0	0,0	0,0
Cant. mín. de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Cant. máx de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Cant. promedio de asignaciones a espera para un estudiante	0,0	0,0	0,0	0,0
Valor envidias	465,0	57,6	0,0	0,0
Envidia mín. de un estudiante	0,6	0,0	0,0	0,0
Envidia máx. de un estudiante	55,8	14,4	0,0	0,0
Envidia media de un estudiante	15,50	1,92	0,00	0,00
Curtosis envidias	0,60	4,55	∞	∞
Tiempo de resolución (s)	3,20	93,45	0,03	0,02

Tabla 4.10: Valores de los indicadores para el modelo de Fase 2 con variación mixta de T_c y q

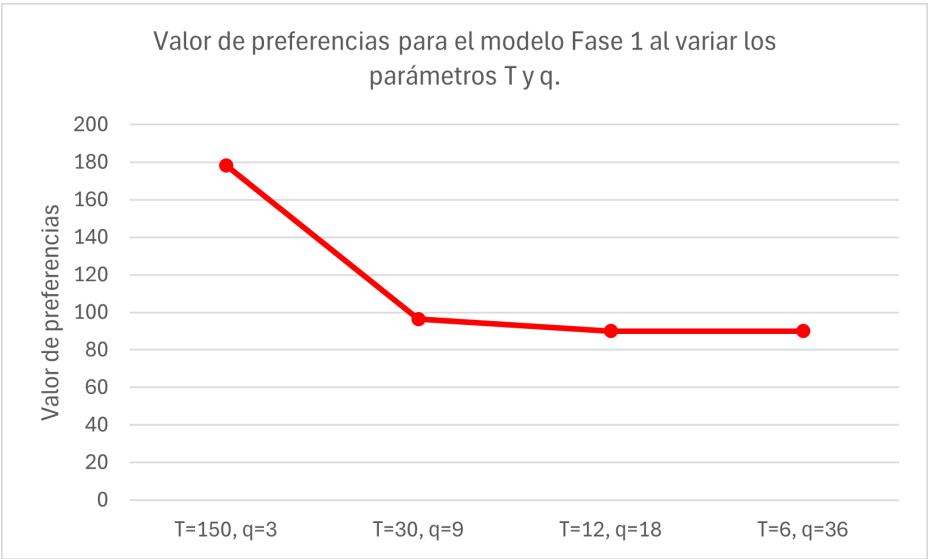


Figura 4.9: Valor de preferencias para el modelo Fase 1 al variar los parámetros T_c y q

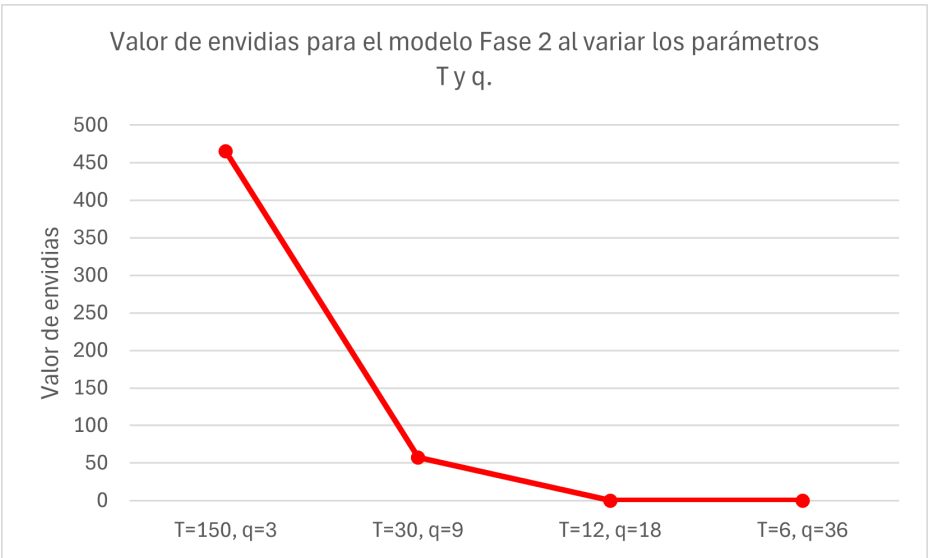


Figura 4.10: Valor de envidias para el modelo Fase 1 al variar los parámetros T_c y q

Capítulo 5

Conclusiones y Trabajo Futuro

En este trabajo se estudió el problema de asignación de estudiantes a turnos de cursos con cupos, considerando la superposición de turnos y maximizando las preferencias de dichos estudiantes. Esto se hizo a través de la formulación de dos modelos de optimización, referidos como Fase 1 y Fase 2, donde el resultado de la Fase 1 es utilizado en la Fase 2, por lo que se resuelven de forma secuencial. Para la Fase 1, se elaboró un modelo que maximiza la satisfacción de las preferencias de los estudiantes. En la Fase 2, el modelo elaborado permite obtener soluciones que mantienen la satisfacción de las preferencias obtenida en la Fase 1, pero con una menor envidia global, obteniendo así soluciones más justas que las que se obtienen en la Fase 1, sin la necesidad de ceder en las preferencias.

Se elaboraron en total 357 instancias de datos para analizar el comportamiento de los modelos frente a una gran variedad de configuraciones. Además, se definieron ciertos indicadores para analizar cuantitativamente las soluciones obtenidas por los modelos, lo que permitió comparar de forma cualitativa las distintas soluciones obtenidas con las diversas configuraciones. Para generar los datos, se elaboró un generador de datos que, a partir de ciertos parámetros modificables, es capaz de generar rápidamente archivos de datos con el formato adecuado para utilizar junto con los modelos propuestos. Con estas instancias de datos e indicadores, fue posible analizar las soluciones obtenidas considerando múltiples situaciones, observando tendencias en las soluciones al variar datos de entrada particulares.

El análisis computacional mostró que la Fase 1 puede ser resuelta en tiempos razonables para casos de más de 1000 estudiantes y otros de más de 100 cursos. Por otra parte, la Fase 2 experimenta un aumento considerable en los tiempos de resolución, incluso en casos con tan solo 60 estudiantes y 6 cursos, volviéndose inviable resolver esta fase con situaciones más grandes.

En el análisis de sensibilidad, se observó que la existencia de superposiciones da lugar a mejoras considerables en las soluciones obtenidas al resolver la Fase 2, permitiendo generar asignaciones más equitativas sin afectar el nivel de satisfacción alcanzado en la Fase 1. Por otro lado, en el resto de los casos, no se detectaron diferencias significativas entre ambas fases, salvo por el sustancialmente mayor tiempo requerido en la Fase 2.

Más allá de las comparaciones entre las fases, se observó que incrementar la capacidad y/o la cantidad de turnos disponibles permite obtener asignaciones que reflejan mejor las preferencias de los estudiantes y reducen las situaciones de envidia. Además, el análisis de la distribución de las preferencias entre los turnos muestra que, cuando la mayoría de los estudiantes elige un mismo turno como primera opción, se generan menores niveles de satisfacción y mayores niveles de inequidad. En contraste, cuando las preferencias se reparten de manera más uniforme entre los turnos disponibles, las asignaciones resultan más satisfactorias y con menores valores de envidia.

Si bien se logró cumplir con el objetivo del trabajo, se discutieron ciertas oportunidades de mejora, y posibilidades de trabajo futuro. Una de ellas es fomentar asignaciones más uniformes en cuanto a valores individuales de preferencias y envidias. Como la resolución en fases de los modelos propuestos busca reducir el valor global de preferencias y envidias, es posible que un estudiante se vea muy desfavorecido frente a otros, y que aun así el valor global sea el menor posible. En la mayoría de los casos de experimentación numérica considerados se observan diferencias significativas entre la cantidad mínima y máxima de asignaciones a espera de un estudiante, y entre la envidia mínima y máxima de un estudiante. Una posible mejora para minimizar estas desigualdades sería penalizar múltiples asignaciones de un estudiante a listas de espera, promoviendo así una distribución más uniforme de dichas asignaciones, que son las que tienen valores más grandes, tanto en términos de preferencias como de envidias.

Otra posible extensión de este trabajo sería analizar en detalle las diferencias en los valores de envidias entre las Fases 1 y 2 ante casos con la presencia de superposiciones horarias. Aún no se ha determinado con exactitud qué particularidad de la existencia de superposiciones en la estructura de la asignación explica estas diferencias. Este aspecto queda pendiente como trabajo futuro, con el objetivo de examinar detalladamente la influencia de las superposiciones en la redistribución de las asignaciones y en la disminución de la envidia.

Por otra parte, como se explicó en la Sección 4.3.2, las instancias de datos elaboradas para analizar el comportamiento de los modelos frente a cambios en las penalizaciones de preferencias demasiado elevadas no fueron de utilidad. Por esta razón, resultaría interesante considerar un caso en el cual se asignen preferencias elevadas, y así poder observar cómo influye el hecho de penalizar cada vez más este tipo de preferencias en los indicadores de las soluciones obtenidas.

Por último, dado que la resolución de la Fase 2 se torna prácticamente inviable para situaciones con una gran cantidad de estudiantes y cursos, sería relevante explorar enfoques alternativos para obtener asignaciones más justas, que también respeten las preferencias de los estudiantes. Además, ya que en la mayoría de las instancias de datos analizadas no se obtienen mejores soluciones en la Fase 2 frente a la Fase 1, se podría indagar sobre las condiciones que deben cumplirse para que la Fase 2 mejore la envidia global, y de esta forma poder determinar si sería necesario aplicarla en casos reales.

Referencias

- Agustín-Blas, L. E., Salcedo-Sanz, S., Ortiz-García, E., Pérez-Bellido, A., y Portilla-Figueras, A. (2008). Assignment of students to preferred laboratory groups using a hybrid grouping genetic algorithm [Conference paper]. En (p. 48 – 52). doi: 10.1109/HIS.2008.37
- Beroš, I., y Meter, J. (2015). An integer programming model for assigning students to elective courses. *Croatian Operational Research Review*, 6, 511-524. doi: 10.17535/corr.2015.0038
- Bichler, M., y Merting, S. (2021). Randomized scheduling mechanisms: Assigning course seats in a fair and efficient way [Article]. *Production and Operations Management*, 30(10), 3540 – 3559. doi: 10.1111/poms.13449
- Chartier, T. P., Ellison, V., y Langville, A. N. (2014, 01 de Dec). A davidson college multi-objective assignment problem: a case study. *4OR*, 12(4), 379-401. doi: 10.1007/s10288-014-0259-2
- Chen, V., y Hooker, J. N. (2023). A guide to formulating fairness in an optimization model. *Annals of Operations Research*, 326(1), 581–619.
- Chiarandini, M., Fagerberg, R., y Gualandi, S. (2019). Handling preferences in student-project allocation [Article]. *Annals of Operations Research*, 275(1), 39 – 78. doi: 10.1007/s10479-017-2710-1
- Darmann, A., Döcker, J., Dorn, B., y Schneckenburger, S. (2022, 01 de Mar). Simplified group activity selection with group size constraints. *International Journal of Game Theory*, 51(1), 169-212. doi: 10.1007/s00182-021-00789-7
- Domenech, B., y Lusa, A. (2016). A milp model for the teacher assignment problem considering teachers' preferences [Article]. *European Journal of Operational Research*, 249(3), 1153 – 1160. doi: 10.1016/j.ejor.2015.08.057
- Gralla, E., Goentzel, J., y Fine, C. (2014). Assessing trade-offs among multiple objectives for humanitarian aid delivery using expert preferences. *Production and Operations Management*, 23(6), 978–989.
- Heitmann, H., y Brüggemann, W. (2014). Preference-based assignment of university students to multiple teaching groups [Article]. *OR Spectrum*, 36(3), 607 – 629. doi: 10.1007/s00291-013-0332-9
- Manlove, D., Milne, D., y Olaosebikan, S. (2018). An integer programming approach to the student-project allocation problem with preferences over

- projects. En *International symposium on combinatorial optimization* (pp. 313–325).
- Netzer, A., Meisels, A., y Zivan, R. (2016). Distributed envy minimization for resource allocation. *Autonomous Agents and Multi-Agent Systems*, 30, 364–402.
- Nogareda, A.-M., y Camacho, D. (2017, 01 de Sep). Optimizing satisfaction in a multi-courses allocation problem combined with a timetabling problem. *Soft Computing*, 21(17), 4873–4882. doi: 10.1007/s00500-016-2375-8
- Proll, L. G. (1972, 01 de Jun). A simple method of assigning projects to students. *Journal of the Operational Research Society*, 23(2), 195–201. doi: 10.1057/jors.1972.29
- Schneider, J. J., Hirtreiter, C., y Morgenstern, I. (2009). On the problem of finding a suitable distribution of students to universities in germany. *Physica A: Statistical Mechanics and its Applications*, 388(20), 4475–4483. doi: <https://doi.org/10.1016/j.physa.2009.07.010>
- van den Broek, J., Hurkens, C., y Woeginger, G. (2009). Timetabling problems at the tu eindhoven [Article]. *European Journal of Operational Research*, 196(3), 877 – 885. doi: 10.1016/j.ejor.2008.04.038
- Varone, S., y Schindl, D. (2013). Course opening, assignment and timetabling with student preferences [Conference paper]. En (p. 153 – 158).
- Willoughby, K., y Zappe, C. (2006). A methodology to optimize foundation seminar assignments [Article]. *Journal of the Operational Research Society*, 57(8), 950 – 956. doi: 10.1057/palgrave.jors.2602070

Apéndice A

Apéndice

A.1. Estado del arte

Problemas de asignación en el ámbito educativo considerando preferencias y otras características

Acceso: https://drive.google.com/file/d/1kDWIg0Fr5Xs6HXN81nV_uzjnLthjeh3b/view?usp=sharing

A.2. Modelo unificado

```
set E; # Set de estudiantes
set C; # Set de cursos
set C_e{e in E} within C; # Cursos en los que el estudiante e expresa preferencias
set T_c{c in C}; # Set de turnos para cada curso
set T; # Union de todos los sets de turnos para cada curso
set S{c in C, t in T_c[c]} within {C cross T}; # Pares curso-turno (i,j)
# que se superponen con el turno t del curso c, union (c,t)

param p{e in E, c in C, t in T_c[c]}; # Preferencia del estudiante e
# por el turno t del curso c
param k; # Valor de preferencia que se considera muy elevado
param n; # Valor que multiplica las preferencias mayor o igual a k,
# para asignarles mayor peso
param r{e in E}; # Maximo valor de preferencia
# que el estudiante e marco por algun turno de algun curso, + 1
param q{c in C, t in T_c[c]}; # Capacidad maxima del turno t del curso c
param a; # Peso asignado al objetivo asociado con maximizar preferencias
param h; # Maximo valor de envidia total que un estudiante
# puede tener (sumando la envidia con cada estudiante y para cada curso)
```

```

var x{e in E, c in C, t in T_c[c]} binary;
# 1 si el estudiante e es asignado al turno t del curso c, 0 si no
var w{e in E, c in C} binary;
# 1 si el estudiante e queda en lista de espera para el curso c, 0 si no
var v{e in E, f in E, c in (C_e[e] inter C_e[f]) : e != f} >= 0;
# El estudiante e envidia al estudiante f por el curso c con un valor de v[e,f,c]
var costo_preferencia; # Mide la sumatoria de las preferencias
var costo_envidia; # Mide la sumatoria de las envidias

# Se busca una asignacion que minimice la suma de las preferencias
# (penalizando aun mas para preferencias mayores o iguales a k)
# cuando a vale 1, y la envidia cuando a vale 0
minimize costo: a*(costo_preferencia) + (1-a)*(costo_envidia);

s.t. factor_preferencias:
    sum{e in E, c in C_e[e], t in T_c[c] : p[e,c,t] < k}
        p[e,c,t] * x[e,c,t] +
    sum{e in E, c in C_e[e], t in T_c[c] : p[e,c,t] >= k}
        n * p[e,c,t] * x[e,c,t] +
    sum{e in E, c in C_e[e]}
        n * r[e] * w[e,c] = costo_preferencia;

s.t. factor_envidia:
    sum{e in E, f in E, c in (C_e[e] inter C_e[f]) : e != f}
        v[e,f,c] = costo_envidia;

# Cada estudiante debe ser asignado a solo un turno de un curso
# por el que expresa preferencia, o quedar en lista de espera de ese curso
# si no puede ser asignado a ningun turno por el cual expreso preferencias
s.t. asignacion {e in E, c in C_e[e]}:
    sum{t in T_c[c] : p[e,c,t] < 100} x[e,c,t] + w[e,c] = 1;

# Un estudiante no debe ser asignado a un turno de un curso
# por el que no expresa preferencias, ni quedar en lista de espera de ese curso
s.t. no_asignacion {e in E, c in (C diff C_e[e])}:
    sum{t in T_c[c]} x[e,c,t] + w[e,c] = 0;

# No se puede exceder la capacidad de ningun turno de ningun curso
s.t. capacidad {c in C, t in T_c[c]}:
    sum{e in E} x[e,c,t] <= q[c,t];

# Si un turno de un curso se superpone con otro turno de algun curso,
# un estudiante no puede ser asignado a ambos
s.t. superposicion {e in E, i in C_e[e], j in T_c[i]}:
    sum{(c,t) in S[i,j] : (c in C_e[e]) and (t in T_c[c]) and (p[e,c,t] < 100)}

```

```

x[e,c,t] <= 1;

# Para el caso de a = 0, se define la envidia: el estudiante e envidia
# al estudiante f por el curso con un valor de v[e,f,c]
s.t. definicion_envidia {e in E, f in E, c in (C_e[e] inter C_e[f]) : e != f}:
    v[e,f,c] >= (sum{t in T_c[c] : (p[e,c,t] < 100) or (p[f,c,t] < 100)})
    (x[e,c,t]*p[e,c,t] - x[f,c,t]*p[e,c,t]) + w[e,c]*r[e] - w[f,c]*r[e])*(1-a);

# Para el caso de a = 0, se pide que las preferencias sean menores o iguales
# que las que da para el caso a = 0, b = 1
s.t. limite_preferencia_total:
    (1-a)*costo_preferencia <= 0; #reemplazar por el valor obtenido con a = 1

# INUTILIZADA. Se limita el maximo valor de envidia total que un estudiante puede tener
# (sumando la envidia con cada estudiante y para cada curso)
# s.t. envidia_maxima {e in E}:
#     sum{f in E, c in C_e[e] : e != f} (1-a)*v[e,f,c] <= h;

end;

```

A.3. Cantidad máxima de superposiciones

La máxima cantidad de superposiciones que podría haber en una configuración dada se puede calcular como las combinaciones de la cantidad de turnos totales tomados de a 2, de la siguiente manera:

$$\frac{(\sum_{c \in C} |T_c|)!}{2!((\sum_{c \in C} |T_c|) - 2)!}$$

Ya que en la configuración base, todos los cursos se generarán con la misma cantidad de turnos, la cantidad de turnos totales será de:

$$\sum_{c \in C} |T_c| = |C| * |T_c| = 3 * 3 = 9$$

Reemplazando en la ecuación anterior, la cantidad de superposiciones posibles sería la siguiente:

$$\frac{9!}{2!(9-2)!} = \frac{9!}{2!7!} = 36$$

A.4. Funcionamiento del generador de datos

```

import random
import numpy as np

```

```
import itertools

#####

# CONJUNTOS Y PARAMETROS EDITABLES

# Estudiantes
student_amount = 30

# Cursos
course_amount = 3

# Turnos
shift_amount = 3

# La capacidad de los turnos de los cursos sigue una distribucion
# normal con media q_mean y desviacion estandar q_stdev
capacity_amount = 9

# Cantidad de pares de curso-turno que se superponen
num_pairs = 0

# Valor de preferencia considerado muy elevado
k = 10

# El costo de las preferencias mayores o iguales a k se multiplican por n
n = 1

# La probabilidad de asignarle la preferencia 1 al turno i es
# 'multiplier' veces mas que asignarsela al turno i+1
multiplier = 1.5

# Ponderador del costo de preferencias
ponderador_a = 1

# Valor de envidia considerado muy elevado para un estudiante
# considerando todos los cursos y los demas estudiantes
# h = 1000

# Nombre del archivo que se generara al correr el codigo
file_name = 'test'

#####

# CONJUNTOS Y PARAMETROS NO EDITABLES
```

```

# Estudiantes
E = [f"e{i}" for i in range(1, student_amount + 1)]

# Cursos
C = [f"c{i}" for i in range(1, course_amount + 1)]

#Turnos para cada curso
T_c = {c: [f"{c}t{j}" for j in range(1, shift_amount + 1)] for c in C}

# Conjunto para almacenar la unión de todos los conjuntos T_c
T = {t for sublist in T_c.values() for t in sublist}
T_sorted = sorted(T, key=lambda x: (int(x[1:x.index('t')]), int(x[x.index('t') + 1:])))

# Capacidad
q = {c: {t: capacity_amount for t in T_c[c]} for c in C}

# Superposición de turnos
# Inicializar S
S = {c: {t: set() for t in T_c[c]} for c in C}
for c, shifts in S.items():
    for t in shifts:
        S[c][t].add(t)

def generate_random_pairs(num_pairs):
    # Generar todos los pares posibles
    all_pairs = list(itertools.combinations(T, 2))

    # Asegurarse de no pedir mas pares que los posibles
    total_possible_pairs = len(all_pairs)
    if num_pairs > total_possible_pairs:
        print("ERROR: THE NUMBER OF PAIRS REQUESTED EXCEEDS THE TOTAL POSSIBLE PAIRS")

    # Elegir al azar los pares solicitados
    random_pairs = random.sample(all_pairs, num_pairs)

    return random_pairs

# Generar pares al azar
random_pairs = generate_random_pairs(num_pairs)
# Actualizar los sets S para cada par (a,b) seleccionado
for a, b in random_pairs:
    course_a = a[:a.rfind('t')]
    course_b = b[:b.rfind('t')]
    S[course_a][a].add(b)
    S[course_b][b].add(a)

```

```

# Asignar preferencias a los turnos considerando las probabilidades
p = {e: {c: {t: None for t in T_c[c]} for c in C} for e in E}

def shift_chooser(n):
    # Generar una distribución de probabilidad personalizada
    probs = np.array([1/multiplier ** i for i in range(n)])
    # Normalizar las probabilidades para que sumen 1
    probs = probs / np.sum(probs)

    chosen_shifts = [] # Lista para almacenar los turnos elegidos

    # Iterativamente elegir turnos hasta que todos sean elegidos
    while len(chosen_shifts) < n:
        # Elegir un turno aleatorio según la distribución de probabilidades
        chosen_shift_index = np.random.choice(np.arange(n), p=probs)
        chosen_shift = chosen_shift_index + 1
        chosen_shifts.append(chosen_shift)

        # Eliminar el turno elegido de las probabilidades
        probs[chosen_shift_index] = 0
        if np.sum(probs) > 0:
            probs = probs / np.sum(probs)

    return chosen_shifts

for e in E:
    for c in C:
        # Obtener los turnos elegidos para el curso c
        chosen_shifts = shift_chooser(len(T_c[c]))
        # Asignar las preferencias en el orden que se eligieron los turnos
        preference_value = 1
        for shift in chosen_shifts:
            p[e][c][f"{c}t{shift}"] = preference_value
            preference_value += 1

# Conjuntos C_e que muestran los cursos que el estudiante e expresa preferencias
C_e = {e: {c for c in C if any(p[e][c][t] != 100 for t in T_c[c])} for e in E}

# Parámetro r para cada estudiante
r = {}
for e in E:
    max_pref = max(pref for c in C for pref in p[e][c].values() if pref != 100)
    r[e] = max_pref + 1

#####

```

OUTPUT

```

with open(f"{file_name}.dat", 'w') as f:
    print("set E :=", file=f)
    print(*E, sep='\n', file=f)
    print(";", file=f)

    print("set C :=", file=f)
    print(*C, sep='\n', file=f)
    print(";", file=f)

    for c, t in T_c.items():
        print(f"set T_c[{c}] := {' '.join(t)};", file=f)

    print("set T :=", file=f)
    print(' '.join(T_sorted), ";", file=f)

    for c, shifts in S.items():
        for t, pairs in shifts.items():
            course = t.split('t')[0]
            values = ' '.join(f"({p}, {t})" if p == course
                              else f"({p.split('t')[0]}, {p})" for p in pairs)
            print(f"set S[{c},{t}] := {values};", file=f)

    print("param p :=", file=f)
    for e in E:
        for c in C:
            for t in T_c[c]:
                print(f"[{e},{c},{t}] {p[e][c][t]}", file=f)
    print(";", file=f)

    for e in E:
        courses = ' '.join(sorted(C_e[e], key=lambda x: C.index(x)))
        print(f"set C_e[{e}] := {courses};", file=f)

    print("param r :=", file=f)
    for e in E:
        print(f"[{e}] {r[e]}", file=f)
    print(";", file=f)

    print("param q :=", file=f)
    for c in C:
        for t in T_c[c]:
            print(f"[{c},{t}] {q[c][t]}", file=f)
    print(";", file=f)

```

```

print(f"param k := {k};", file=f)
print(f"param n := {n};", file=f)
print(f"param a := {ponderador_a};", file=f)
#   print(f"param h := {h};", file=f)
print("end;", file=f)

```

```
#####
```

A.5. Archivo para obtener los indicadores

```

reset;
model modelo_pref+env.mod;

data 'base.dat';

option solver cplex;
option cplex_options 'timing = 1 display = 1 mipdisplay = 2 time = 300';

#option solver gurobi;
#option gurobi_options 'timing = 1 outlev = 1 timelim = 300';

solve;

option display_width 150;

#Preferencia mínima

var min_pref integer;
let min_pref := min{e in E, c in C, t in T_c[c]:
round(x[e,c,t]) = 1} round(x[e,c,t]*p[e,c,t]);

#Preferencia máxima

var max_pref integer;
let max_pref := max{e in E, c in C, t in T_c[c]:
round(x[e,c,t]) = 1} round(x[e,c,t]*p[e,c,t]);

#Preferencia media

var contador_asignaciones integer;
let contador_asignaciones :=
    sum{e in E, c in C, t in T_c[c]} round(x[e,c,t]);

var suma_preferencias integer;
let suma_preferencias :=

```

```

    sum{e in E, c in C, t in T_c[c]} round(x[e,c,t]*p[e,c,t]);

var preferencia_media;
let preferencia_media := suma_preferencias/contador_asignaciones;

#Distribución de preferencias

param max_preferencia integer;
let max_preferencia :=
    max{e in E, c in C, t in T_c[c]: p[e,c,t] < 100} p[e,c,t];

var preferencias{1..max_preferencia} integer;
for {j in 1..max_preferencia} {
    for {e in E, c in C, t in T_c[c]} {
        if (round(x[e,c,t]) = 1 and p[e,c,t] = j) then {
            let preferencias[j] := preferencias[j] + 1
        }
    }
};

#Imprimir los indicadores de preferencias

display costo_preferencia;
display min_pref;
display max_pref;
display preferencia_media;
display preferencias;

#Cantidad total de asignaciones a lista de espera

var espera_total integer;
let espera_total := sum{e in E, c in C} round(w[e,c]);

#Cantidad mínima de asignaciones a espera (para un estudiante)

var asignaciones_espera{e in E} integer;
for {e in E} {
    let asignaciones_espera[e] := sum{c in C} round(w[e,c])
};

var min_espera integer;
let min_espera := min{e in E} asignaciones_espera[e];

#Cantidad máxima de asignaciones a espera (para un estudiante)

var max_espera integer;

```

```

let max_espera := max{e in E} asignaciones_espera[e];

#Cantidad promedio de asignaciones a espera (entre los estudiantes)

var espera_media;
let espera_media := espera_total / card(E);

#Imprimir los indicadores de espera

display espera_total;
display min_espera;
display max_espera;
display espera_media;

#Envidia total

var v_aux {e in E, f in E, c in (C_e[e] inter C_e[f]): e != f};
for {e in E, f in E, c in (C_e[e] inter C_e[f]) : e != f} {
  let v_aux[e,f,c] :=
    max(0, round(sum{t in T_c[c] : (p[e,c,t] < 100) or (p[f,c,t] < 100)}
      (x[e,c,t]*p[e,c,t] - x[f,c,t]*p[e,c,t]) + w[e,c]*r[e] - w[f,c]*r[e]))
};

var envidia_estudiante{e in E} integer;
for {e in E} {let envidia_estudiante[e] :=
  sum{f in E, c in (C_e[e] inter C_e[f]) : e != f} v_aux[e,f,c]};

var sumar_env;
let sumar_env :=
  sum{e in E, f in E, c in (C_e[e] inter C_e[f]) : e != f} v_aux[e,f,c];

#Envidia mínima (por estudiante)

var min_envidia integer;
let min_envidia := min{e in E} envidia_estudiante[e];

#Envidia máxima (por estudiante)

var max_envidia integer;
let max_envidia := max{e in E} envidia_estudiante[e];

#Envidia media (estudiante)

var envidia_media;
let envidia_media := sumar_env / card(E);

```

```

#Distribución de envidias

param inicio integer;
let inicio := min_envidia;

param fin integer;
let fin := max_envidia;

var envidias{inicio..fin} integer;
for {j in inicio..fin} {
    for {e in E} {
        if envidia_estudiante[e] = j then {
            let envidias[j] := envidias[j] + 1
        }
    }
};

#Imprimir los indicadores de envidia

display sumar_env;
display min_envidia;
display max_envidia;
display envidia_media;
for {i in inicio..fin} {
    if envidias[i] > 0 then {
        display i, envidias[i]
    }
};

```

A.6. Envidia por listas de espera

En esta sección se determina la envidia máxima a partir de asignaciones a listas de espera para un conjunto de datos dado. La cantidad total de asignaciones en cualquiera de los modelos propuestos se puede calcular sumando la cantidad total de asignaciones a espera, y la cantidad total de asignaciones a turnos. Por lo que, dada una solución a un modelo en donde la cantidad total de asignaciones es n , con una cantidad de asignaciones a espera a , y una cantidad de asignaciones a turnos b , se cumplirá que $a + b = n$. Se busca demostrar que el máximo valor de $a \cdot b$ se da cuando $a = b = \frac{n}{2}$.

La desigualdad de la media aritmética y geométrica demuestra que:

1. $(a + b)/2 \geq \sqrt{a \cdot b}$
2. $(a + b)/2 = \sqrt{a \cdot b} \iff a = b$

Esto quiere decir que la multiplicación toma su mayor valor cuando $a = b$

En nuestro caso, tenemos que $a + b = n$, por lo que la multiplicación toma su mayor valor cuando $a = b = \frac{n}{2}$.