

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“INTERROGATION SCHEDULING PROBLEM: UNA
ESTRATEGIA BASADA EN PREFERENCIAS DE
PROFESORES Y ALUMNOS”

KEVIN ALEXANDER LAGOS LAVÍN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Dra. María Cristina Riff Rojas
Profesor Correferente: Dra. Elizabeth Montero Ureta

Agosto - 2021

DEDICATORIA

Dedico este trabajo a mi madre, Marisol Lavín, por todo el amor que me ha brindado durante todos los años de mi vida y el amor incondicional que me ha hecho feliz. A mi padre, Raúl Lagos, por todo el sacrificio y trabajo que ha hecho por mi y por la familia, por todos los valores que me inculcó desde pequeño y que me hicieron la persona que soy actualmente.

AGRADECIMIENTOS

Quiero agradecer principalmente a la profesora María Cristina Riff, quien me brindó este tema, quien con todo su conocimiento y buena onda me guió en este proceso de la creación de mi memoria y me generó este gran interés de esta increíble área, apoyándome y logrando entenderme durante las distintas dificultades que pude haber presentado en las asignaturas que cursé previamente con ella.

Además, deseo mencionar a cada una de las personas que estuvieron presentes durante toda mi vida universitaria y que aún mantengo contacto con ellas, ya que no hubiese sido posible llegar hasta acá sin el apoyo que me brindaron durante todos estos años.

A mis compañeros de la universidad, los cuales generamos un grupo de estudio bastante genial, y que si bien muchos de ellos cayeron durante el camino, supieron levantarse durante todo el proceso: Andrés Navarro, Sonny Muñoz, Bárbara Uribe, Javier Moreno, Sebastián Muñoz, Matías Santander, Ignacio Oñate, Javier Castro, Julio Bueno, Fabián Bugueño y Antonia Cádiz. También quiero agradecer a otros amigos más, que si bien, no somos compañeros de la misma generación, ni uno tampoco estudió en la misma universidad, tuvieron un impacto bastante positivo y logramos formar una bonita amistad: Bastián Quezada, Matías Valenzuela, Matías Arancibia y Eliacer Sandoval.

A mis amigos de la vida y las fiestas, los cuales los conozco hace años tanto desde mi infancia como del colegio: César Gutierrez, Dennis Gonzalez, Sebastián Cataldo, Nicolás Bravo, Aaron Valenzuela, Jorge Uberuaga, Francisca Vallejos, Luna Herrera, Fabián Soto, Gonzalo Carrasco, Gonzalo Cornejo, Guillermo Salas y Sebastián Espinoza.

Agradezco inmensamente a Valentina Allendes Luke, quien me ha entregado estos últimos meses mucho amor y cariño, acompañándome en todo este proceso de finalización de mi carrera y que sin sus mensajes de apoyo diarios, me hubiese echado para abajo en bastantes ocasiones. Te amo muchísimo.

Finalmente, deseo agradecer a las distintas playlists de lofi hip-hop, las cuales me acompañaron en todas las noches de desvelo que realicé escribiendo esta memoria, y a los cafés de Vainilla Latte que me acompañaron en las distintas noches de invierno.

RESUMEN

Resumen— En Chile, distintos profesores de diferentes asignaturas desean muchas veces realizar interrogaciones a sus alumnos y por lo tanto deben generar una planificación previa acorde a la disponibilidad horaria suya y de los alumnos, muchas veces incluso fuera de horario de la asignatura. Por lo general, dicha tarea se realiza de forma manual y ocupa una gran cantidad de tiempo, dada la alta complejidad del problema. Por lo anterior, se hace necesario un mecanismo para generar planificaciones de interrogaciones de mejor manera para reducir el tiempo de esta tarea, dándole distintas opciones de planificación a los profesores para poder escoger.

Palabras Clave— Horario; Preferencias; Algoritmos; Interrogaciones; Planificación

ABSTRACT

Abstract— In Chile, different professors of different subjects often want to interrogate their students and therefore they must generate a prior planning according to their and students' time availability, many times even outside of the course schedule. Generally, this task is carried out manually and takes a great deal of time given the high complexity of the problem. Therefore, a mechanism is necessary to generate interrogating schedules in a better way to reduce the time of this task, giving professors different planning options to choose from.

Keywords— Schedule; Preferences, Algorithms; Interrogations; Planification

GLOSARIO

CSP: Constraint Satisfaction Problem

CSOP: Constraint Satisfaction and Optimization Problem

B&B: Branch and Bound

B&C: Branch and Cut

ISP: Interrogation Scheduling Problem

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 REPLANIFICACIÓN	3
1.2 ASIGNACIONES INTELIGENTES	4
1.3 USO DE DOODLE PARA PLANIFICAR REUNIONES	5
1.4 SITUACIÓN ACTUAL EN ASIGNATURA DE INTELIGENCIA ARTIFICIAL UTFSM	6
1.5 USO DE DOODLE EN LA UTFSM	6
1.6 OBJETIVOS	8
1.6.1 OBJETIVO GENERAL	8
1.6.2 OBJETIVOS ESPECÍFICOS	8
1.7 RESUMEN	8
CAPÍTULO 2: ESTADO DEL ARTE	10
2.1 PROBLEMAS RELACIONADOS	10
2.1.1 EXAM TIMETABLING	10
2.2 COMPLEJIDAD DEL PROBLEMA	11
2.3 PROBLEMA DE SATISFACCIÓN Y OPTIMIZACIÓN DE RESTRICCIONES	11
2.4 MÉTODOS DE RESOLUCIÓN	12
2.4.1 BRANCH AND BOUND	13
2.4.2 BRANCH AND CUT	14
2.5 ALGORITMOS INCOMPLETOS	14
2.5.1 ALGORITMO HILL-CLIMBING	14
2.5.2 ALGORITMO TABÚ SEARCH	15
2.6 RESUMEN	16
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	17
3.1 ASIGNACIÓN DE BLOQUES CONSIDERANDO PREFERENCIAS DE PROFESORES Y ALUMNOS	17
3.1.1 CONJUNTOS	17
3.1.2 PARÁMETROS	18
3.1.3 VARIABLE DE DECISIÓN	18

3.1.4	VARIABLE AUXILIAR	18
3.1.5	MODELO MATEMÁTICO	19
3.1.6	FUNCIÓN OBJETIVO	19
3.1.7	RESTRICCIONES	19
3.2	CAMBIOS EN DISPONIBILIDAD DE HORARIOS	20
3.2.1	CONJUNTOS	20
3.2.2	PARÁMETROS	20
3.2.3	VARIABLE DE DECISIÓN	21
3.2.4	MODELO MATEMÁTICO	21
3.2.5	FUNCIÓN OBJETIVO	21
3.3	PLANIFICACIÓN PARA ASIGNATURA DE INTELIGENCIA ARTIFICIAL UTFSM	21
3.3.1	HERRAMIENTAS A UTILIZAR	22
3.4	CONCLUSIONES	22
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		24
4.1	IMPLEMENTACIÓN DEL ISP SOLVER PARA INTELIGENCIA ARTIFICIAL	24
4.1.1	MENÚ PRINCIPAL	24
4.1.2	PLANIFICACIÓN DE UN NUEVO HORARIO	25
4.1.3	EJECUCIÓN DE INSTANCIA DE PRUEBA EN ISP SOLVER	26
4.1.4	REPLANIFICACIÓN DE HORARIO	31
4.2	VALOR DEL PROGRAMA	32
4.3	EXPERIMENTACIÓN TOTALMENTE ALEATORIA	33
4.3.1	RESULTADOS DEL TIEMPO DE EJECUCIÓN	33
4.3.2	RESULTADOS DE PREFERENCIAS Y ASIGNACIONES INTELIGENTES	34
4.4	EXPERIMENTACIÓN CON DISPONIBILIDAD	36
4.5	EXPERIMENTACIÓN CON DISPONIBILIDADES	38
4.6	INSTANCIAS CON DISPONIBILIDADES NORMALES	39
4.7	INSTANCIAS CON ALTA DISPONIBILIDAD	40
4.8	INSTANCIAS CON BAJA DISPONIBILIDAD	42
4.9	ANÁLISIS DE SENSIBILIDAD RESPECTO A BLOQUES ADICIONALES	44
4.9.1	CON UN BLOQUE EXTRA	44
4.9.2	CON DOS BLOQUES EXTRAS	46
4.10	RESULTADOS GENERALES	48
4.11	CONCLUSIONES	50
CAPÍTULO 5: CONCLUSIONES		51
5.1	RECOMENDACIONES	52
5.2	TRABAJO FUTURO	52
ANEXOS		53
REFERENCIAS BIBLIOGRÁFICAS		56

ÍNDICE DE FIGURAS

1	Una encuesta de ejemplo realizada en Doodle.	5
2	Encuesta de ejemplo exportada a formato Excel.	6
3	Gráfico de torta respecto a la utilización de Doodle en la universidad.	7
4	Histograma de motivos de utilización de Doodle en la universidad.	7
5	Diagrama de las tres componentes principales de Branch & Bound.	13
6	Flujo general del programa, la línea punteada representa un input opcional que se utiliza para el proceso de replanificación.	22
7	Captura de Menú Principal del programa.	24
8	Captura de sección para agregar valoraciones a alumnos.	25
9	Captura de sección que muestra la mejor planificación generada.	26
10	Horarios de prueba para la generación de planificaciones.	27
11	Mejor planificación obtenida con el ISP Solver	27
12	Segunda planificación obtenida por el ISP Solver.	28
13	Tercera planificación obtenida por el ISP Solver.	28
14	Cuarta planificación obtenida por el ISP Solver.	29
15	Quinta planificación obtenida por el ISP Solver.	29
16	Sexta planificación obtenida por el ISP Solver.	30
17	Séptima y última planificación obtenida por el ISP Solver.	30
18	Archivo excel con las nuevas disponibilidades del alumno 1.	31
19	Resultados de la replanificación generadas por el ISP Solver.	32
20	Promedio del tiempo de ejecución de las tres simulaciones.	34
21	Gráfico de preferencias satisfechas en función de la cantidad de alumnos.	35
22	Gráfico de asignaciones inteligentes en función de la cantidad de alumnos.	35

23	Promedio de preferencias satisfechas y asignaciones inteligentes en función de la cantidad de alumnos.	36
24	Histograma de disponibilidad de horarios.	37
25	Histograma de preferencias de horarios	37
26	Histograma de preferencias satisfechas con disponibilidades normales.	39
27	Histograma de asignaciones inteligentes con disponibilidades normales.	40
28	Histograma de preferencias satisfechas con alta disponibilidad.	41
29	Histograma de asignaciones inteligentes con alta disponibilidad.	42
30	Histograma de preferencias satisfechas con baja disponibilidad.	43
31	Histograma de asignaciones inteligentes con baja disponibilidad.	44
32	Histograma de preferencias satisfechas con baja disponibilidad y 31 bloques.	45
33	Histograma de asignaciones inteligentes con baja disponibilidad y 31 bloques.	46
34	Histograma de preferencias satisfechas con baja disponibilidad y 32 bloques.	47
35	Histograma de asignaciones inteligentes con baja disponibilidad y 32 bloques.	48
36	Diagrama de flujo del programa para realizar una planificación.	54
37	Diagrama de flujo del programa para realizar una replanificación.	55

ÍNDICE DE TABLAS

1	Ejemplo de solución a la planificación de interrogaciones tomando en consideración las preferencias de los alumnos.	3
2	Ejemplo de solución a la replanificación de interrogaciones.	3
3	Ejemplo de solución con la realización de asignaciones inteligentes.	4
4	Cantidad de soluciones factibles e infactibles con disponibilidades normales.	40
5	Cantidad de soluciones factibles e infactibles con disponibilidades bajas.	42
6	Cantidad de soluciones factibles e infactibles con disponibilidades bajas.	44

7	Cantidad de soluciones factibles e infactibles con disponibilidades bajas y 31 bloques.	46
8	Cantidad de soluciones factibles e infactibles con disponibilidades bajas y 32 bloques.	48
9	Resultados Obtenidos respecto a la distribución de preferencias satisfechas. . .	49
10	Resultados Obtenidos respecto a la distribución de asignaciones inteligentes. .	49
11	Resultados Obtenidos respecto a la factibilidad de las soluciones.	49

INTRODUCCIÓN

Dentro de los problemas que deben enfrentar profesores y alumnos de universidades está la planificación de interrogaciones de alguna asignatura, en la cual el profesor debe asignar de forma manual el horario en que cada alumno debe asistir. Generar esta planificación es una tarea bastante compleja a realizar ya que no todos los alumnos tienen la misma disponibilidad horaria y las mismas preferencias de horario. Además, pueden existir casos especiales en los cuales en último momento algún alumno puede informar que por alguna emergencia no podrá asistir y debe realizarse nuevamente la asignación de horarios para la interrogación, quitando bastante tiempo y siendo algunas veces tedioso este proceso al no existir un software que resuelva este problema.

Por otro lado, la complejidad del problema presentado es alta, ya que el número de posibles combinaciones para encontrar una solución es bastante grande, más aún considerando que agregar las preferencias de los alumnos muchas veces no se toma en cuenta, ya que le agregaría mucha más complejidad al problema. Debido a lo anterior, este problema se puede ver como un problema de satisfacción de restricciones, el cual puede ser resuelto a través de distintas técnicas basadas en conocimiento.

Contar con un software que permita realizar distintas planificaciones de interrogaciones teniendo en cuenta las preferencias de profesores y alumnos resultaría muy beneficioso para ambos actores, reduciendo considerablemente el tiempo para el profesor y haciendo la tarea mucho más fácil para generar distintas planificaciones a escoger para su asignatura. Por otro lado, se genera valor para los alumnos ya que se tendrían en cuenta sus preferencias de bloques horarios y también obtendrían esta información de planificación más rápida para organizarse, tanto para los casos donde se deba replanificar por alguna emergencia como también en el caso en el que el profesor decida cambiar el horario.

Este documento está organizado de la siguiente manera. Inicialmente, en la Sección 1 se define el problema a abordar en la presente memoria y los objetivos tanto generales como específicos. Luego, en la Sección 2 se realiza una revisión de la literatura asociada a los problemas relacionados a la planificación de asignaturas y las distintos algoritmos que se utilizan para resolver los problemas de asignación. En la Sección 3 se presenta la propuesta de solución, sus fundamentos y la metodología aplicada para resolver el problema. Posteriormente, en la Sección 4 se muestra la implementación del software *ISP Solver*, el cual le da la posibilidad al profesor de generar distintas planificaciones a partir de encuestas sobre disponibilidad generadas a través de Doodle, como también le da la opción de realizar una replanificación si él lo desea. Además, en la sección 4.3 se muestran distintos experimentos y resultados realizados para analizar la sensibilidad del problema a través de la variación de sus parámetros. Finalmente, en la Sección 5 se presentan las conclusiones de trabajo realizado, algunas recomendaciones a tomar en consideración y el trabajo futuro.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

El Interrogation Scheduling Problem es un problema de asignación de bloques de interrogaciones para algún curso en particular. Un escenario de este problema se compone de los siguientes elementos:

- **Actores del Problema:** Profesor y alumnos.
- **Disponibilidad del Profesor:** Día y Horas en las cuales el profesor puede asistir a realizar las interrogaciones.
- **Disponibilidad de los alumnos:** Día y Horas en las cuales el alumno puede asistir a la interrogación.
- **Preferencia del Profesor:** Día y Horas en las cuales el profesor tiene como preferencia realizar las interrogaciones.
- **Preferencia del Alumno:** Día y Horas en las cuales el alumno tiene como preferencia asistir a la interrogación.

A continuación se presenta un ejemplo de una instancia en particular de este problema y lo que se esperaría como solución (Ver tabla 1):

- **Entrada:**
 - **Disponibilidad Profesor:** Lunes 09:00 a 11:30 hrs.
 - **Disponibilidad Alumno 1:** Lunes 09:00 y 09:30.
 - **Disponibilidad Alumno 2:** Lunes 09:30 y 10:00.
 - **Disponibilidad Alumno 3:** Lunes 09:00 y 09:30.
 - **Preferencia Alumno 1:** Lunes 09:30.
 - **Preferencia Alumno 2:** No aplica.
 - **Preferencia Alumno 3:** Lunes 09:00.

Tabla 1: Ejemplo de solución a la planificación de interrogaciones tomando en consideración las preferencias de los alumnos.

Fuente: Elaboración Propia.

Día Lunes	
09:00 - 09:30	Alumno 3
09:30 - 10:00	Alumno 1
10:00 - 10:30	Alumno 2
10:30 - 11:00	-
11:00 - 11:30	-

Como se puede ver en la tabla anterior, la solución del problema toma en consideración las preferencias de los alumnos, pero no siempre una instancia de algún problema va a poder satisfacer todas las preferencias que se presenten. Por otro lado, pueden existir varias soluciones al problema con lo cual quedaría a criterio del profesor escoger una planificación que le parezca mejor.

1.1. REPLANIFICACIÓN

Algunas veces ocurren emergencias a nivel personal de cada alumno o profesor que impactan en la planificación. Lo anterior significa que el profesor debe replanificar, siendo una tarea nuevamente no trivial y que dependiendo de la cantidad de bloques que se deban asignar nuevamente, agrega una nueva complejidad al profesor.

Tomando el mismo ejemplo anterior, si el **Alumno 1** avisa que no puede asistir a la interrogación asignada de **09:30 - 10:00** debido a que tiene cita al médico, se debe volver a asignar un bloque para que este alumno pueda asistir a interrogación sin problema, obviamente respetando su disponibilidad.

Para volver a asignar, significaría que el **Alumno 1** debe asistir solamente a la interrogación de las **09:00** por su disponibilidad. Lo anterior, implica que el **Alumno 3** ya no pueda asistir a ese bloque y deba ser asignado a las **09:30**, dejando como nueva solución la siguiente planificación (Ver tabla 2):

Tabla 2: Ejemplo de solución a la replanificación de interrogaciones.

Fuente: Elaboración Propia.

Día Lunes	
09:00 - 09:30	Alumno 1
09:30 - 10:00	Alumno 3
10:00 - 10:30	Alumno 2
10:30 - 11:00	-
11:00 - 11:30	-

Esta replanificación resuelve el problema presentado, pero no puede satisfacer las preferencias de los alumnos debido a la emergencia presentada.

1.2. ASIGNACIONES INTELIGENTES

Una nueva problemática que afecta tanto al profesor y a los alumnos son los atrasos que podrían ocurrir debido a que uno o varios alumnos se tomen más minutos de los destinados a la interrogación. Este atraso generaría atrasos en cadena los cuales terminarían perjudicando bastante la planificación e intentar resolver esta problemática realizando asignaciones de forma manual, dificulta mucho más conseguir un resultado el cual sea factible y también sea óptimo.

Por lo anterior, es necesario también que la planificación realizada siga un orden inteligente en donde dos bloques consecutivos de interrogación, asista primero un alumno que probablemente exceda el tiempo de interrogación, para que así el siguiente alumno que probablemente demore menos evite atrasar los siguientes bloques, ganando el tiempo perdido por la primera interrogación. A la asignación mencionada anteriormente se le denominará en este trabajo como **asignación inteligente**.

Una asignación inteligente se generaría comenzando con la valoración del profesor a los alumnos en términos del posible tiempo que podrían tardar en la interrogación:

- **Alumno 1:** Podría tardar menos
- **Alumno 2:** Sin valoración
- **Alumno 3:** Sin valoración
- **Alumno 4:** Podría tardar más

Tabla 3: Ejemplo de solución con la realización de asignaciones inteligentes.

Fuente: Elaboración Propia.

Día Lunes	
Bloque 1	Alumno 4
Bloque 2	Alumno 1
Bloque 3	Alumno 3
Bloque 4	Alumno 2

Como se puede apreciar en la Tabla 3, la solución posee una asignación inteligente debido que en los bloques 1 y 2, se asignó primero al *Alumno 4*, el cual podría tardarse más y atrasar la interrogación, para luego ganar el tiempo del posible atraso con la asignación al bloque 2 del *Alumno 1*, el cual es un alumno que podría tardar menos del tiempo de la interrogación.

1.3. USO DE DOODLE PARA PLANIFICAR REUNIONES

Doodle es un servicio de programación que permite a grupos de personas encontrar un horario de mutuo acuerdo. Una persona crea una encuesta de Doodle que se envía a los miembros del grupo. La encuesta pregunta a cada miembro del grupo para declarar si están disponibles para cada una de las opciones de franjas horarias elegidas originalmente por el iniciador de la encuesta (Ver Figura 1). Después de que todos los miembros hayan respondido a la encuesta, la persona que inició la encuesta puede determinar qué horario es el más apropiado para programar el evento. Esto es por lo general, el espacio que acomoda a la mayoría de los miembros del grupo [Romero *et al.*, 2017].

	jul 18 DOM 10:00 10:15	jul 18 DOM 10:15 10:30	jul 18 DOM 10:30 10:45	jul 18 DOM 10:45 11:00	jul 18 DOM 11:00 11:15	jul 18 DOM 11:15 11:30	jul 18 DOM 11:30 11:45	jul 18 DOM 11:45 12:00
6 participantes	✓5	✓6	✓4	✓5	✓4	✓3	✓4	✓6
Kevin Lagos Lavín	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sebato		✓		✓			✓	✓
Alumno 5	(✓)	✓	✓	✓	✓		✓	(✓)
Alumno 4	(✓)	(✓)	✓					✓
Alumno 3	✓	✓	(✓)	✓	(✓)	(✓)	✓	✓
Alumno 2	✓	✓	✓	(✓)	✓	(✓)	✓	(✓)
Alumno 1	(✓)	✓		✓	(✓)	✓		✓

Figura 1: Una encuesta de ejemplo realizada en Doodle.

Fuente: Elaboración Propia

Esta herramienta es bastante útil, ya que también permite exportar las respuestas a formato Excel (Ver Figura 2) y se podrían analizar las respuestas para obtener por ejemplo: Promedio de bloques elegidos por persona, bloques más escogidos, cantidad de preferencias por alumno, bloques menos escogidos, etc.

Poll "Prueba Horarios 3"
<https://doodle.com/poll/6cxzdhrz4xvevxbz>

July 2021								
Sun 18								
	10:00 AM – 10:15 AM	10:15 AM – 10:30 AM	10:30 AM – 10:45 AM	10:45 AM – 11:00 AM	11:00 AM – 11:15 AM	11:15 AM – 11:30 AM	11:30 AM – 11:45 AM	11:45 AM – 12:00 PM
Alumno 1	(OK)	OK		OK	(OK)	OK		OK
Alumno 2	OK	OK	OK	(OK)	OK	(OK)	OK	(OK)
Alumno 3	OK	OK	(OK)	OK	(OK)	(OK)	OK	OK
Alumno 4	(OK)	(OK)	OK					OK
Alumno 5	(OK)	OK	OK	OK	OK		OK	(OK)
sebato		OK		OK			OK	OK
Count	2:3:1	5:1:0	3:1:2	4:1:1	2:2:2	1:2:3	4:0:2	4:2:0

Figura 2: Encuesta de ejemplo exportada a formato Excel.
 Fuente: Elaboración Propia

1.4. SITUACIÓN ACTUAL EN ASIGNATURA DE INTELIGENCIA ARTIFICIAL UTFSM

Actualmente en la asignatura INF-295 llamada Inteligencia Artificial, se están realizando interrogaciones para evaluar los conocimientos de los alumnos en los distintos tópicos que se abordan durante el semestre. La planificación de las interrogaciones de esta asignatura en Casa Central se realiza a través de una encuesta que deben responder los alumnos vía Doodle, en donde solamente pueden marcar una sola casilla para poder asistir de todas las que estén disponibles y la cantidad de bloques por encuesta son bastante limitados, por lo que se deben realizar varias encuestas para que todos los alumnos puedan ser evaluados.

Otra situación que sucede es la existencia de atrasos en los días de interrogaciones debido a que algunos alumnos se exceden del tiempo límite de la duración de la interrogación, la cual es de aproximadamente de unos 30 minutos. Esto perjudica a los alumnos que les toca asistir a la interrogación en los siguientes bloques ya que deberán comenzar unos minutos más tarde debido a los atrasos que se generaron anteriormente.

1.5. USO DE DOODLE EN LA UTFSM

Se realizó una encuesta a 36 estudiantes de distintas carreras de la Universidad Técnica Federico Santa María, en la cual un 66,7 % respondió que sí ha utilizado Doodle alguna vez en la universidad, mientras que un 33,3 % respondió que nunca lo ha utilizado (Ver Figura 3).

¿Has utilizado Doodle alguna vez a lo largo de la carrera?

36 respuestas

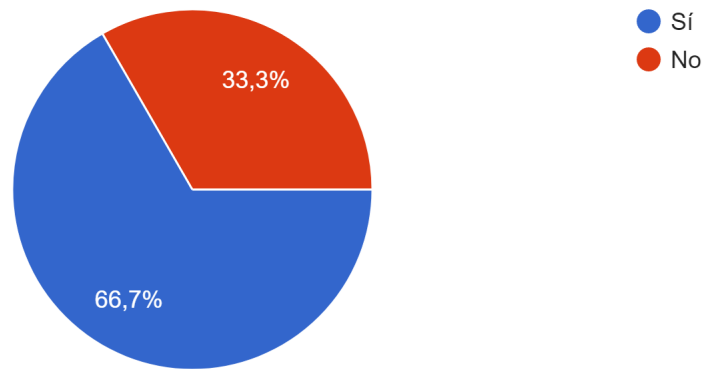


Figura 3: Gráfico de torta respecto a la utilización de Doodle en la universidad.
Fuente: Elaboración Propia

También se le preguntó a los estudiantes con qué fin han utilizado Doodle, obteniendo que un 84,6 % de los estudiantes que respondieron la encuesta dicen haber utilizado Doodle para la organización de ayuntadías, es decir, realizan la encuesta para analizar los horarios en el cual todos o gran parte del curso pueda asistir. Por otro lado, un 67,6 % de los estudiantes lo han utilizado para organizar evaluaciones, donde estas pueden ser evaluaciones fijadas por un ayudante o el profesor mismo (Ver Figura 4).



Figura 4: Histograma de motivos de utilización de Doodle en la universidad.
Fuente: Elaboración Propia

Se puede concluir, a partir de la información presentada anteriormente que la utilización de

esta herramienta es algo bastante común tanto en informática como en las otras carreras de la universidad, donde la mayor parte de los motivos de uso son para organizar y planificar hitos importantes. Sin embargo, esta herramienta es útil para escoger uno o varios bloques de horario donde la mayoría de las personas que respondieron la encuesta tienen disponibilidad. Lo que se busca solucionar en el **ISP** es encontrar la forma de asignar bloques de interrogación a múltiples alumnos, de tal manera que todos puedan asistir y en lo posible lograr satisfacer preferencias y la consideración de asignaciones inteligentes.

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL

Proponer un mecanismo que permita generar diferentes planificaciones para las interrogaciones de asignaturas teniendo en cuenta la disponibilidad y preferencia de horarios tanto de profesores como de alumnos.

1.6.2. OBJETIVOS ESPECÍFICOS

1. Analizar el estado del arte del problema a resolver y similares a éste.
2. Diseñar un prototipo de solución para generar distintas planificaciones de interrogaciones.
3. Implementación de una técnica de resolución que se base en conocimiento.
4. Evaluar los resultados obtenidos para validar la propuesta realizada.

1.7. RESUMEN

En este capítulo se presentó la definición del Interrogation Scheduling Problem, junto con sus características y las dificultades que se presentan. Además, se presentó un ejemplo simple para ilustrar la estructura del problema, se abordó el problema de la replanificación y el deseo de generar una asignación inteligente de bloques con el fin de evitar posibles atrasos y se contextualizó el caso específico de la asignatura de Inteligencia Artificial en la UTFSM junto con datos que muestran la utilización de Doodle a nivel general en la universidad.

Finalmente, se presentaron los objetivos de esta memoria, tanto el objetivo general como los objetivos específicos para lograr comprender el enfoque de este trabajo de título.

En el siguiente capítulo se analizará el estado del arte del ISP junto con problemas similares a éste. Además, se presentarán distintas técnicas que generalmente se utilizan para poder resolver este tipo de problemas.

CAPÍTULO 2

ESTADO DEL ARTE

Los problemas tipo de asignación con recursos limitados tienen numerosas formas y variaciones en la vida real. Es de bastante interés investigar sobre la complejidad del problema presentado y también sobre problemas relacionados con el fin de tener una base en la cual la literatura ha logrado proponer algoritmos y encontrado soluciones de calidad.

2.1. PROBLEMAS RELACIONADOS

Dentro del mundo de los problemas de asignación de un conjunto limitado de recursos, existen distintas variantes las cuales se describen a continuación:

1. **University Course Timetabling** o asignación horaria universitaria, la cual corresponde a la asignación de clases en universidades.
2. **School Timetabling** o asignación horaria escolar, la cual corresponde a la asignación de clases en colegios.
3. **Exam Timetabling** o asignación horaria de exámenes, la cual corresponde a asignación de exámenes en universidades.

Este tipo de problemas son bastante cercanos al **Interrogation Scheduling Problem**, debido a que se centran en un contexto escolar y lo que varían son los actores del problema, el contexto y los tipos de recursos que existen.

2.1.1. EXAM TIMETABLING

Se presenta este problema ya que es de interés investigar sobre las distintas técnicas que se han utilizado hasta ahora para lograr resolver este problema y encontrar ciertos patrones y algoritmos que puedan ser aplicados al ISP.

El Exam Timetabling es un problema que busca asignar exámenes y salas de rendición en ciertos periodos establecidos por alguna institución respetando restricciones [Ayob *et al.*, 2007]. Este problema es altamente complejo debido a las limitaciones de recursos en términos de lugares para rendir exámenes y franjas horarias, teniendo en consideración que algunas instituciones educacionales presentan más reglas como limitar el número de exámenes en un día para cada estudiante. Al igual que el tema en este trabajo, algunas instituciones resuelven generalmente este problema manualmente con el método ensayo y error [Abou Kasm *et al.*, 2019].

En los últimos años se han realizado bastantes investigaciones, donde por ejemplo se han estudiado ampliamente enfoques meta-heurísticos para resolver este problema como Tabú Search, Simulated Annealing, Genetic Algorithms, Memetic Algorithms, Ant Algorithms y Multi-objective Evolutionary Algorithms [McCollum *et al.*, 2012].

Un algoritmo utilizado en [Müller, 2016] explica la fase de construcción de la solución: Se obtiene una solución completa utilizando un algoritmo denominado Iterative Forward Search [Müller, 2005]. Este algoritmo hace uso de estadísticas basadas en conflicto (CBS) [Müller *et al.*, 2004] para prevenir los ciclos. En la siguiente fase, un óptimo local se encuentra utilizando el algoritmo Hill Climbing. Una vez que no se pueda mejorar esta solución encontrada con el algoritmo anterior, se utiliza la técnica denominada Great Deluge [Dueck, 1993]. Esta técnica anterior es alterada de tal manera que permita algunas oscilaciones del límite que se impone en el conjunto donde vive la solución. Esta búsqueda termina después de que se cumpla un tiempo límite predeterminado y se retorna la mejor solución encontrada.

2.2. COMPLEJIDAD DEL PROBLEMA

La complejidad del problema presentado es alta, ya que el número de combinaciones para encontrar una solución factible es bastante grande, teniendo en consideración un problema con n alumnos y n bloques, existen $n!$ posibles asignaciones a realizar. Por otro lado, el problema se vuelve más complejo considerando que se incluirán preferencias de profesores y alumnos con posibilidad de generarse cambios en casos de emergencia o si el profesor lo desea. Debido a las características del problema, este se puede ver como un CSP (Problema de satisfacción de restricciones) solamente para el problema de asignación de bloques, sin embargo, el hecho de agregar preferencias y asignaciones inteligentes, termina convirtiendo este problema en un CSOP (Problema de satisfacción y optimización de restricciones). En ambos problemas existen bastantes técnicas para resolverlos, como por ejemplo técnicas completas o incompletas de inteligencia artificial.

2.3. PROBLEMA DE SATISFACCIÓN Y OPTIMIZACIÓN DE RESTRICCIONES

Un problema de satisfacción de restricciones consiste en un conjunto de variables, cada una definida en un conjunto discreto y finito de valores (también llamado dominio) y un conjunto de restricciones que restringe los valores que las variables pueden tomar. La solución a un CSP es la asignación de valores a variables de su dominio de tal manera que todas las restricciones del problema son satisfechas [Bidar *et al.*, 2018].

En comparación con el CSP que considera todas las soluciones potenciales como buenas soluciones, los CSOPs exploran todas esas soluciones, evaluando sus respectivos costos en relación de una función objetivo la cual evalúa la calidad de la solución

[Ghédira y Dubuisson, 2013]. Formalmente un CSOP es un conjunto de cinco elementos (X, D, C, R, f) definido por:

- Un conjunto de n variables: $X = \{X_1, X_2, \dots, X_n\}$.
- Un conjunto de n dominios finitos y discretos: $D = \{D_1, D_2, \dots, D_n\}$ donde D_i es el conjunto de valores asociados a la variable X_i .
- Un conjunto de m restricciones: $C = \{C_1, C_2, \dots, C_m\}$ donde cada restricción C_i concierne un subconjunto de variables.
- Un conjunto de m relaciones: $R : \{R_1, R_2, \dots, R_m\}$ donde cada relación R_i está definida por un subconjunto del producto cartesiano $D_{i1} \times D_{i2} \times \dots \times D_{ik}$ correspondiente al conjunto de posibles combinaciones de valores para C_i , donde k es el número de variables involucradas en C_i .
- Una función de costo o función objetivo definida por:
 $f : S_p \rightarrow$ valor numérico donde S_p es el conjunto de soluciones potenciales.
 Por lo tanto, un CSOP (X, D, C, R, f) es un CSP (X, D, C, R) que busca instancias completas y consistentes optimizando una función de costo f .

Debido a que casi todos los problemas en el mundo real poseen restricciones que restringen el número de soluciones aceptables, enfrentar estos problemas de una manera adecuada ha sido un centro de consideración. Existen bastantes ejemplos de CSPs incluyendo: Problemas de coloreo de grafos, Problema de las N-Reinas y también aplicaciones en el mundo real como planificación, configuración, programación, etc.

2.4. MÉTODOS DE RESOLUCIÓN

Los problemas de satisfacción de restricciones son conocidos por ser problemas NP-Completos, lo que significa que resolverlos con métodos clásicos como una búsqueda sistemática requiere un costo de tiempo exponencial. Teniendo en consideración un problema con n variables y un dominio de tamaño d , la resolución de este problema tendría complejidad en el peor caso $O(d^n)$ con un algoritmo de búsqueda exhaustiva como backtracking.

Los métodos de resolución de estos problemas se pueden clasificar en dos grandes categorías: métodos exactos que garantizan la integridad de la resolución pero requieren bastante complejidad temporal, y por otro lado están los métodos aproximados que sacrifican la integridad para ganar en complejidad temporal.

- **Métodos completos:** A pesar de la calidad óptima de sus resultados, estos enfoques generalmente encuentran dificultades con aplicaciones de un tamaño significativo. En

esta categoría, se discuten los métodos más tradicionales: branch and bound y branch and cut.

- **Métodos incompletos:** Estos métodos se utilizan generalmente en problemas de optimización a gran escala en los que se requiere una resolución rápida y la optimización no es de primordial importancia.

2.4.1. BRANCH AND BOUND

Uno de los algoritmos más utilizados para encontrar soluciones óptimas en problemas de optimización discreta es llamado *Branch and Bound* (B&B). Este algoritmo necesita una heurística que mapea el etiquetado parcial a un valor numérico. Representa una subestimación (en caso de minimización) de la función objetivo para el mejor etiquetado completo obtenido de un etiquetado parcial.

El algoritmo busca soluciones en profundidad primero y se comporta como backtracking cronológico a excepción de que tan pronto como se asigna un valor a la variable, se calcula el valor de la función heurística para el etiquetado. Si este valor excede el límite, entonces el subárbol bajo el etiquetado parcial actual se poda inmediatamente. Inicialmente, el límite se establece en infinito y durante el cálculo registra el valor de la mejor solución encontrada hasta el momento.

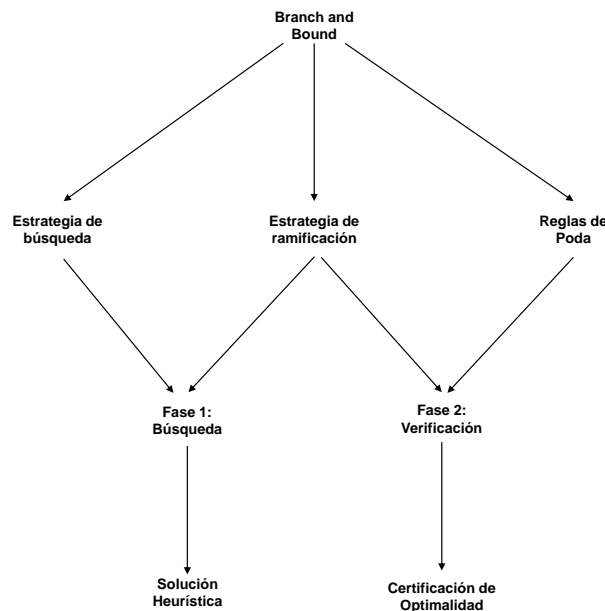


Figura 5: Diagrama de las tres componentes principales de Branch & Bound.

Fuente: [Morrison *et al.*, 2016]

La eficiencia de B&B está determinada por dos factores: la calidad de la función heurística

y si se encuentra pronto un buen límite. Las observaciones de problemas de la vida real muestran también que el último paso hacia el óptimo, es decir, mejorar aún más una buena solución, suele ser la parte más costosa desde el punto de vista computacional del proceso de resolución. Afortunadamente, en muchas aplicaciones, los usuarios están satisfechos con una solución cercana a la óptima si esta solución se encuentra pronto. El algoritmo de branch and bound también se puede utilizar para encontrar soluciones subóptimas mediante el uso del segundo límite de aceptabilidad. Si el algoritmo encuentra una solución que es mejor que el límite de aceptabilidad, esta solución puede devolverse al usuario incluso si no se demuestra que sea óptima [Barták, 1999].

2.4.2. BRANCH AND CUT

Este método es muy exitoso y eficiente para resolver una variedad de problemas de programación lineal entera y al mismo tiempo ofrece una garantía de optimización. Muchos problemas involucran variables que no son continuas sino que tienen valores enteros o mezclan variables enteras y continuas y pueden resolverse mediante el método de *Branch and Cut*. Este método combina B&B y el método de planos cortantes. El principio de este último método es agregar nuevas restricciones para obtener soluciones completas. De una solución óptima x_{opt} , las restricciones, también llamadas planos cortantes, se agregan de tal manera que la solución óptima entera es preservada, pero x_{opt} viola al menos una de esas nuevas restricciones [Naud *et al.*, 2020]. Estos planos cortantes se aplican hasta que no se consigue encontrar ningún plano que afecte a la solución. Finalmente, se utiliza *Branch and Bound* en el espacio resultante para hallar la solución óptima.

2.5. ALGORITMOS INCOMPLETOS

Existen diversos algoritmos de búsqueda local para resolver los problemas tipo Timetabling. El término de búsqueda local expresa la idea de algoritmos que modifican una asignación de manera local para moverse a una mejor asignación. Durante cada iteración, se busca realizar movimientos a la solución dentro del vecindario de la solución actual.

2.5.1. ALGORITMO HILL-CLIMBING

Este algoritmo siempre selecciona la mejor asignación con respecto a la actual desde un vecindario. Esto significa que para un CSP, se busca la asignación que minimiza el número de restricciones que no se cumplan. En el caso de problemas de optimización, selecciona un vecino que mejora el valor de la función de evaluación.

Cuando no existe mejor asignación que la actual, la búsqueda se estanca en un óptimo local.

Este algoritmo usualmente reinicia la búsqueda desde otra solución inicial, muchas veces de manera aleatoria.

Algorithm 1 Pseudocódigo del algoritmo Hill-Climbing

```
1: procedure Hill-Climbing
2:    $L \leftarrow \text{NEIGHBORS}(\text{currentNode})$ 
3:   nextEval  $\leftarrow -\infty$ 
4:   nextNode  $\leftarrow \text{NULL}$ 
5:   for todo  $x$  en  $L$  do
6:     if  $\text{EVAL}(x) > \text{nextEval}$  then
7:       nextNode  $\leftarrow x$ 
8:       nextEval  $\leftarrow \text{EVAL}(x)$ 
9:     end if
10:  end for
11:  if nextEval  $\leq \text{EVAL}(\text{currentNode})$  then
12:    return currentNode
13:  end if
14:  currentNode  $\leftarrow \text{nextNode}$ 
15: end procedure
```

2.5.2. ALGORITMO TABÚ SEARCH

La búsqueda tabú es otro método para prevenir ciclos y no quedarse estancados en un óptimo local. Se basa en la noción de una lista tabú, la cual es un término especial de memoria corta, que mantiene un historial selectivo compuesto de configuraciones anteriores o de atributos de estas configuraciones.

La búsqueda tabú utiliza un procedimiento de búsqueda local o por vecindades para moverse iterativamente desde una solución x hacia otra solución x^* en el vecindario de x , hasta satisfacer algún criterio de parada. Para realizar la exploración del espacio de búsqueda, este algoritmo modifica la estructura de vecinos para cada solución a medida que la búsqueda progresa. Las soluciones admitidas para el nuevo vecindario, son determinadas mediante el uso de una lista tabú. La búsqueda entonces progresa moviéndose iterativamente de una solución x hacia una solución x^* en el nuevo vecindario.

Para evitar quedar estancado en ciclos, la lista tabú al ir almacenando los últimos movimientos que se aplican a la solución ayuda a que estos nuevos movimientos no se puedan repetir durante un tiempo determinado.

Algorithm 2 Pseudocódigo del algoritmo Tabú Search

```
1: procedure Tabú Search
2:   mejorS  $\leftarrow$  s0
3:   mejorCandidato  $\leftarrow$  s0
4:   listaTabu  $\leftarrow$  [s0]
5:   while no se cumple el criterio de parada do
6:     sVecindario  $\leftarrow$  obtenerVecinos(mejorCandidato)
7:     mejorCandidato  $\leftarrow$  sVecindario[0]
8:     for sCandidato en sVecindario do
9:       if candidato not in listaTabú y fit(sCandidato) >fit(mejorCandidato) then
10:        mejorCandidato  $\leftarrow$  sCandidato
11:      end if
12:    end for
13:    if fit(mejorCandidato) >fit(mejorS) then
14:      sBest  $\leftarrow$  mejorCandidato
15:    end if
16:    Agregar a listaTabú mejorCandidato
17:    if tamaño de listaTabú >máximoTamaño then
18:      Remover primer elemento de listaTabú
19:    end if
20:  end while
21:  return mejorS
22: end procedure
```

2.6. RESUMEN

En este capítulo se analizó la complejidad del problema y se nombraron y explicaron las maneras tradicionales de resolución de estos problemas y métodos de optimización clásicos completos, como el conocido Branch and Bound y Branch and Cut y también métodos incompletos como Hill-Climbing y Tabú Search.

Se explicó el problema Exam Timetabling por su similitud con el problema de esta memoria, donde se explica que en los últimos años se han realizado bastantes investigaciones donde se han estudiado enfoques metaheurísticos para la resolución del problema, sin embargo, muchas instituciones siguen resolviendo el problema con el método ensayo y error.

En el siguiente capítulo, se realiza una definición de la propuesta de solución, sus fundamentos y la metodología aplicada para implementar la solución.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

Como se mencionó en la sección , el ISP es un problema bastante complejo de resolver manualmente, más aún considerando las posibles emergencias que necesitan generar una re-planificación, complicando aún más el problema. La propuesta de solución de esta memoria se compone de dos partes:

1. Plantear dos modelos matemáticos los cuales modelarán dos escenarios:
 - a) La generación de planificaciones considerando preferencias y asignaciones inteligentes.
 - b) La replanificación de horarios.
2. Implementar un software el cual haga uso de estos modelos matemáticos para resolver instancias del ISP con la técnica completa denominada Branch and Cut.

Realizando lo anterior, se podrán realizar distintas investigaciones y análisis del problema más a fondo en un futuro, logrando por ejemplo la posible identificación de nuevas variantes de este problema de acuerdo a la necesidad de cada profesor o universidad, siendo un aporte a la comunidad científica y a la sociedad.

3.1. ASIGNACIÓN DE BLOQUES CONSIDERANDO PREFERENCIAS DE PROFESORES Y ALUMNOS

3.1.1. CONJUNTOS

\mathcal{I} conjunto de alumnos, $i \in 1, \dots, n$

\mathcal{J} conjunto de bloques, $j \in 1, \dots, m$

3.1.2. PARÁMETROS

$$d_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ posee disponibilidad en el bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$
$$p_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ posee preferencia en asistir al bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$
$$t_i = \begin{cases} 2 & \text{si el profesor cree que el alumno } i \text{ se podría tardar más en la interrogación} \\ 1 & \text{si el profesor cree que el alumno } i \text{ se podría tardar menos en la interrogación} \\ 0 & \text{si no posee valoración} \end{cases}$$

3.1.3. VARIABLE DE DECISIÓN

$$x_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ es asignado al bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$

3.1.4. VARIABLE AUXILIAR

$$y_j = \begin{cases} 1 & \text{si los bloques } j \text{ y } j + 1 \text{ son una asignación inteligente} \\ 0, -1, -2 & \text{en caso contrario} \end{cases}$$

3.1.5. MODELO MATEMÁTICO

$$\begin{aligned}
 \text{máx} \quad & \sum_{i=1}^n \sum_{j=1}^m p_{ij} \cdot x_{ij} + \sum_{j \text{ impar}}^m y_j \\
 \text{s.t.} \quad & \sum_{i=1}^n x_{ij} \leq 1 \quad \forall j \in \mathcal{J} \\
 & \sum_{j=1}^m x_{ij} = 1 \quad \forall i \in \mathcal{I} \\
 & x_{ij} \leq d_{ij} \quad \forall i \in \mathcal{I} \quad \forall j \in \mathcal{J} \\
 & \sum_{i=1}^n t_i \cdot x_{ij} - \sum_{i=1}^n t_i \cdot x_{i(j+1)} - y_j = 0 \quad \forall j \text{ impar} \in \mathcal{J} \\
 & y_j \leq 1 \quad \forall j \text{ impar} \in \mathcal{J}
 \end{aligned}$$

3.1.6. FUNCIÓN OBJETIVO

La función objetivo de este modelo busca maximizar la cantidad de preferencias satisfechas de alumnos a bloques y la cantidad de asignaciones inteligentes entre bloques consecutivos.

$$\text{máx} \underbrace{\sum_{i=1}^n \sum_{j=1}^m p_{ij} \cdot x_{ij}}_{\text{Preferencias satisfechas}} + \underbrace{\sum_{j \text{ impar}}^m y_j}_{\text{Asignaciones inteligentes}}$$

3.1.7. RESTRICCIONES

1. La primera restricción impone que en cada bloque j puede asistir a lo más un alumno.
2. La segunda restricción impone que cada alumno i debe ser asignado a un solo bloque.
3. La tercera restricción impone que la asignación de un alumno i a un bloque j solo se puede realizar si el alumno posee disponibilidad en ese bloque.
4. La cuarta restricción busca asignar valores a la variable y_j . La primera y segunda sumatoria obtienen el valor que le dio el profesor al alumno asignado al bloque j y $j + 1$ respectivamente y la resta de ambos valores se le asigna a la variable y_j . Esta restricción es auxiliar debido que la asignación de valores resultantes de la resta a la variable y_j busca ser maximizada en la función objetivo, y al mismo tiempo intenta moldear a que la asignación entre bloques consecutivos, en el mejor de los casos, sea una asignación inteligente (con un valor igual a 1).

5. La última restricción, si bien, se menciona que a lo más esta variable puede tomar el valor 1, por lo que sería redundante, se necesita ya que podría ocurrir que se asigne un alumno con valoración 2 y luego un alumno sin valoración, por lo que se intentaría maximizar este tipo de asignaciones. Para evitar lo anterior, es necesario restringir el valor de esta variable a lo más 1, que sería el caso de una asignación inteligente.

3.2. CAMBIOS EN DISPONIBILIDAD DE HORARIOS

Una problemática bastante común que ocurre al momento de realizar planificaciones de interrogaciones, es el hecho de que se necesite reasignar a un alumno a un bloque diferente al cual fue asignado, debido a alguna emergencia la cual le impida poder asistir en ese horario. Debido a lo anterior, resulta bastante complejo poder realizar una reasignación de este alumno cuando la cantidad de bloques es igual a la cantidad de alumnos, ya que obliga al profesor a intercambiar asignaciones entre alumnos y nuevamente este proceso es no trivial cuando la disponibilidad de los alumnos es bastante baja.

Debido a lo anterior, es necesario plantear una nueva manera de solucionar esta problemática a través de un modelo que **minimice** los cambios de la planificación anteriormente realizada, de tal manera que se perjudique lo menos posible tanto al profesor, como a los alumnos en los horarios que ya fueron asignados.

3.2.1. CONJUNTOS

\mathcal{I} conjunto de alumnos, $i \in 1, \dots, n$
 \mathcal{J} conjunto de bloques, $j \in 1, \dots, m$

3.2.2. PARÁMETROS

$$d_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ posee disponibilidad en el bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$
$$s_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ fue asignado anteriormente al bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$

3.2.3. VARIABLE DE DECISIÓN

$$x_{ij} = \begin{cases} 1 & \text{si el alumno } i \text{ es asignado al bloque } j \\ 0 & \text{en caso contrario} \end{cases}$$

3.2.4. MODELO MATEMÁTICO

$$\begin{aligned} \text{máx} \quad & \sum_{i=1}^n \sum_{j=1}^m s_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^n x_{ij} \leq 1 \quad \forall j \in \mathcal{J} \\ & \sum_{j=1}^m x_{ij} = 1 \quad \forall i \in \mathcal{I} \\ & x_{ij} \leq d_{ij} \quad \forall i \in \mathcal{I} \quad \forall j \in \mathcal{J} \end{aligned}$$

3.2.5. FUNCIÓN OBJETIVO

Debido a que se busca minimizar las perturbaciones del escenario de asignaciones, la función objetivo maximiza la igualdad entre la planificación anterior y la nueva, realizando el producto entre las asignaciones obtenidas antes y las asignaciones obtenidas después del cambio de disponibilidad, donde este producto es 1 solamente si la asignación del alumno i al bloque j se respetó en la nueva solución, es decir:

$$s_{ij} \cdot x_{ij} = \begin{cases} 1 & \text{si } s_{ij} = x_{ij} = 1 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

3.3. PLANIFICACIÓN PARA ASIGNATURA DE INTELIGENCIA ARTIFICIAL UTFSM

Para resolver la problemática presentada en la asignatura de Inteligencia Artificial en la sección 1.4, se realizará la creación de un programa el cual deberá recibir para el caso de la planificación el archivo excel descargado directamente desde Doodle con la disponibilidad

de los alumnos y también una planificación generada anteriormente para el caso de la replanificación, generando como output distintas planificaciones de interrogaciones, de las cuales el profesor podrá escoger cuál es la más apropiada según su criterio.

3.3.1. HERRAMIENTAS A UTILIZAR

Este programa se crea en lenguaje Python, con la utilización principalmente de la librería *TKINTER* para la creación de la interfaz gráfica y *PYTHON-MIP*, donde esta última corresponde a un conjunto de herramientas las cuales ayudan a resolver problemas de programación lineal mixta, utilizando como método de resolución *Branch and Cut*.

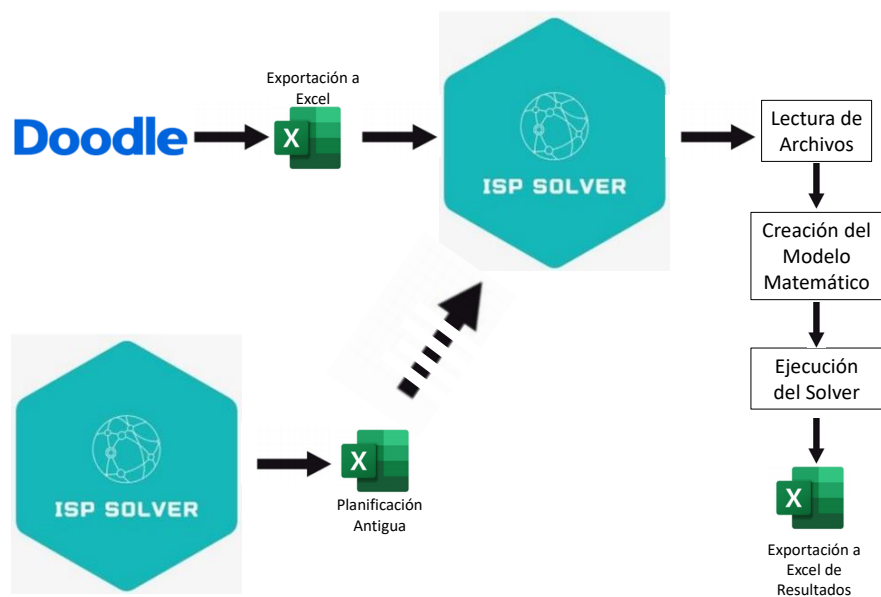


Figura 6: Flujo general del programa, la línea punteada representa un input opcional que se utiliza para el proceso de replanificación.

Fuente: Elaboración Propia

3.4. CONCLUSIONES

En esta sección se presentaron los modelos matemáticos que modelan el *Interrogation Scheduling Problem*, tanto para los casos de planificación con satisfacción de preferencias y asignaciones inteligentes, como también para los casos de replanificación. El primer modelo busca la maximización de preferencias satisfechas, a través del producto entre la asignaciones de alumnos a bloques con las preferencias. Además, maximiza la cantidad de asignaciones inteligentes a través de una variable auxiliar, la cual asigna un valor numérico a partir de

la resta de la valoración de dos alumnos asignados entre bloques consecutivos. El segundo modelo, busca la maximización de igualdad entre bloques asignados en una planificación generada previamente y la nueva replanificación que se busca generar, es decir, minimiza las posibles perturbaciones que podrían ocurrir para encontrar una solución.

Finalmente, se propone la creación de un programa para la asignatura de Inteligencia Artificial de la UTFSM, el cual servirá de apoyo para resolver la problemática presentada en este trabajo de título, dando distintas opciones de planificaciones al profesor.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

4.1. IMPLEMENTACIÓN DEL ISP SOLVER PARA INTELIGENCIA ARTIFICIAL

A continuación se mostrarán capturas de pantalla del programa realizado para la asignatura de Inteligencia Artificial y se mostrarán los distintos resultados generados a partir de una instancia de prueba.

4.1.1. MENÚ PRINCIPAL

En el menú principal (Ver Figura 7), el profesor deberá subir el archivo excel correspondiente a la encuesta que realizó en Doodle, en la cual se indicarán las disponibilidades y preferencias del alumno. Por otro lado, también se le da la opción de subir una planificación antigua (que debe haber sido generada por el mismo programa anteriormente), con la cual podrá realizar una replanificación en casos de cambio de disponibilidad de uno o varios alumnos.



Figura 7: Captura de Menú Principal del programa.
Fuente: Elaboración Propia

4.1.2. PLANIFICACIÓN DE UN NUEVO HORARIO

Si el profesor desea generar una nueva planificación, deberá subir el archivo excel del Doodle y luego presionar sobre el botón **Generar Planificación**, sobre el cual se abrirá una nueva ventana donde deberá realizar la asignación de valores a cada alumno que representan la valoración de cada uno de ellos (Ver Figura 8). Estos valores representan al vector T que se ha mencionado anteriormente.

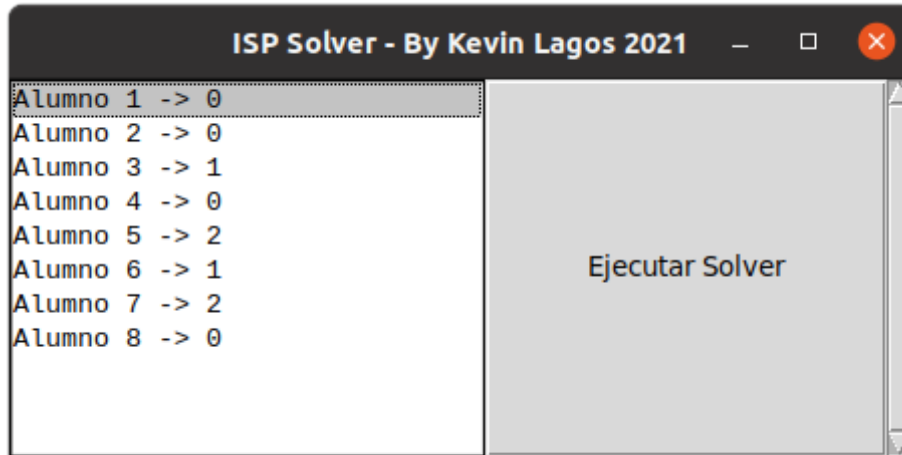


Figura 8: Captura de sección para agregar valoraciones a alumnos.
Fuente: Elaboración Propia

Una vez que el profesor haya ingresado las valoraciones, deberá pulsar el botón **Ejecutar Solver**, el cual desencadenará el proceso de resolución del problema para luego abrir una nueva ventana y mostrar el mejor resultado obtenido que fue evaluado a través de la función de evaluación (Ver Figura 9). De manera opcional el profesor puede exportar este resultado a un archivo excel, en el cual además de mostrarle el mejor resultado, también le mostrará como máximo 10 posibles planificaciones de interrogaciones, ordenadas de manera decreciente en términos de la calidad de la solución.



The screenshot shows a window titled "Resultados Planificación" with a table of results. Below the table is a text input field for the filename and an "Exportar a Excel" button.

Bloque	Nombre Alumno
18/07 10:00 - 10:15	Alumno 2
18/07 10:15 - 10:30	Alumno 8
18/07 10:30 - 10:45	Alumno 4
18/07 10:45 - 11:00	Alumno 1
18/07 11:00 - 11:15	Alumno 5
18/07 11:15 - 11:30	Alumno 3
18/07 11:30 - 11:45	Alumno 7
18/07 11:45 - 12:00	Alumno 6

Nombre archivo a generar

planificacion18-07 .xls

Exportar a Excel

Figura 9: Captura de sección que muestra la mejor planificación generada.
Fuente: Elaboración Propia

4.1.3. EJECUCIÓN DE INSTANCIA DE PRUEBA EN ISP SOLVER

Se presenta una instancia de prueba generada directamente desde Doodle, la cual se muestra a continuación (Ver Figura 10).

Encuesta "Prueba Horarios 3"
<https://doodle.com/poll/6cxzdhrz4xvevxbz>

julio 2021								
dom 18								
	10:00 – 10:15	10:15 – 10:30	10:30 – 10:45	10:45 – 11:00	11:00 – 11:15	11:15 – 11:30	11:30 – 11:45	11:45 – 12:00
Alumno 1	(OK)	OK			(OK)	OK		OK
Alumno 2	OK	OK	OK	(OK)	OK	(OK)	OK	(OK)
Alumno 3	OK	OK	(OK)	OK	(OK)	(OK)	OK	OK
Alumno 4	(OK)	(OK)	OK					OK
Alumno 5	(OK)	OK	OK	OK	OK		OK	(OK)
Alumno 6	(OK)	(OK)	OK					OK
Alumno 7		OK		OK			OK	OK
Alumno 8		OK		OK			OK	OK
Contador	2:4:1	5:2:0	4:1:2	4:1:2	2:2:3	1:2:4	4:0:3	5:2:0

Figura 10: Horarios de prueba para la generación de planificaciones.
 Fuente: Elaboración Propia

Este archivo se ingresa como input al *ISP Solver*, junto con las valoraciones mostradas en la Figura 8, donde los *Alumnos 5 y 7* se valoraron como alumnos que podrían excederse del tiempo, mientras que los *Alumnos 3 y 6* se valoraron como alumnos que podrían tardar menos en la interrogación. A partir de los inputs anteriormente mencionados se obtuvo como mejor resultado la siguiente planificación (Ver Figura 11):

Planificación de Interrogaciones - 20/08/2021 21:25:40		
Bloques	Alumnos	
18/07 10:00 – 10:15	Alumno 2	
18/07 10:15 – 10:30	Alumno 8	
18/07 10:30 – 10:45	Alumno 4	
18/07 10:45 – 11:00	Alumno 1	
18/07 11:00 – 11:15	Alumno 5	
18/07 11:15 – 11:30	Alumno 3	
18/07 11:30 – 11:45	Alumno 7	
18/07 11:45 – 12:00	Alumno 6	

Figura 11: Mejor planificación obtenida con el *ISP Solver*
 Fuente: Elaboración Propia

Como se puede apreciar, se realizaron las dos asignaciones inteligentes esperadas de este problema, donde los *Alumnos 5 y 3* fueron asignados consecutivamente al igual que los *Alumnos 7 y 6*. Por otro lado, las asignaciones realizadas lograron satisfacer las preferencias de 7 alumnos de un total de 8, por lo cual es un muy buen resultado a considerar.

A continuación se muestra las demás planificaciones obtenidas por el *ISP Solver*:

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 1		
18/07 10:15 – 10:30	Alumno 2		
18/07 10:30 – 10:45	Alumno 4		
18/07 10:45 – 11:00	Alumno 8		
18/07 11:00 – 11:15	Alumno 5		
18/07 11:15 – 11:30	Alumno 3		
18/07 11:30 – 11:45	Alumno 7		
18/07 11:45 – 12:00	Alumno 6		

Figura 12: Segunda planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 12 muestra la segunda solución obtenida, la cual realiza dos asignaciones inteligentes y satisface las preferencias de 6 alumnos.

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 5		
18/07 10:15 – 10:30	Alumno 7		
18/07 10:30 – 10:45	Alumno 6		
18/07 10:45 – 11:00	Alumno 3		
18/07 11:00 – 11:15	Alumno 1		
18/07 11:15 – 11:30	Alumno 2		
18/07 11:30 – 11:45	Alumno 8		
18/07 11:45 – 12:00	Alumno 4		

Figura 13: Tercera planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 13, muestra la tercera planificación obtenida, la cual posee una asignación inteligente en el segundo y tercer bloque y además satisface las preferencias de 5 alumnos.

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 4		
18/07 10:15 – 10:30	Alumno 8		
18/07 10:30 – 10:45	Alumno 6		
18/07 10:45 – 11:00	Alumno 3		
18/07 11:00 – 11:15	Alumno 1		
18/07 11:15 – 11:30	Alumno 2		
18/07 11:30 – 11:45	Alumno 7		
18/07 11:45 – 12:00	Alumno 5		

Figura 14: Cuarta planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 14 muestra la cuarta planificación obtenida, la cual no posee ninguna asignación inteligente, pero satisface las preferencias de 4 alumnos.

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 6		
18/07 10:15 – 10:30	Alumno 5		
18/07 10:30 – 10:45	Alumno 4		
18/07 10:45 – 11:00	Alumno 8		
18/07 11:00 – 11:15	Alumno 1		
18/07 11:15 – 11:30	Alumno 2		
18/07 11:30 – 11:45	Alumno 3		
18/07 11:45 – 12:00	Alumno 7		

Figura 15: Quinta planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 15 muestra la quinta planificación, la cual tampoco posee asignaciones inteligentes, pero logra satisfacer la preferencia de 5 alumnos.

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 6		
18/07 10:15 – 10:30	Alumno 5		
18/07 10:30 – 10:45	Alumno 3		
18/07 10:45 – 11:00	Alumno 7		
18/07 11:00 – 11:15	Alumno 1		
18/07 11:15 – 11:30	Alumno 2		
18/07 11:30 – 11:45	Alumno 8		
18/07 11:45 – 12:00	Alumno 4		

Figura 16: Sexta planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 16 muestra la sexta planificación obtenida, la cual posee una asignación inteligente entre el segundo y tercer bloque, satisfaciendo las preferencias de 4 alumnos.

Planificación de Interrogaciones - 20/08/2021 21:25:40			
Bloques	Alumnos		
18/07 10:00 – 10:15	Alumno 5		
18/07 10:15 – 10:30	Alumno 6		
18/07 10:30 – 10:45	Alumno 3		
18/07 10:45 – 11:00	Alumno 7		
18/07 11:00 – 11:15	Alumno 1		
18/07 11:15 – 11:30	Alumno 2		
18/07 11:30 – 11:45	Alumno 8		
18/07 11:45 – 12:00	Alumno 4		

Figura 17: Séptima y última planificación obtenida por el ISP Solver.
Fuente: Elaboración Propia

La figura 17 muestra la séptima y última planificación obtenida, la cual posee una sola asignación inteligente, pero logra satisfacer las preferencias solamente de 3 alumnos.

Como se puede ver, tanto la primera y la segunda planificación logran satisfacer ambas asignaciones inteligentes, donde solamente en la segunda varía la cantidad de preferencias satisfechas, disminuyendo su valor de 7 a 6 preferencias totales satisfechas.

De la tercera planificación en adelante, la cantidad de asignaciones inteligentes se reduce, realizando para las planificaciones 3,6 y 7 una sola asignación inteligente (Ver Figuras 13, 16 y 17) y ninguna asignación inteligente para las planificaciones 4 y 5 (Ver Figuras 14 y 15).

4.1.4. REPLANIFICACIÓN DE HORARIO

Para realizar la replanificación, el profesor deberá subir en el apartado **Subir Doodle**, el excel con las nuevas disponibilidades de los alumnos, y también deberá subir en el apartado **Subir Planificación Antigua**, el archivo excel con **una sola planificación** exportada anteriormente desde el *ISP Solver*. Para obtener los resultados de la replanificación directamente debe pulsar el botón que dice **Replanificar**.

A continuación se muestran los resultados al realizar un cambio de disponibilidad en el *Alumno 1*, es decir, que el alumno ya no pueda asistir en el bloque que se le fue asignado previamente en la planificación anterior, generándose como nuevo excel de disponibilidad el presentado en la Figura 18 y la mejor planificación generada anteriormente (Ver Figura 11). Tomando ambos archivos excel como input al programa, se obtuvo como resultado la siguiente replanificación:


Encuesta "Prueba Horarios 3"
<https://doodle.com/poll/6cxzdhrz4xvevxbz>

julio 2021								
dom 18								
	10:00 – 10:15	10:15 – 10:30	10:30 – 10:45	10:45 – 11:00	11:00 – 11:15	11:15 – 11:30	11:30 – 11:45	11:45 – 12:00
Alumno 1	(OK)	OK		OK	(OK)			OK
Alumno 2	OK	OK	OK	(OK)	OK	(OK)	OK	(OK)
Alumno 3	OK	OK	(OK)	OK	(OK)	(OK)	OK	OK
Alumno 4	(OK)	(OK)	OK					OK
Alumno 5	(OK)	OK	OK	OK	OK		OK	(OK)
Alumno 6	(OK)	(OK)	OK					OK
Alumno 7		OK		OK			OK	OK
Alumno 8		OK		OK			OK	OK
Contador	2:4:1	5:2:0	4:1:2	4:1:2	2:2:3	1:2:4	4:0:3	5:2:0

Figura 18: Archivo excel con las nuevas disponibilidades del alumno 1.

Fuente: Elaboración Propia

Como se puede apreciar en la figura 19, el *ISP Solver* solamente realizó un swap entre el *Alumno 1* y *2*. Respecto a lo anterior, efectivamente se realizó la mínima cantidad de cambios para poder generar una replanificación que sea factible para el problema.



The screenshot shows a window titled "Resultados Planificación" with a table of scheduling results. The table has two columns: "Bloque" and "Nombre Alumno". Below the table is a text input field for "Nombre archivo a generar" with ".xls" as a suffix, and an "Exportar a Excel" button.

Bloque	Nombre Alumno
18/07 10:00 - 10:15	Alumno 1
18/07 10:15 - 10:30	Alumno 8
18/07 10:30 - 10:45	Alumno 4
18/07 10:45 - 11:00	Alumno 2
18/07 11:00 - 11:15	Alumno 5
18/07 11:15 - 11:30	Alumno 3
18/07 11:30 - 11:45	Alumno 7
18/07 11:45 - 12:00	Alumno 6

Figura 19: Resultados de la replanificación generadas por el ISP Solver.
Fuente: Elaboración Propia

4.2. VALOR DEL PROGRAMA

El *ISP Solver* es un programa con alta modularidad, es decir, todas las componentes del programa, como por ejemplo, funciones para lecturas de archivos, funciones para las creaciones de los modelos, funciones que ejecutan el solver para obtener una solución y funciones para la exportación a excel están totalmente separadas entre ellas. Lo anterior, permite que para posibles profesores que deseen agregar un nuevo criterio de optimización, o agregar nuevas restricciones al problema, simplemente deben agregar este criterio o esta restricción a la función que corresponda, sin la necesidad de tener que modificar el programa completo. Lo mismo ocurre si es que se desea modificar el estilo del formato del excel, o si se desea modificar el parámetro que genera múltiples planificaciones. Todo el diagrama de flujo para entender el funcionamiento del programa se encuentra en la sección de Anexos 5.2

Además, este programa se puede llevar a otros campos no necesariamente que busquen generar planificaciones de interrogaciones para un contexto académico. Por ejemplo, se podría tomar el mismo modelo matemático y los mismos input de Doodle para generar una planificación de entrevistas a distintas personas que deseen aplicar a un trabajo, donde el reclutador o la reclutadora podría asignarles valores a los distintos candidatos en caso de que desee que ciertas personas piense que deban ir juntas, generando finalmente una planificación óptima que satisfaga las preferencias de horarios que poseen los candidatos.

Finalmente, el programa aporta bastante valor a los distintos actores del problema de planificación de interrogaciones, ya que además de satisfacer las preferencias de profesores y alumnos, logra entregar múltiples planificaciones en sólo segundos, lo que le terminaría ahorrando bastante tiempo al profesor y le entregaría más holgura en casos de emergencia si se necesitara replanificar.

4.3. EXPERIMENTACIÓN TOTALMENTE ALEATORIA

En la realización de experimentos de este tipo, es importante saber cuánto es lo que tardará el programa en encontrar soluciones a instancias del problema, para saber si es útil la utilización de un solver que utiliza una técnica exacta en términos de tiempo, teniendo en consideración que muchas veces los profesores no poseen el suficiente tiempo para generar planificaciones.

Para el primer experimento, se realizará tres simulaciones donde se generarán en cada una de ellas instancias con solución comenzando desde 10 alumnos, incrementando la cantidad de alumnos en 1 hasta llegar a una instancia de 300 alumnos, es decir, la simulación i se definirá como $s_i = \{I_{10}, I_{11}, \dots, I_{300}\}$. La configuración de cada instancias se define de la siguiente manera:

- Cada alumno tiene entre 4 y 7 bloques de disponibilidad de manera aleatoria.
- Cada alumno tiene entre 1 y 3 bloques de preferencia de manera aleatoria.
- La valoración de los alumnos por el profesor en términos del tiempo que probablemente demore cada uno de ellos, se toma una muestra aleatoria del 35 % de los alumnos de cada instancia, asignando de manera equiprobable los valores 2 y 1 (alumno que probablemente exceda el tiempo y alumno que probablemente demore menos del tiempo, respectivamente) a cada alumno de esa muestra.

4.3.1. RESULTADOS DEL TIEMPO DE EJECUCIÓN

Los resultados en relación al tiempo de ejecución (Ver Figura 20), hablan de que a medida que crece la cantidad de alumnos y, por ende, de bloques, el tiempo de ejecución va aumentando

exponencialmente, llegando a tener un tiempo promedio alrededor de los 9,5 segundos para la instancia de 300 alumnos. Por otro lado, se puede ver que para las instancias más pequeñas y mucho más realistas desde 10 alumnos hasta los 50 alumnos, el programa no tarda más de 1 segundo en promedio en lograr obtener una respuesta a cada problema.

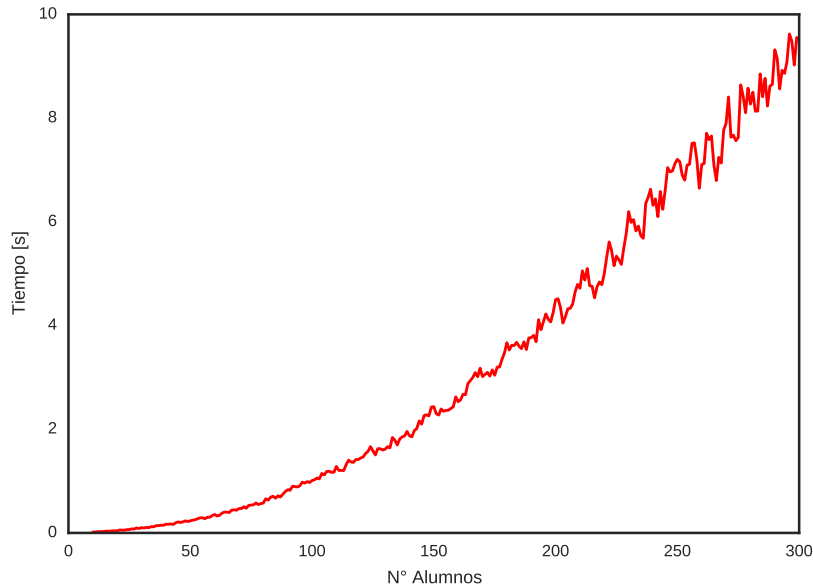


Figura 20: Promedio del tiempo de ejecución de las tres simulaciones.
Fuente: Elaboración Propia

4.3.2. RESULTADOS DE PREFERENCIAS Y ASIGNACIONES INTELIGENTES

Los resultados en relación al número de preferencias satisfechas y la cantidad de asignaciones inteligentes, nos muestran resultados bastante interesantes. En relación a la cantidad de preferencias satisfechas, se puede ver que el número incrementa linealmente a medida que incrementa la cantidad de alumnos en cada instancia (Ver Figuras 21 y 23). Sin embargo, se puede ver que es bastante difícil y poco común que las asignaciones realizadas puedan satisfacer las preferencias de todos los alumnos.

Respecto a la cantidad de asignaciones inteligentes, se puede ver un comportamiento mucho más errático en las tres simulaciones, con un crecimiento mucho más lento comparado con la cantidad de preferencias satisfechas (Ver Figuras 22 y 23). A partir de lo anterior, se deduce que es mucho más complejo para esta configuración de instancias realizar asignaciones inteligentes que satisfacer preferencias.

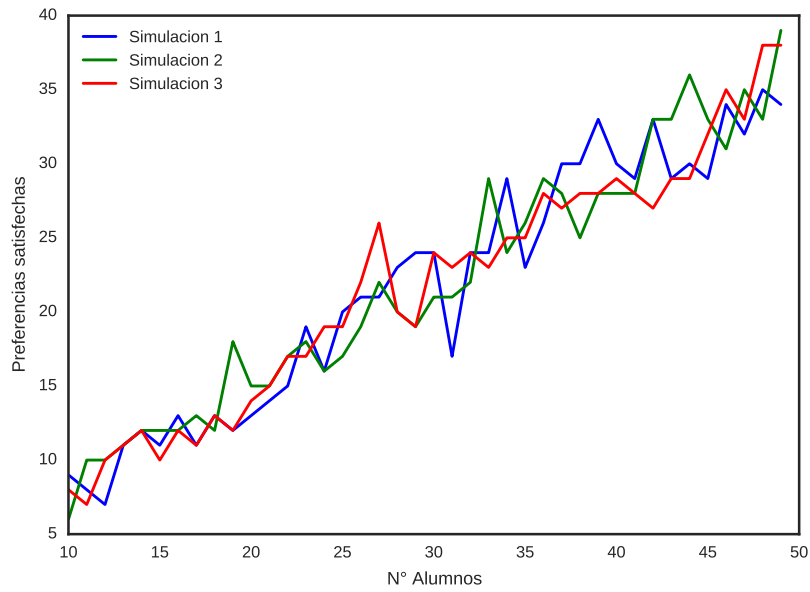


Figura 21: Gráfico de preferencias satisfechas en función de la cantidad de alumnos.
Fuente: Elaboración Propia

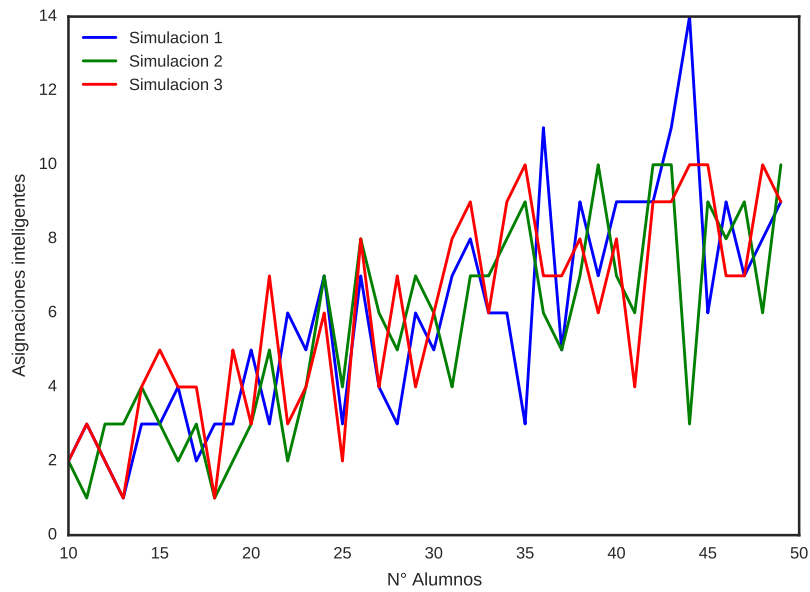


Figura 22: Gráfico de asignaciones inteligentes en función de la cantidad de alumnos.
Fuente: Elaboración Propia

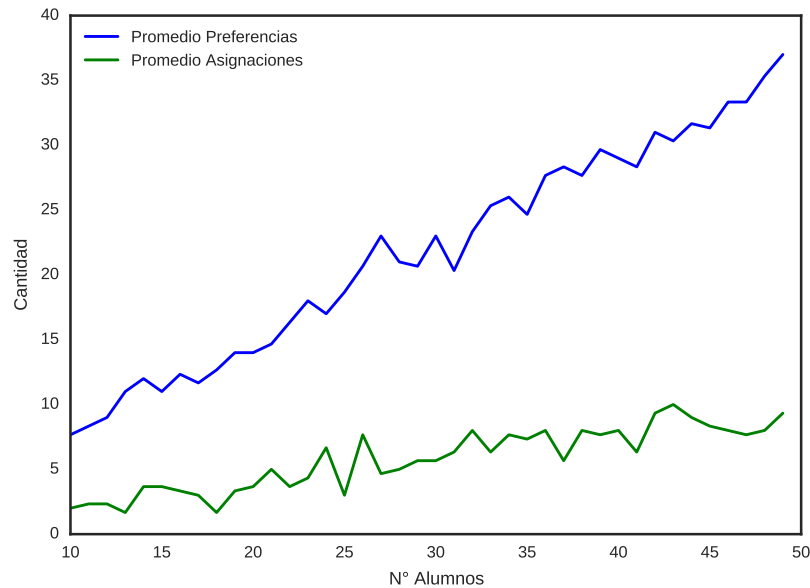


Figura 23: Promedio de preferencias satisfechas y asignaciones inteligentes en función de la cantidad de alumnos.

Fuente: Elaboración Propia

4.4. EXPERIMENTACIÓN CON DISPONIBILIDAD

Para esta experimentación, se realizó una encuesta a los alumnos de la asignatura de Inteligencia Artificial 2021-1 de la UTFSM, los cuales a través de una encuesta en Doodle debían responder con su disponibilidad y preferencias en 20 distintos horarios, con el fin de analizar cómo están distribuidas ambas características.

La encuesta la respondieron solamente 8 alumnos, logrando obtener los siguientes histogramas tanto para la disponibilidad de horarios como para las preferencias (Ver Figuras 24 y 25).

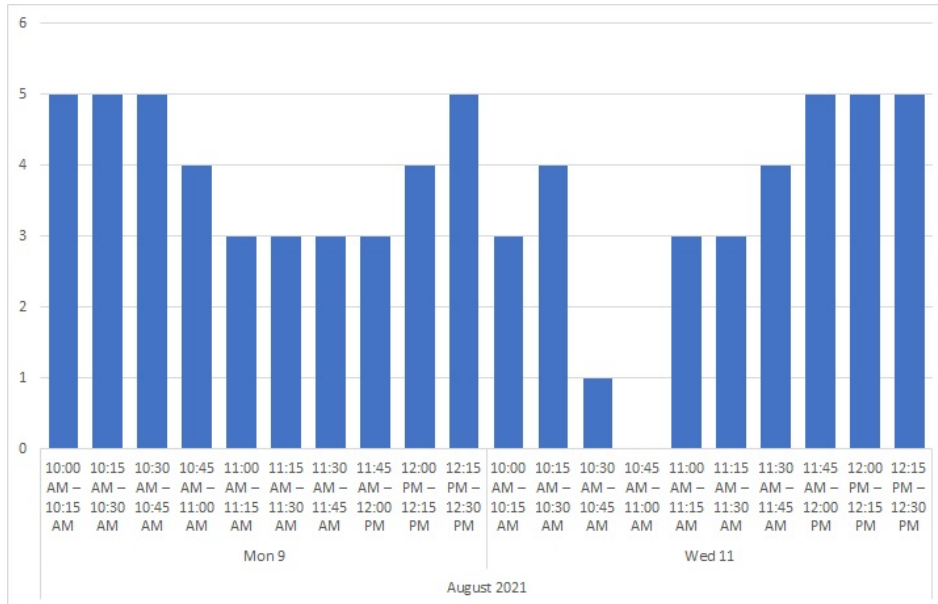


Figura 24: Histograma de disponibilidad de horarios.
Fuente: Elaboración Propia

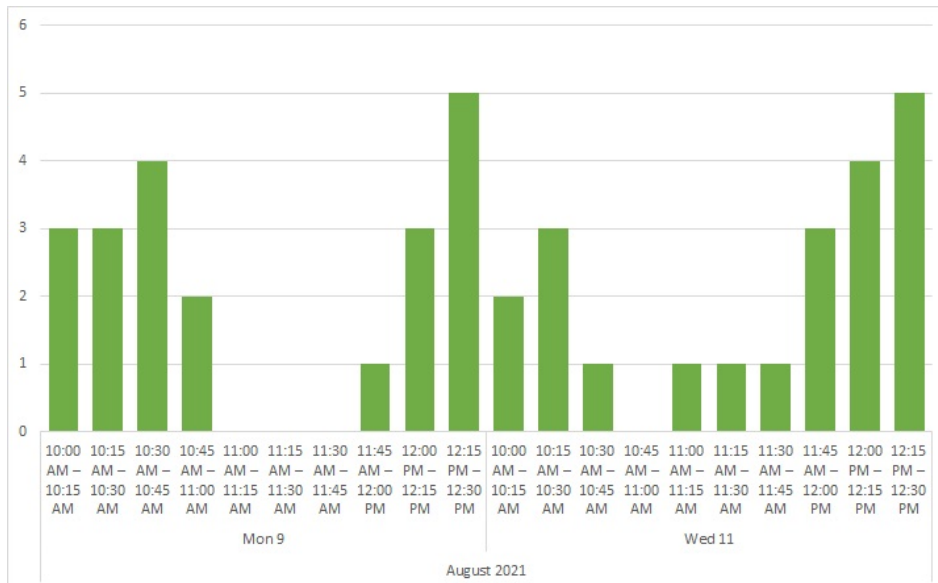


Figura 25: Histograma de preferencias de horarios
Fuente: Elaboración Propia

Se puede ver que para el caso de la disponibilidad, no existe sesgo hacia ningún horario específico. Además, se puede evidenciar una distribución alta de disponibilidad para los bloques tempranos de la mañana del día Lunes, como también los bloques cercanos a mediodía del Miércoles.

Por otro lado, respecto a las preferencias se ve un comportamiento similar en ambos días, teniendo más preferencia a los bloques cercanos a mediodía y preferencia bastante baja e incluso casi nula en los horarios de las 11 : 00 – 11 : 30.

A partir de los resultados obtenidos anteriormente, resulta importante también analizar las medidas de tendencia central de ambas características en los alumnos encuestados. A continuación se muestran las distintas medidas estadísticas de la cantidad de disponibilidad de los alumnos encuestados:

- Mínimo: 3
- Máximo: 16
- Promedio: 9,13
- Desviación Estándar: 5,57

Respecto a las medidas estadísticas de la cantidad preferencias de los alumnos encuestados, se obtuvo los siguientes resultados:

- Mínimo: 2
- Máximo: 10
- Promedio: 5,25
- Desviación Estándar: 2,66

Como se puede ver, tanto para las disponibilidades y las preferencias, existen alumnos que se encuentran en los extremos, es decir, tienen muy poca disponibilidad y, por ende, muy pocas preferencias y viceversa.

4.5. EXPERIMENTACIÓN CON DISPONIBILIDADES

A continuación se realizarán 3 tipos de experimentos, los cuales buscan analizar y obtener resultados a partir de la generación de distintas instancias con una tasa de disponibilidad alta, normal y baja de los alumnos, para ver de qué manera impacta la cantidad de preferencias satisfechas, la cantidad de asignaciones inteligentes y finalmente la cantidad de instancias que poseen soluciones factibles.

4.6. INSTANCIAS CON DISPONIBILIDADES NORMALES

Para este experimento, se generaron 1000 instancias con 30 alumnos y 30 bloques, con la disponibilidad de alumnos $D_i \sim N(9, 5)$, las preferencias de los alumnos $P_i \sim U_i(1, \max(d_i))$, siendo estas distribuciones escogidas en base a los resultados analizados de la encuesta. Para el caso del vector T de valoraciones a alumnos, se escoge en cada instancia 14 alumnos totalmente al azar, se realiza un *shuffle* entre ellos y se le asigna a la primera mitad el valor 2 y a la segunda mitad un 1 de valoración.

Los resultados obtenidos para esta experimentación son bastante buenos e interesantes. Respecto al histograma de preferencias satisfechas se puede ver que la frecuencia se distribuye similar a una curva normal normal, alcanzando aproximadamente como promedio 21 quitando los valores atípicos (Ver Figura 26). Por otro lado, respecto a la cantidad de asignaciones inteligentes se puede ver un comportamiento exponencial (Ver Figura 27), donde el número de las asignaciones inteligentes realizadas se concentran en el máximo. Esto nos indica que con esta configuración es posible realizar la mayoría de asignaciones inteligentes y se obtiene un muy buen número de preferencias satisfechas.

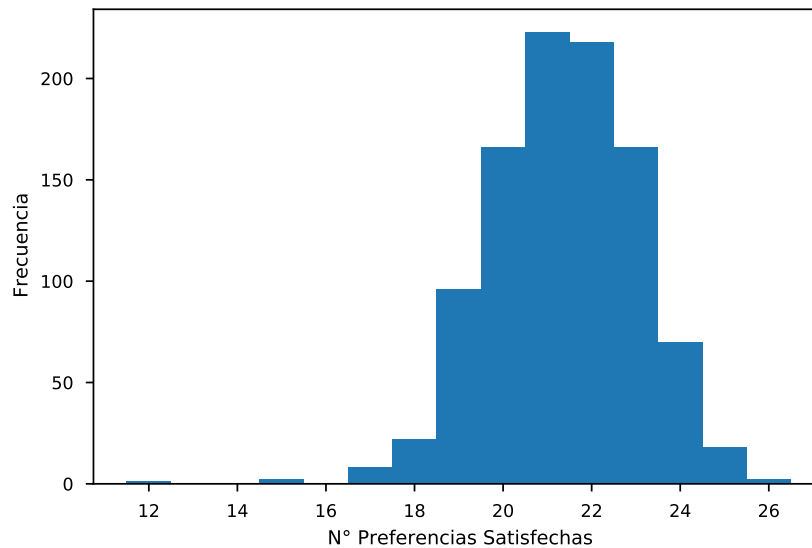


Figura 26: Histograma de preferencias satisfechas con disponibilidades normales.

Fuente: Elaboración Propia

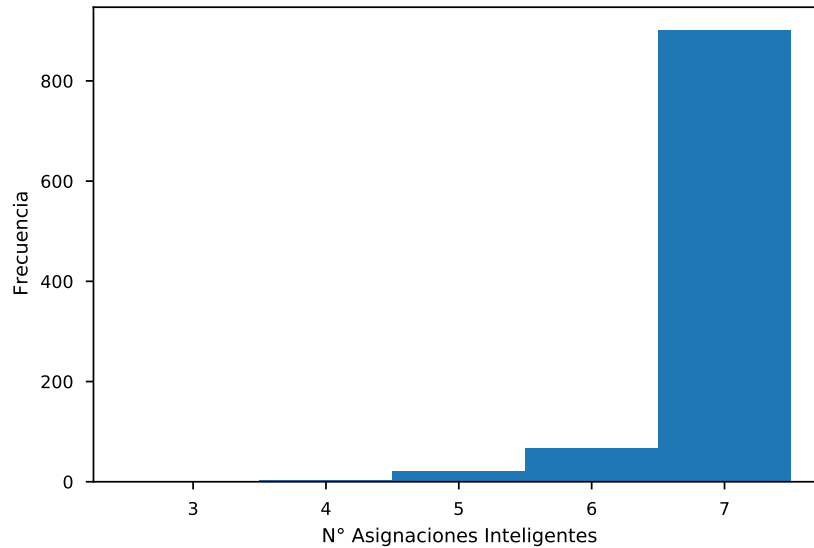


Figura 27: Histograma de asignaciones inteligentes con disponibilidades normales.
Fuente: Elaboración Propia

En términos de la factibilidad de las soluciones con esta configuración, se obtuvieron resultados casi excelentes siendo casi la totalidad de la soluciones factibles (Ver Tabla 4):

Total Instancias	Soluciones Factibles	Soluciones Infactibles
1000	992	8

Tabla 4: Cantidad de soluciones factibles e infactibles con disponibilidades normales.
Fuente: Elaboración Propia.

4.7. INSTANCIAS CON ALTA DISPONIBILIDAD

Respecto a los resultados obtenidos anteriormente, resulta interesante también analizar qué pasaría si existe bastante disponibilidad en los alumnos. Para este experimento, se generaron 1000 instancias con 30 alumnos y 30 bloques, con la disponibilidad de alumnos $D_i \sim N(12, 4)$ y las preferencias de los alumnos $P_i \sim U_i(1, \max(d_i))$. La generación del vector T de valoraciones a alumnos, se realizó de la misma manera que en el experimento anterior.

Se observa nuevamente las preferencias satisfechas se distribuyen semejante a una curva normal según su frecuencia, incrementando el número de éstas en relación a si fuese una configuración con una cantidad de disponibilidades normales. El promedio se encuentra

aproximadamente en 25 y se obtuvo soluciones en las cuales alcanzó a realizar 29 asignaciones que satisfacían las preferencias de esos alumnos (Ver Figura 28).

Respecto a las asignaciones inteligentes, se observa que casi la totalidad de las instancias generadas pudieron realizar la totalidad de asignaciones inteligentes posibles (Ver Figura 29).

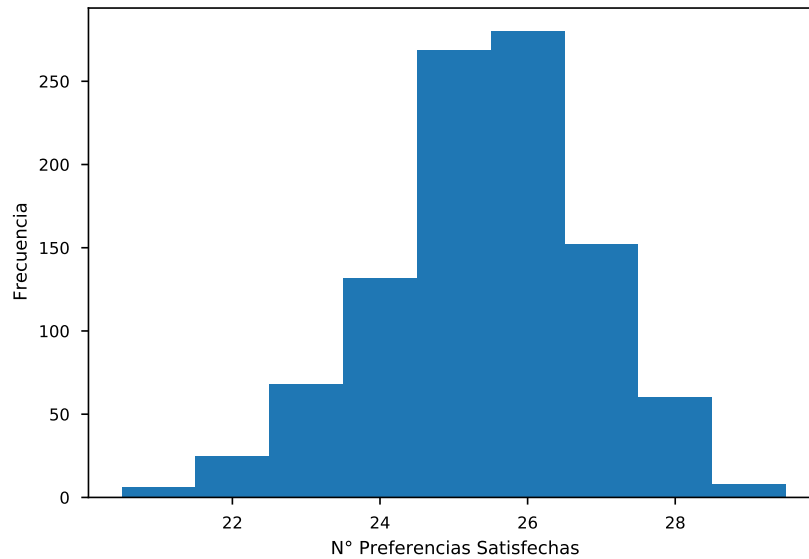


Figura 28: Histograma de preferencias satisfechas con alta disponibilidad.
Fuente: Elaboración Propia

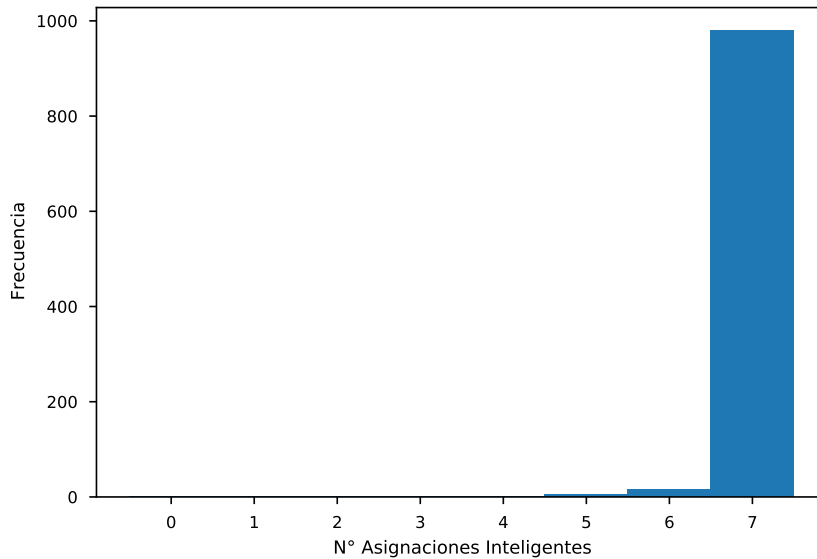


Figura 29: Histograma de asignaciones inteligentes con alta disponibilidad.
Fuente: Elaboración Propia

Finalmente, para este experimento se obtuvo un 100 % de factibilidad en todas las instancias generadas, por lo que nos indica que si la disponibilidad de los alumnos es alta, la probabilidad de que la solución sea factible incrementará (Ver Tabla 5):

Total Instancias	Soluciones Factibles	Soluciones Infactibles
1000	1000	0

Tabla 5: Cantidad de soluciones factibles e infactibles con disponibilidades bajas.
Fuente: Elaboración Propia.

4.8. INSTANCIAS CON BAJA DISPONIBILIDAD

Por el contrario a la experimentación anterior, se experimentará y se analizarán los resultados para instancias donde los alumnos tengan muy poca disponibilidad para ver de qué manera afectan las características estudiadas. Para este experimento, se generaron 1000 instancias con 30 alumnos y 30 bloques, con la disponibilidad de alumnos $D_i \sim N(4, 2)$ y las preferencias de los alumnos $P_i \sim U_i(1, \max(d_i))$. La generación del vector T de valoraciones a alumnos, se realizó de la misma manera que en el experimento anterior.

Las preferencias satisfechas para este experimento también se distribuyen de manera normal, sin embargo, la cantidad de preferencias satisfechas se ve mucho más reducida, tenien-

do la moda en el valor 16 y como valor mínimo 10, siendo valores muchos más bajos que los vistos en los dos experimentos anteriormente realizados (Ver Figura 30).

Respecto a las asignaciones inteligentes, se puede observar una distribución totalmente distinta a las obtenidas anteriormente (Ver Figura 37), donde los otros valores de asignaciones inteligentes salen a la vista en el gráfico, teniendo como moda el valor 5 y como mínimo el valor 0. También se puede apreciar que la frecuencia del máximo de asignaciones inteligentes que se podría realizar para esta simulación, tiene casi la misma frecuencia que la realización de tres asignaciones inteligentes.

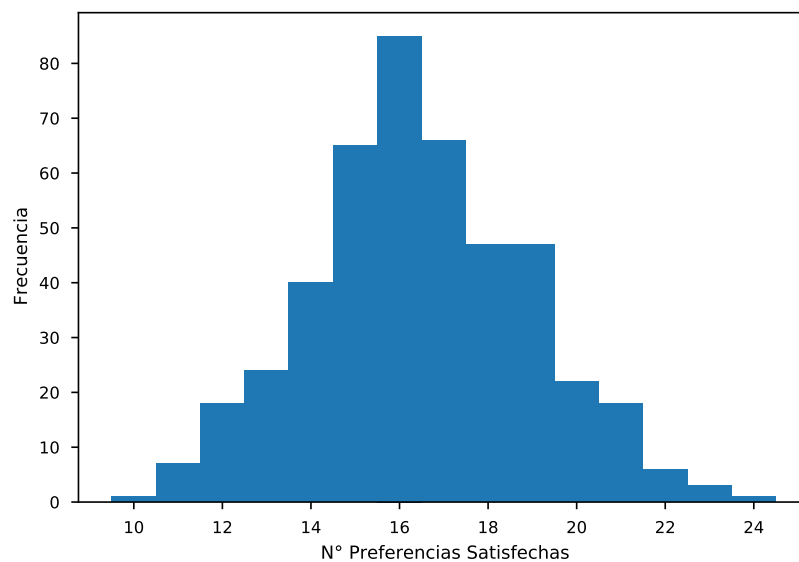


Figura 30: Histograma de preferencias satisfechas con baja disponibilidad.
Fuente: Elaboración Propia

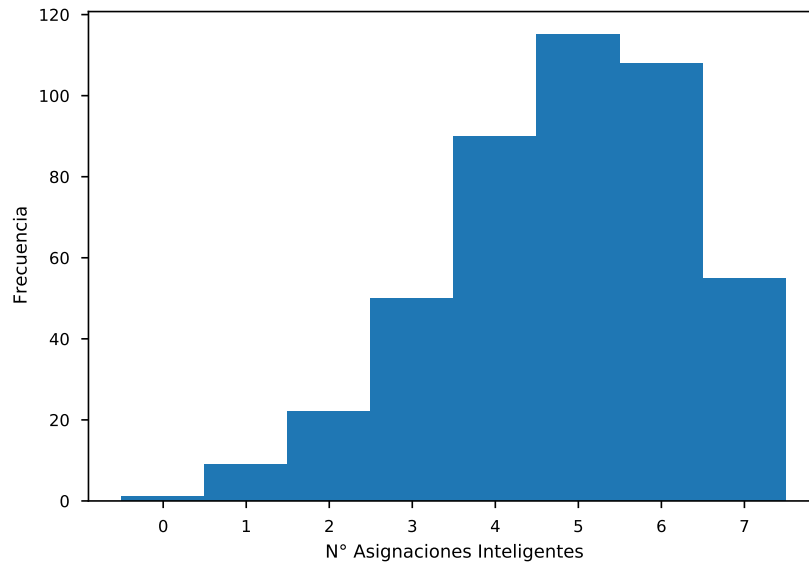


Figura 31: Histograma de asignaciones inteligentes con baja disponibilidad.
Fuente: Elaboración Propia

Respecto a la cantidad de soluciones factibles e infactibles para este experimento, se obtuvo que un 45 % de las soluciones fueron factibles, y el otro 55 % de las soluciones fueron infactibles (Ver Tabla 6):

Total Instancias	Soluciones Factibles	Soluciones Infactibles
1000	450	550

Tabla 6: Cantidad de soluciones factibles e infactibles con disponibilidades bajas.
Fuente: Elaboración Propia.

Estos resultados obtenidos indican que el hecho de trabajar con alumnos que posean muy poca disponibilidad afectan de manera negativa la cantidad de asignaciones inteligentes, las preferencias satisfechas y también la factibilidad de la instancia con la que se está trabajando.

4.9. ANÁLISIS DE SENSIBILIDAD RESPECTO A BLOQUES ADICIONALES

4.9.1. CON UN BLOQUE EXTRA

A partir de los resultados obtenidos anteriormente, se experimentará con la misma configuración de generación de instancias, cambiando solamente la cantidad de bloques para ver

de qué manera afectan los resultados al agregar holgura de bloques para cada instancia.

Para el primer experimento, se agregó un solo bloque a cada instancia, es decir, en lugar de trabajar con 30 alumnos y 30 bloques, ahora se generarán instancias de 30 alumnos y 31 bloques.

Se puede observar que en el histograma de preferencias satisfechas (Ver Figura 32), la distribución de las preferencias tiene un crecimiento mucho más suave a diferencia del resultado de la experimentación anterior y la frecuencia de aparición de cada número de preferencias satisfechas incrementa. Por otro lado, para el número de asignaciones inteligentes se puede ver un incremento en la frecuencia para los últimos valores del gráfico, por lo que se puede deducir que agregar un solo bloque más al total de la instancia, afecta de manera bastante positiva la cantidad de asignaciones inteligentes que puedan realizarse (Ver Figura 33).

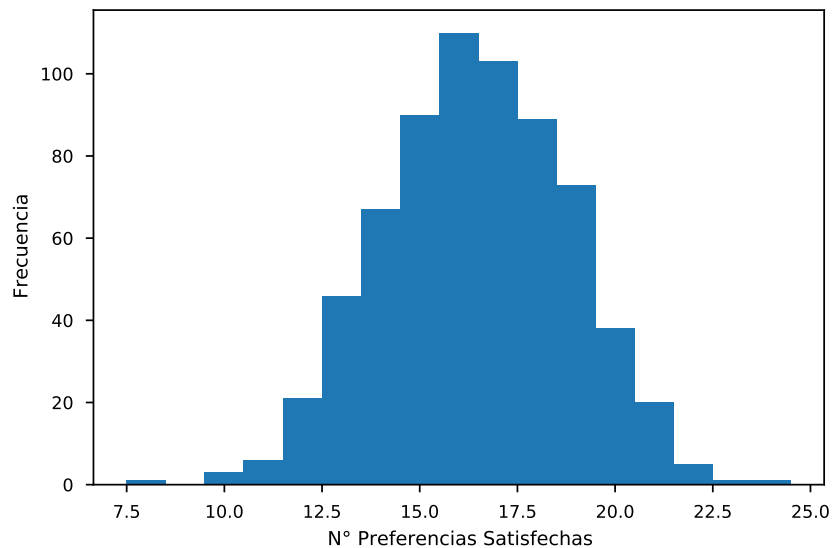


Figura 32: Histograma de preferencias satisfechas con baja disponibilidad y 31 bloques.
Fuente: Elaboración Propia

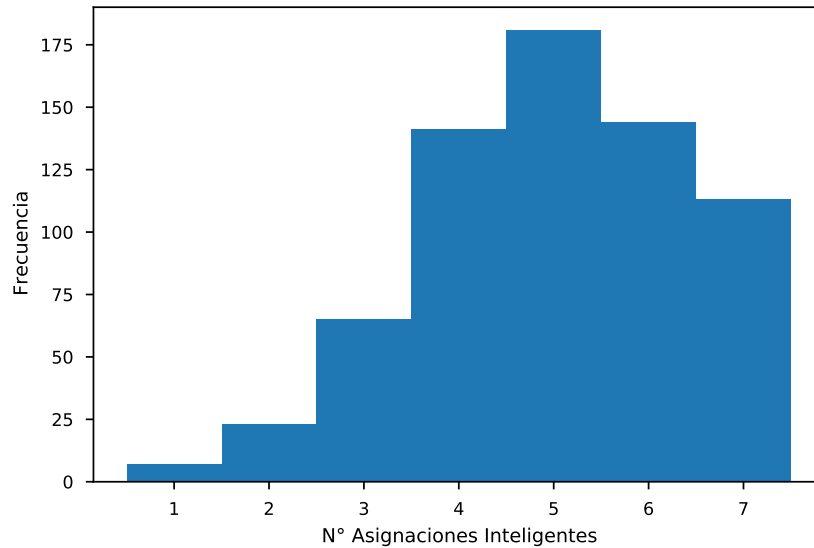


Figura 33: Histograma de asignaciones inteligentes con baja disponibilidad y 31 bloques.
Fuente: Elaboración Propia

Respecto a la factibilidad de las soluciones obtenidas, se puede ver una mejora respecto al experimento anterior, obteniendo esta vez un 67,4 % de soluciones factibles y un 32,6 % de soluciones infactibles, lo que resulta en una mejora de un 22 % en términos de factibilidad (Ver Tabla 11):

Total Instancias	Soluciones Factibles	Soluciones Infactibles
1000	674	326

Tabla 7: Cantidad de soluciones factibles e infactibles con disponibilidades bajas y 31 bloques.

Fuente: Elaboración Propia.

4.9.2. CON DOS BLOQUES EXTRAS

Para este experimento, se decidió agregar dos bloques extras, es decir, en vez de tener una instancia de 30 alumnos y 30 bloques, esta vez se tendrán 30 alumnos y 32 bloques, dando mucha más holgura al problema, esperando que mejoren de manera mucho más significativa los resultados con baja disponibilidad.

En relación a las preferencias satisfechas, se puede ver que también posee un crecimiento parecido a una distribución exponencial (Ver Figura 34, con una nueva moda esta vez en 17 y una curva también suave en relación con el primer histograma obtenido con disponibilidad

baja, teniendo la cola derecha con una caída un poco más suave.

Por otro lado, para las asignaciones inteligentes se observa que cambian de manera bastante positiva para este experimento, donde se ve claramente un incremento en la frecuencia de aparición para los números más altos de asignaciones inteligentes (Ver Figura 35).

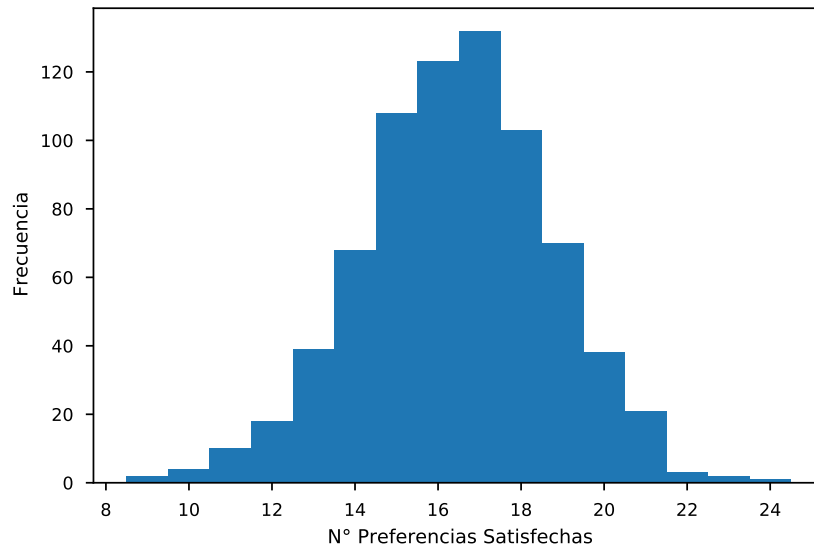


Figura 34: Histograma de preferencias satisfechas con baja disponibilidad y 32 bloques.
Fuente: Elaboración Propia

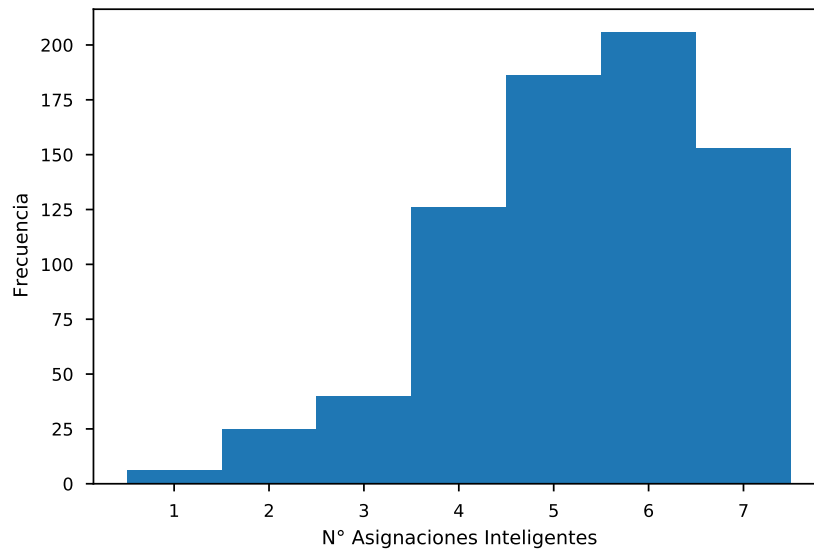


Figura 35: Histograma de asignaciones inteligentes con baja disponibilidad y 32 bloques.
Fuente: Elaboración Propia

Respecto a la factibilidad de las soluciones, se obtuvo que un 75,2% de las soluciones eran factibles y un 25,8% eran infactibles, por lo que es una mejora del 30,2% en relación a los resultados obtenidos inicialmente para los 30 bloques (Ver Tabla 8):

Total Instancias	Soluciones Factibles	Soluciones Infactibles
1000	752	258

Tabla 8: Cantidad de soluciones factibles e infactibles con disponibilidades bajas y 32 bloques.

Fuente: Elaboración Propia.

4.10. RESULTADOS GENERALES

A continuación se mostrarán los resultados generales en una tabla respecto a la experimentación de disponibilidad realizada en los puntos anteriores:

	Moda	Frecuencia Modal
Alta Disponibilidad	26	275
Disponibilidad Normal	21	225
Baja Disponibilidad	16	85
Baja Disponibilidad +1 Bloque	16	110
Baja Disponibilidad +2 Bloques	17	130

Tabla 9: Resultados Obtenidos respecto a la distribución de preferencias satisfechas.
Fuente: Elaboración Propia.

	Moda	Frecuencia Modal
Alta Disponibilidad	7	950
Disponibilidad Normal	7	850
Baja Disponibilidad	5	110
Baja Disponibilidad +1 Bloque	5	175
Baja Disponibilidad +2 Bloques	6	200

Tabla 10: Resultados Obtenidos respecto a la distribución de asignaciones inteligentes.
Fuente: Elaboración Propia.

	Factibilidad
Alta Disponibilidad	100 %
Disponibilidad Normal	99,2 %
Baja Disponibilidad	45 %
Baja Disponibilidad +1 Bloque	67,4 %
Baja Disponibilidad +2 Bloques	75,2 %

Tabla 11: Resultados Obtenidos respecto a la factibilidad de las soluciones.
Fuente: Elaboración Propia.

Como se puede apreciar, tanto para los casos de disponibilidad normal y alta, se logra ver que se obtienen resultados bastante buenos para la cantidad de preferencias y asignaciones inteligentes, por lo que un aumento de la disponibilidad de los alumnos afecta todos los factores que se evaluaron en estos experimentos. Además, se logra casi un 100 % de factibilidad para el caso de disponibilidad normal, tomando en cuenta los datos utilizados directamente de la encuesta.

Por otro lado, se ve que una configuración con baja disponibilidad afecta de manera bastante negativa el problema, donde se nota que más de la mitad de las instancias generadas a través de esta configuración no poseen una solución factible al problema. Por lo tanto, la agregación de bloques extra permite agregar más holgura al problema, mejorando la factibilidad a un 75 % en el caso de agregar dos bloques extra, y también mejorando bastante la cantidad de asignaciones inteligentes y preferencias satisfechas del problema.

4.11. CONCLUSIONES

En esta sección, se realizaron bastantes experimentos para ver el comportamiento tanto de la factibilidad de las soluciones encontradas, como también el número de preferencias satisfechas y de asignaciones inteligentes que se pueden realizar en distintos tipos de instancias.

Primero, se comenzó con una presentación del programa realizado en este trabajo llamado *ISP Solver*, sobre el cual se explicó cada sección que posee para la generación de planificaciones y replanificaciones, mostrando además ejemplos de posibles entradas que se le pueden ingresar al programa y un breve análisis de los resultados obtenidos en ambos casos. Se concluye que este programa logra resolver de manera efectiva y bastante rápida instancias de un ISP, agregándole bastante valor a la asignatura de Inteligencia Artificial en la UTFSM, tanto para los profesores, como para los alumnos.

Por otro lado, se realizó una experimentación totalmente aleatoria con tres simulaciones, cada una de ellas con 290 instancias de un ISP. Se pudo observar que el tiempo de ejecución del programa aumenta de manera exponencial a medida que aumenta la cantidad de bloques y alumnos, como también se pudo notar un crecimiento lineal tanto en las preferencias satisfechas como en la cantidad de asignaciones inteligentes.

Finalmente, se realizó un estudio con una pequeña muestra del curso de la asignatura de Inteligencia Artificial, para tener una noción sobre cómo es la distribución de disponibilidades y preferencias que poseen los alumnos en un horario acotado. A partir de los resultados obtenidos, se realizaron tres tipos de experimentos, primero se realizaron experimentos con disponibilidad normal, en los cuales se obtuvieron buenos resultados tanto en factibilidad, como en las otras características. Luego se realizó un experimento con alta disponibilidad en los alumnos, teniendo resultados bastante positivos y esperables, mientras que para el tercer experimento se realizó con disponibilidades bastante bajas, obteniendo que la mayoría de las soluciones encontradas eran soluciones infactibles al problema y que la cantidad de preferencias satisfechas y asignaciones inteligentes se veían bastante reducidas para esta configuración. En relación a lo anterior, se experimentó con agregar holgura para la configuración con baja disponibilidad, obteniendo mejoras significativas en la factibilidad, alcanzando incluso aproximadamente un 30 % de mejora y también un incremento en la frecuencia de preferencias satisfechas y asignaciones inteligentes.

CAPÍTULO 5

CONCLUSIONES

El problema de la planificación de interrogaciones es un asunto de suma importancia en la vida real, debido a su alta complejidad para poder resolver una instancia de forma manual y la cantidad de posibles combinaciones que se pueden ir generando a medida que el espacio de búsqueda aumenta. En esta memoria se propuso un modelo matemático del problema y se propuso aplicar un enfoque basado en técnicas exactas para la resolución de este, teniendo en consideración que podría llegar a tomar bastante tiempo encontrar una solución a este problema bajo este enfoque.

Por otro lado, el estudio de este problema se realizó mediante distintos experimentos, variando principalmente la distribución de las disponibilidades y las preferencias, inicialmente realizando una distribución totalmente aleatoria para ambas características, donde se pudo observar que las soluciones obtenidas a través de tres distintas simulaciones lograron buenos resultados, satisfaciendo poco más de la mitad de las preferencias posibles en cada instancia y también evidenciando un crecimiento más lento al momento de realizar asignaciones inteligentes. Finalmente, se decidió realizar experimentos más cercanos a la realidad, tomando datos directamente de una encuesta realizada a una pequeña muestra de la asignatura de Inteligencia Artificial, donde se generaron instancias aleatorias con una distribución normal de la cantidad de disponibilidades de los alumnos. A partir de la experimentación anterior, se obtuvo que para una cantidad de disponibilidades normal (es decir, la misma configuración que se obtuvo directamente de la encuesta), se obtuvieron resultados bastante buenos y casi el 100 % de las soluciones obtenidas fueron factibles. Al incrementar la cantidad de disponibilidad de los alumnos afecta de manera bastante positiva la factibilidad de las soluciones y también el número de preferencias satisfechas y asignaciones inteligentes. Por otro lado, disminuir la cantidad de disponibilidad de los alumnos impacta de manera bastante negativa y brusca la factibilidad del problema, llegando incluso a que más del 50 % de las soluciones obtenidas en las instancias eran infactibles.

Finalmente, se pudo ver que el software creado en este trabajo resuelve de manera rápida y eficiente instancias de un ISP, obteniendo múltiples planificaciones a evaluar por el profesor en solamente segundos, lo cual quitaría una carga de trabajo bastante importante y al mismo tiempo le agrega valor a los alumnos, debido que logra tomar en consideración las preferencias cada uno de ellos posee. Además, la manera de cómo está implementado el programa, permitirá a futuros profesores o personas interesadas en este problema lograr agregar nuevas restricciones al modelo matemático, modificar las funciones de lectura o escritura del programa, entre otras posibles configuraciones sin la necesidad de modificar el programa completo debido a su alta modularidad.

5.1. RECOMENDACIONES

Las recomendaciones para abordar este problema son las siguientes:

- Realizar un breve estudio sobre la disponibilidad de los alumnos, al menos tener cuáles días son los que poseen una mayor cantidad de disponibilidad para fijar los horarios en esos días.
- Agregar un poco de holgura a la cantidad de bloques independiente de la disponibilidad de los alumnos, ya que aumenta bastante las probabilidades de que el problema pueda tener una solución.
- Si la disponibilidad de los alumnos es muy baja, se recomienda crear dos encuestas que apunte a distintos grupos de alumnos a criterio del profesor.

5.2. TRABAJO FUTURO

Se propone la creación de una plataforma web la cual incluya todas las funcionalidades que otorga Doodle para realizar la encuesta a los estudiantes, y también las funcionalidades del programa implementado en esta memoria. Esto con el fin de no generar una dependencia directa con la plataforma Doodle, ya que en algún futuro podría realizar algún tipo de cambio por ejemplo en el formato del excel exportado, lo que implicaría volver a configurar el proceso de lectura de los archivos.

Respecto al modelo matemático presentado para generar planificaciones, se propone estudiar la aplicación de pesos a ambas componentes (tanto a la de satisfacción de restricciones y asignaciones inteligentes), para ver el efecto que produce aumentar el peso hacia una componente u otra y lograr llegar a un análisis concreto sobre qué valores deberían tomar ambos pesos para llegar a un equilibrio, o también para priorizar una sola componente de manera justa.

Para finalizar, sería interesante también realizar ajustes al modelo matemático de la replanificación, pudiendo agregar a la función objetivo tanto la maximización de preferencias satisfechas, como también la priorización de asignaciones inteligentes generadas en alguna planificación anterior.

ANEXOS

En las siguientes dos páginas, se mostrarán los diagramas de flujo para los procesos de planificación y replanificación.

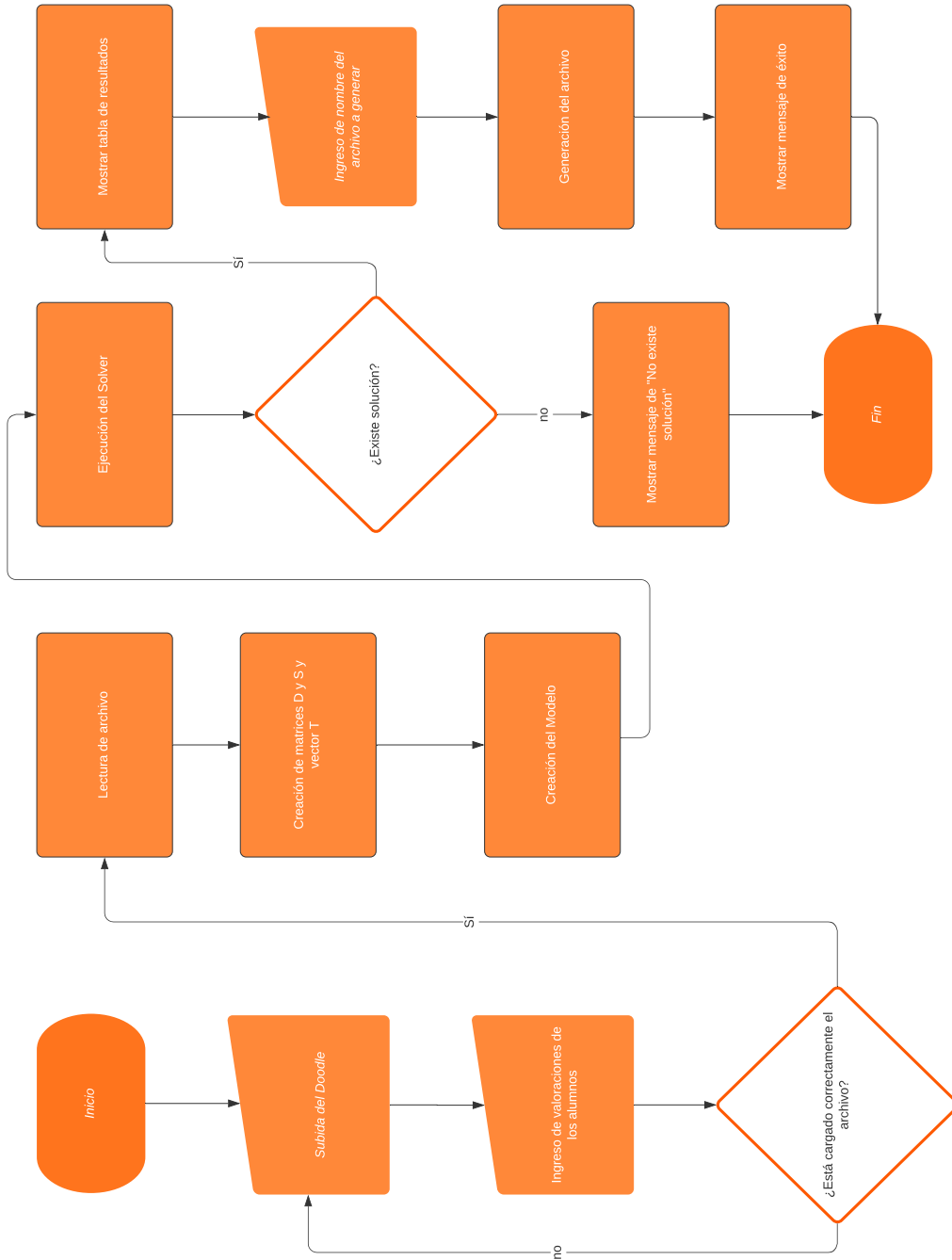


Figura 36: Diagrama de flujo del programa para realizar una planificación.
Fuente: Elaboración Propia

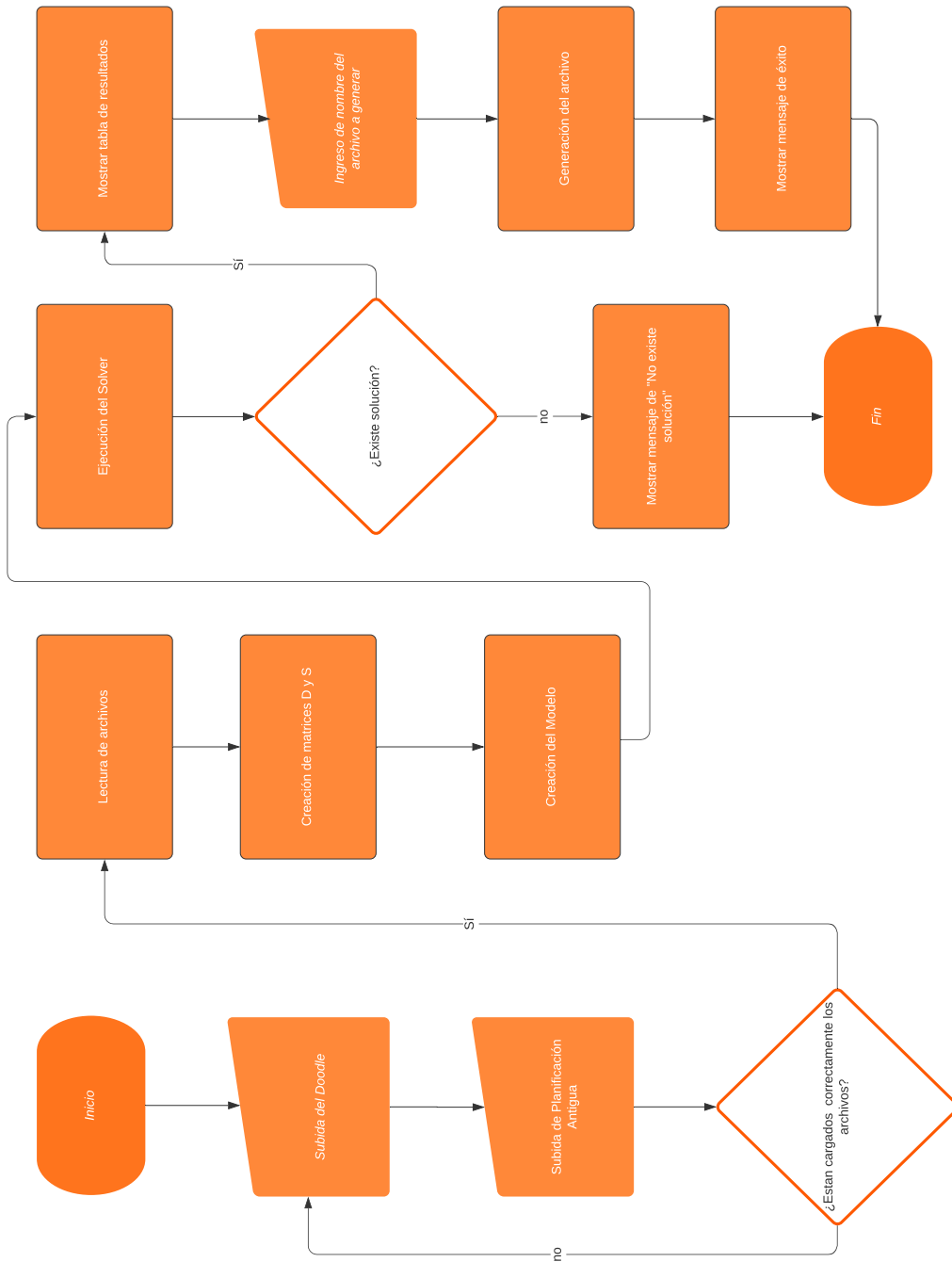


Figura 37: Diagrama de flujo del programa para realizar una replanificación.
Fuente: Elaboración Propia

REFERENCIAS BIBLIOGRÁFICAS

- [Abou Kasm *et al.*, 2019] Abou Kasm, O., Mohandes, B., Diabat, A., y El Khatib, S. (2019). Exam timetabling with allowable conflicts within a time window. *Computers & Industrial Engineering*, 127:263–273.
- [Ayob *et al.*, 2007] Ayob, M., Ab Malik, A. M., Abdullah, S., Hamdan, A. R., Kendall, G., y Qu, R. (2007). Solving a practical examination timetabling problem: a case study. En *International Conference on Computational Science and Its Applications*, pp. 611–624. Springer.
- [Barták, 1999] Barták, R. (1999). Constraint programming: What is behind. *Proceedings of CPDC99*, pp. 7–15.
- [Bidar *et al.*, 2018] Bidar, M., Mouhoub, M., Sadaoui, S., y Bidar, M. (2018). Solving constraint satisfaction problems using firefly algorithms. En *Canadian Conference on Artificial Intelligence*, pp. 246–252. Springer.
- [Dueck, 1993] Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, 104(1):86–92.
- [Ghédira y Dubuisson, 2013] Ghédira, K. y Dubuisson, B. (2013). Constraint satisfaction and optimization problems. *Constraint Satisfaction Problems*, pp. 165–180.
- [McCollum *et al.*, 2012] McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., y Qu, R. (2012). A new model for automated examination timetabling. *Annals of Operations Research*, 194(1):291–315.
- [Morrison *et al.*, 2016] Morrison, D. R., Jacobson, S. H., Sauppe, J. J., y Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102.
- [Müller, 2005] Müller, T. (2005). *Constraint-based Timetabling Ph. D.* Tesis doctoral, Charles University in Prague.
- [Müller, 2016] Müller, T. (2016). Real-life examination timetabling. *Journal of Scheduling*, 19(3):257–270.
- [Müller *et al.*, 2004] Müller, Tomáš and Barták, Roman and Rudová, Hana and others (2004). Conflict-based statistics. En *J. Gottlieb, D. Landa Silva, N. Musliu, and E. Soubeiga, editors, EU/ME Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics. University of Nottingham.*
- [Naud *et al.*, 2020] Naud, O., Taylor, J., Colizzi, L., Giroudeau, R., Guillaume, S., Bourreau, E., Crestey, T., y Tisseyre, B. (2020). Support to decision-making. En *Agricultural Internet of Things and Decision Support for Precision Smart Farming*, pp. 183–224. Elsevier.

[Romero *et al.*, 2017] Romero, D. M., Reinecke, K., y Robert Jr, L. P. (2017). The influence of early respondents: information cascade effects in online event scheduling. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 101–110.