

Inteligencia Artificial

Estado del Arte: Problema Examination Timetabling Problem

Isidora Ubilla Zavala

December 7, 2021

Evaluación

Resumen (5%):	_____
Introducción (5%):	_____
Definición del Problema (10%):	_____
Estado del Arte (35%):	_____
Modelo Matemático (20%):	_____
Conclusiones (20%):	_____
Bibliografía (5%):	_____
Nota Final (100%):	_____

Abstract

El en el presente paper se realiza un estudio al problema del mundo real que enfrentan muchas instituciones educacionales, conocido como Examination Timetabling Problem, el que surge al momento de asignar horarios a los diversos exámenes que los alumnos deben rendir, teniendo en consideración una serie de restricciones, algunas de las más comunes están relacionadas con cantidad y capacidad de salas disponibles o evitar topes de horarios entre dos exámenes, y es debido a la complejidad de las restricciones que el problema es todo un desafío para los investigadores del área. Este estudio presenta el estado del arte de ETP, centrándose en algunos de los estudios más interesantes y una revisión a un modelo matemático con enfoque multicriterio más relevante planteado hasta el momento, además se explora las clasificaciones de métodos heurísticos, finalizando con una conclusión y reflexión sobre ETP.

1 Introducción

Los Examination Timetabling Problems (ETP) son un desafío constante al que se deben enfrentan las organizaciones educacionales al momento de organizar los exámenes de sus alumnos dentro de un periodo de tiempo. El objetivo principal de los ETP es asignar los exámenes en ciertos bloques de horarios, donde existe una serie de restricciones asociadas [7]. Los ETP pertenecen a la gran familia de problemas Timetabling (TP), existiendo otras variantes en el mundo del transporte, salud, deportes, minería, entre muchos otros, los que han sido estudiado a lo largo de los años por los investigadores de la comunidad de la Inteligencia Artificial debido a su alto nivel de complejidad. En el contexto universitario, el proceso de organizar dichos exámenes

debe considerar que sus alumnos tienen horarios distintos, pues no todos tienen las mismas asignaturas, se debe pensar en la cantidad de salas disponibles para realizar las evaluaciones, los alumnos por su parte siempre buscan tener horarios de exámenes convenientes para lograr estudiar de manera adecuada para todos ellos, además de restricciones más importantes, como que un alumno no puede rendir dos pruebas al mismo tiempo. Sin duda toda la organización requiere una gran cantidad de tiempo y trabajo para las educaciones que tienen que tratar con ETP. En los últimos años este interés por encontrar una metodología que solucione en gran parte el problema se ha incrementado y esto se ve reflejado en los diversos papers y estudios que han salido a la luz, en general se enfocan en la aplicación de algoritmos basados en metaheurísticas, métodos de propósito general diseñados para resolver problemas de optimización combinatoria [20], que para resolver ETP en el contexto universitario funcionan de buena forma entregando soluciones de alto nivel [15]. A raíz de lo anterior surge el propósito de realizar el presente estudio; se busca recopilar información histórica de los ETP a lo largo de los papers más relevantes en la materia, para luego plantear un modelo matemático y por medio de la implementación de algoritmos Greedy, el que busca una solución inicial, y Simulated Annealing, que busca el resto de las soluciones, se logra resolver algunas instancias del problema detalladas más adelante.

A continuación, se enlista y detalla la estructura del presente estudio:

- **Definición del problema**, donde se explica en qué consiste el problema ETP de manera más detallada, se presentan sus variables, restricciones más comunes, el objetivo a resolver y sus variantes más conocidas.
- **Estado del Arte**, sección en la que se describen los estudios históricos más relevantes sobre el ETP hasta el presente. Se tratan las diversas metodologías planteadas para resolver el problema, las heurísticas utilizadas, además de los algoritmos más importantes diseñados para esto.
- **Modelo Matemático**, donde se presentan modelos importantes para resolver el problema, detallando en variables, restricciones, función objetivo y fórmulas involucradas.
- **Representación**, sección donde se detalla la representación de las soluciones con ejemplo para mayor comprensión.
- **Descripción del algoritmo**, en esta sección se muestra el pseudocódigo de la estructura general utilizada para la implementación de la solución.
- **Conclusiones**, se presentan las ideas más relevantes del estudio y se realiza una reflexión en torno a ciertas preguntas.

2 Definición del Problema

La definición general de Examination Timetabling Problem es la asignación de ciertos recursos, en este caso exámenes que juegan el rol de variable del problema, para ciertas tareas, en ciertos periodos de tiempo, cumpliendo una cantidad finita de restricciones que varían dependiendo de las necesidades de la institución educacional en cuestión. El objetivo principal que persigue es encontrar la mejor asignación posible entre exámenes, tiempos y actividades, de esta forma se puede asegurar un proceso de evaluación eficiente. Las organizaciones educacionales deben tratar constantemente este problema del mundo real, que debido a la dificultad de las diversas restricciones, entra en la clasificación de optimización combinatoria y en términos de complejidad computacional se tiene un NP-completo, donde el espacio de búsqueda es dependiente de la cantidad de exámenes para asignar [21].

En la literatura se identifican dos grandes tipos de restricciones, primero se tienen las restricciones duras, en inglés *Hard constraints*, son aquellas que se deben priorizar satisfacer bajo cualquier circunstancia, si el ETP satisface todas las restricciones duras, entonces es una *Solución Factible*; por otra parte, las restricciones blandas o *Soft constraints*, son aquellas que son deseadas pero no son mandatorias, se buscan satisfacer siempre que sea posible [2]. A continuación se enlistan algunas de las restricciones duras y blandas más comunes de los ETP [19].

- **Restricciones Duras:**

1. Los alumnos no pueden rendir dos exámenes al mismo tiempo.
2. La cantidad de alumnos que rinden examen no puede superar la capacidad de la sala.

- **Restricciones Blandas**

1. Evitar que dos exámenes de distintos cursos con alumnos en común sean en periodos de tiempo cercanos.
2. Programar con anticipación los exámenes grandes para tener tiempo de corrección.
3. Ciertos exámenes no pueden ser tomados en ciertos horarios o días.
4. Cumplir precedencia de los exámenes.
5. Cantidad limitada de estudiantes en ciertos horarios o ubicaciones.
6. Disponibilidad de Evaluadores.

Si bien, actualmente es imposible obtener una solución factible sin violar las restricciones blandas, es necesario apuntar a minimizar el número de incumplimientos de estas, pues la calidad del ETP se puede medir en base a satisfacción de restricciones blandas [21].

Cada organización educacional posee su propia variante de ETP, porque cada institución tiene sus restricciones y necesidades particulares, en consecuencia, la complejidad cambia dependiendo del tamaño de la infraestructura, cantidad de alumnos, cursos, salas, entre otros. Alguno de los problemas relacionados a ETP como el High School Timetabling Problem, curriculum based course timetabling, Course Timetabling, Sports Timetabling son de gran interés para la comunidad debido a la International Timetabling Competition (ITP) [1], competencia auspiciada por la organización The International Series of Conferences on the Practice and Theory of Automated Timetabling (PATAT), que buscan incentivar y desafiar a investigar en entorno al tema.

Gracias al creciente interés sobre ETP, han surgido diversos estudios con diferentes técnicas, metodologías y enfoques para lograr encontrar soluciones factibles. Algunas de las metodologías exitosas surgen de la mezcla de técnicas bajo enfoques no tradicionales. En la Tabla 1 se muestran algunos enfoques utilizados en investigaciones destacables que se pueden encontrar actualmente en la literatura, con respectivas referencias:

Enfoques	Referencias
Enfoques exactos	Boufflet and Nègre (1996); Bosch and Trick (2005); Sierksma (2001).
Basado en restricciones	Thompson and Dowsland (1996a); Duong and Lam (2004).
Técnicas heurísticas constructivas	Asmuni et al., (2009); Burke and Newall (1999); Carter et al., (1996); Gogos et al (2008); Laporte and Desroches (1984).
Técnicas heurísticas meta-heurísticas y de mejora	Petrovic and Burke (2004) and Qu et al. (2009b).
Técnicas hiperheurísticas y de razonamiento basado en casos	Yang and Petrovic (2004); Kendall and Mohd Hussin (2005a);
Técnicas multicriterio y multi-objetivos	Landa-Silva et al. (2004); Paquete & Fonseca (2001); Cheong et al. (2007).

Table 1: Diferentes enfoques utilizados para el estudio de ETP

Para el presente estudio se trabajara sobre cierta variante de ETP, esta se caracteriza por tener la restricción dura de no permitir topes de horarios entre dos exámenes, además se considera que la capacidad de las salas y la cantidad de timeslots son infinitos. En esta variante existen penalizaciones por estudiantes, la cual es directamente proporcional a la carga académica, está depende de la separación entre exámenes que debe rendir un mismo estudiante. La Tabla 2 refleja las penalizaciones a utilizar. Para la variante se presentan dos objetivos:

- **Objetivo Principal:** Minimizar los timeslots para rendir los exámenes.
- **Objetivo Secundario:** Minimizar penalización promedio por estudiante.

Sigla	Penalización
W_0	16
W_1	8
W_2	4
W_3	2
W_4	1
W_5	0

Table 2: Tabla de penalizaciones.

3 Estado del Arte

El Examination Timetabling Problem es un problema que lleva años siendo estudiado por la comunidad de investigadores y producto de esto han surgido una serie de diversos enfoques y metodologías para abordar el problema. Las primeras publicaciones del problema son del año 1964, el autor Sol Broder propone en su estudio un método probabilístico para programar exámenes con una cantidad mínima de conflictos, esto se lograba mediante una repetición de evaluaciones a un sistema de ecuaciones lineales, sin embargo, la solución planteada no era óptima [5]. En el mismo año, Cole estudiaba ETP bajo otro punto de vista, aplicando una heurística de orden, largest degree. En el paper propone una matriz de conflictos para los cursos

con el objetivo de organizar los cursos en timeslots [13].

Desde entonces el interés en los ETP solo fueron en aumento, es así que en 1986 se publica una de las investigaciones más destacadas y revisadas de la literatura. Carter realizó un completo análisis en el que discute los principales enfoques estudiados hasta aquel entonces para el problema, en otras palabras, un estado del arte organizado, además se enfoca en la aplicación de teoría de grafos para ETP, con orientación a heurísticas de coloreo de grafos [9]. Una década después, Carter en conjunto con Laporte continuaron con el estudio, el nuevo enfoque de la investigación era un análisis a las metodologías y técnicas utilizadas para ETP desde 1986 a 1996, además identificaron algunas de las restricciones blandas más comunes en las variantes de ETP. Un punto importante de la investigación es que identificaron una clasificación para los métodos: métodos de cluster, métodos secuenciales, búsqueda generalizada (metaheurística) y técnicas basadas en restricciones [10]. Por otra parte, los autores convergen en la sugerencia de que los estudios y análisis a problemas ETP deben ser reportados con resultados de prueba. Es debido a esto que presentaron un dataset público de 13 Examination Timetabling Problems del mundo real, conocidos como "the Toronto benchmarks" [11].

Avanzando en la línea cronológica, en el año 2002 surge otro importante estudio en la materia por parte de Burke y Petrovic [8]. En el estudio se muestra una revisión a los avances en investigaciones referentes a problemas de timetabling en universidades, incluyendo cursos y exámenes, poniendo en discusión grandes enfoques como algoritmos heurísticos y evolutivos, toma de decisiones multicriterio y razonamiento basado en casos. En la investigación se identificó que existía una tendencia a los enfoques híbridos y meta-heurísticos, esto es porque se descubrió que al utilizar solo heurísticas constructivas no es posible proporcionar una buena calidad de solución. Luego, en 2004, Petrovic y Burke [17] realizaron otro estudio, esta vez en aquella ocasión se enfocaron en el estado del arte de los métodos más avanzados que habían logrado buenos resultados en el área. Nuevamente se destacó los enfoques, razonamiento basado en casos e hiperheurística, metaheurísticos, multicriterio, y enfoques autoadaptativos. En esta parte de la historia, los estudios de ETP muestran el éxito de la metaheurística para encontrar soluciones de buena calidad, sin embargo, es dependiente de los parámetros utilizados, esto quiere decir que a partir de un buen ajuste de parámetros es posible encontrar buenos resultados mediante enfoques metaheurísticos.

Los siguientes años surgieron grandes cantidades de estudios interesantes, muchos de aquellos tuvieron éxito y tenían un factor en común, utilizan metodologías híbridas o enfoques no tradicionales, por lo que clasificar los enfoques se ha vuelto una tarea un poco compleja. En cuanto a los algoritmos utilizados en los estudios anteriormente mencionado (y muchos más que no fueron mencionados), es posible clasificarlos en seis grandes categorías, las que a su vez se pueden dividir en más subcategorías: enfoques exactos, basados en restricciones, técnicas heurísticas constructivas, técnicas heurísticas, metaheurísticas y de mejora, hiperheurísticas y de razonamiento basado en casos, además de las técnicas multicriterio y multiobjetivos, todas mostradas en la tabla 1.

3.1 Enfoques Exactos

Los enfoques exactos tratan sobre aplicar metodologías clásicas de búsqueda mediante herramientas matemáticas. El modelo se presenta en términos matemáticos y debe ser desarrollado, por lo que es buena opción al momento de tratar con algún problema en particular, sin embargo, la debilidad que presenta es en situaciones donde se debe resolver problemas generalizados, y es por esta razón que nunca se han aplicado tests de pruebas bajo este enfoque. Ejemplo de algoritmos de enfoque exacto son Branch and Bound, programación lineal y entera [3].

3.2 Basados en Restricciones

Destacada por Burkner y Laporte [10] en 1996 debido a la facilidad de aplicación, además de su flexibilidad para adecuarse a los problemas ETP. En este enfoque los exámenes son trabajados como variables cuyo dominio es finito, los valores de los dominios representan los periodos de tiempo y salas, estos son asignados secuencialmente para armar las soluciones. Una desventaja de este enfoque es que los algoritmos suelen ser costosos en términos computacionales, por lo que por si solo este enfoque es incapaz de entregar soluciones de alta calidad [4]. Ejemplos de este enfoque son la programación lógica de restricciones y las técnicas de satisfacción de restricciones.

3.3 Técnicas heurísticas constructivas

Son conocidos por ser iterativos, su construcción va creciendo a medida que pasan las iteraciones hasta lograr una solución. Sus ventajas son la rapidez, eficiencia y la que calidad de las soluciones son aceptables. Una heurística ejemplar de esta técnica, popular entre los mayores exponentes del área, es la basada en grafos, utilizada desde los inicios de la investigación de esta clase de problemas, donde el ETP se trabaja con un coloreo de grafos, de esta forma se ordenan los exámenes según su grado de dificultad para luego ser asignados cada iteración [12].

3.4 Técnicas heurísticas, meta-heurísticas y de mejora

Esta es una de las técnicas más utilizadas en el ámbito de los ETP por su éxito y eficacia al momento de encontrar soluciones. La metaheurística es un método de aproximación basado en heurísticas que no depende en gran medida de clase de problema que se presente y por ende, la dependencia a la configuración de parámetros es en menor grado. Petrovic y Burke [17] indican que las técnicas heurísticas funcionan a partir de un conjunto de soluciones iniciales y a partir de estas se aplican estrategias de búsqueda. En general, las técnicas metaheurísticas son clasificadas en dos categorías:

- **Búsqueda local:** el principal objetivo de estas técnicas es evitar los estanques producidos por óptimos locales. Existen muchos algoritmos que trabajan con esta técnica, algunos ejemplos más populares son Hill Climbing, Tabu Search, Simulated Annealing, Large Neighbourhood Search y Great Deluge.
- **Búsqueda basada en población:** se denomina población al conjunto de soluciones iniciales y a partir de estas se generan simultáneamente nuevas soluciones. Ejemplos de esta búsqueda aplicada en ETP son los algoritmo genéticos, algoritmos meméticos y algoritmo de hormigas.

3.5 Técnicas hiperheurísticas y de razonamiento basado en casos

Estas técnicas son propuestas para solucionar la falta de generalidad de las técnicas o enfoques antes mencionados, pues una de las desventajas que tiene las metaheurísticas es la dependencia del ajuste de parámetros que afecta de manera negativa para lograr obtener mejores soluciones de los problemas ETP. Comenzando por la hiperheurística, Burke [6] entrega la siguiente definición: "Una hiperheurística es una metodología automatizada para seleccionar o generar heurísticas para resolver problemas difíciles de búsqueda computacional". Dicho de otra forma, las hiperheurísticas es el equivalente a una heurística de alto nivel, la que busca y elige heurísticas de bajo nivel basadas en mecanismos de aprendizaje, de tal forma que permita adecuarse a problemas más generales.

Por otro lado esta el razonamiento basado en casos (CBR) [14], es un sistema que modela el proceso de conocimiento mediante el almacenamiento en memoria de casos anteriores. Los

RBC buscan resolver los problemas que se le presentan mediante el análisis de casos similares previamente almacenados en memoria para adecuarlo a las necesidades del presente problema. Sin embargo, una de las debilidades de CBR es que un mínimo cambio de parámetros entre el los casos almacenados y el caso actual puede conducir a soluciones muy diferentes, por lo que el esfuerzo y trabajo computacional de los casos previos pueden ser desperdiciados [18].

3.6 Técnicas multicriterio y multiobjetivos

Pensar los ETP bajo el enfoque multiobjetivo no es un común. El objetivo de estas técnicas es facilitar el manejo de varios criterios en paralelo para apoyar la toma de decisiones, donde cada uno es considerado una restricción que posee un grado de importancia, es así como surge el concepto de vector de restricciones. Últimamente los estudios sobre ETP optan por el enfoque multicriterio ya que se han logrado avances significativos gracias a esta técnica.

3.7 Resumen

Luego de mencionar las técnicas más relevantes para resolver Examination Timetabling Problem, se destaca la popularidad de las meta-heurísticas híbridas, como Simulated Annealing, Hill Climbing, Tabu Search y otros que utilizan estrategias de búsqueda para encontrar soluciones candidatas [16]. Para este trabajo se aplican meta-heurísticas híbridas utilizando algoritmos Greedy+Simulated Annealing, que serán probados eventualmente con algunas benchmark estandarizadas para el ETP. Las ventajas que presentan utilizar Simulated Annealing es el poder escapar de óptimos locales por medio de criterios probabilísticos, por el contrario, Greedy es incapaz de escapar de óptimos locales al solo enfocarse en buscar localmente.

4 Modelo Matemático

El modelo matemático a plantear puede variar dependiendo de la variación a considerar, el ajuste parámetros y en consecuencia el enfoque a utilizar para resolver el Examination Timetabling Problem. Para esta sección se presenta el modelo matemático de enfoque multicriterio propuesto por Petrovic Y Burke [17].

4.1 Parámetros

- t_{np} : 1 si el examen n es asignado en el periodo de tiempo p . 0 caso contrario.

4.2 Constantes

- N : Número de exámenes.
- P : Número de periodos de tiempo.
- K : Número de criterio.
- C_{nm} : Es el número de estudiantes tomando el examen n y m . Con $n, m = 1..N$
- $f_k(T)$: Valor del criterio C_k . Con $k = 1..K$.
- w_k : El peso del criterio
- C : $[C_{nm}]_{N \times N}$ Matriz simétrica que almacena los conflictos entre exámenes.
- T : $[T_{np}]_{N \times P}$ Matriz que almacena la asignación de exámenes a los periodos de tiempo.

4.3 Función Objetivo

Para este modelo matemático, la función objetivo opera de la siguiente forma:

Dado una cantidad N de exámenes y sea P la cantidad de timeslots, se busca determinar la Timetabling T que minimiza la suma ponderada del vector $WF = (w_1 f_1(T), \dots, w_K f_K(T))$:

$$\text{Min } WF = \sum_{k=1}^K w_k f_k(T) \quad (1)$$

4.4 Restricciones

Cada examen es programado sólo una vez:

$$\sum_{p=1}^P t_{np} = 1 \quad (2)$$

Dos exámenes no pueden ser asignados al mismo periodo de tiempo:

$$\sum_{n=1}^{N-1} \sum_{m=n+1}^N \sum_{p=1}^P t_{np} t_{mp} c_{nm} = 0 \quad (3)$$

5 Representación

Para este trabajo se asigna dos técnicas para resolver las diversas instancias propuestas. Primero se tiene un enfoque Greedy, está se utiliza para encontrar una solución inicial factible y en segunda instancia se utiliza Simulated Annealing (SA) para mejorar la solución, es decir, minimizar la cantidad de timeslots a utilizar.

En cuanto a la representación de la solución, se debe trabajar con algunas estructuras de datos que permitan almacenar los conflictos, las asignaciones entre exámenes y timeslots, además de vectores y/o array que permiten almacenar información, como por ejemplo, los exámenes a rendir de cada estudiante, entre otros.

- **Matriz de conflictos:** Estructura de datos simétrica de $N \times N$, siendo N la cantidad de exámenes. En esta matriz se almacena un 1 en la posición $C[n][m]$ si existe conflicto entre el examen n y el examen m , en caso contrario, donde no existe conflicto entre n y m , se almacena un 0 en $C[n][m]$. En la Tabla 3 es posible ver que el examen e_1 se encuentra en conflicto con el examen e_2 , por lo que no pueden ser agendados en el mismo timeslot.

	e_1	e_2	e_n	e_N
e_1	0	1	0	0
e_2	1	0	0	1
e_n	0	0	0	0
e_N	0	1	0	0

Table 3: Representación matriz de conflictos para los exámenes.

- **Matriz de asignación:** Matriz de tamaño $P \times N$, siendo N la cantidad de exámenes y P el número de timeslots. En esta matriz se almacena un 1 en la posición $T[p][n]$ si se asigna el examen n al timeslot p , en caso contrario se almacena un 0. En la Tabla 4 es posible notar que e_1 y e_2 están asignados a distintos bloques de tiempo, siendo el examen e_1 asignado a t_2 y el examen e_2 asignado a t_1 .

	e_1	e_2	e_n	e_N
t_1	0	1	0	0
t_2	1	0	1	1
t_p	0	0	0	0
t_P	0	0	0	0

Table 4: Representación matriz de asignación para los exámenes y timeslots.

- **Representación Solución Final :** La solución final se presenta en tres archivos. El primero lleva por nombre **PROBLEM.RES**, este archivo muestra un entero con la cantidad de timeslots asignados. El segundo output con nombre **PROBLEM.SOL**, muestra línea por línea la asignación entre exámenes y timeslots. Por último, se tiene **PROBLEM.PEN** muestra la penalización promedio de la solución final.

6 Descripción del algoritmo

Para la implementación de las técnicas Greedy y Simulated Annealing se decide utilizar el lenguaje de programación en C. Para lograr desarrollar funciones auxiliares se implementaron listas enlazadas que facilitaran el trabajo.

El primer algoritmo importante que es necesario para la construcción de la solución final es **FillMatriz**. Esta función se encarga de completar con 1 la matriz de conflictos en caso de que existan conflictos entre exámenes que deben ser rendidos por un mismo alumno, tal como se muestra en la Tabla 3. A partir del archivo de estudiantes, por cada alumno crea una lista enlazada que almacena las pruebas que dicha persona debe rendir, luego para generar los conflictos es necesario recorrer la lista comparando todos los pares existentes, por cada par se almacena un 1 en la matriz de conflictos en la casilla que representa la intersección de ambos exámenes. Cabe mencionar que en las diagonales se tienen 0 ya que una misma prueba no entra en conflicto consigo misma. En el Algoritmo 1 se muestra el pseudocódigo para esta función.

El segundo es el algoritmo **Greedy**, que busca plantear una solución inicial, de tal forma que se pueda minimizar la cantidad de timeslots a ocupar para asignar los exámenes sin que tengan un conflicto. Para la implementación del algoritmo Greedy es importante identificar la cantidad mínima de bloques de horario para poder agendar los exámenes, esto es posible sumando la cantidad de conflictos por cada fila, luego de recorrer todas las filas se obtendrá el número más grande de conflictos encontrado, como se muestra en la Tabla 5, el máximo T_c es 2, donde T_c es el total de conflictos por fila, esto quiere decir que para e_2 existen 2 conflictos y por lo tanto es necesario tener $2 + 1$ timeslots para almacenar los exámenes sin problemas. Generalizando, se debe encontrar la fila con mayor cantidad de conflictos T_c y luego la mínima cantidad de timeslots necesarios son $T_c + 1$. El segundo paso es completar la matriz de solución, esto se hace mediante listas soportes que almacenan los exámenes para cada timeslot, siempre cumpliendo la condición de que el examen que será asignado no se encuentre en conflicto con los demás exámenes ya asignados al timeslot.

	e_1	e_2	e_n	e_N	T_c
e_1	0	1	0	0	1
e_2	1	0	0	1	2
e_n	0	0	0	0	0
e_N	0	1	0	0	1

Table 5: Representación matriz de conflictos para los exámenes.

Por último se tiene el algoritmo Simulated Annealing (SA), que se centra en encontrar soluciones de mejor calidad que la entregada por Greedy. Para la función de SA se deben implementar algunas funciones auxiliares como **Movimiento()** que permite explorar diversas combinaciones de asignación de exámenes, esta función toma un examen aleatorio de la matriz solución actual y luego prueba reasignarlo a otro timeslot de forma que no exista conflicto. Por otra parte, el método **EvaluarCalidad()** toma la matriz actual y evalúa su calidad en base a la Tabla 2 que indica las penalizaciones correspondientes. Luego de realizar cierta cantidad de mejoras, que son entregadas por parámetro, y se acabe el ciclo principal, se obtendrá la matriz final con una solución de mejor calidad para nuestra instancia de ETP.

Algorithm 1 Algoritmo Completar Matriz de Conflicto

```

1: procedure FILLMATRIZ(MatrizConflicto, ArchivoEstudiante)
2:   contador_examenes  $\leftarrow$  0
3:   lista_examenes  $\leftarrow$  [ ] ▷ Buffer que guarda las pruebas de cada alumno
4:   ultimo_visitado  $\leftarrow$  'first' ▷ ID ultimo estudiante revisado
5:   for student, exam in ArchivoEstudiante do
6:     if student  $\neq$  ultimo_visitado then
7:       if contador_examenes  $\neq$  0 then
8:         if length(lista_examenes) > 1 then
9:           a  $\leftarrow$  0
10:          while a < contador do
11:            b  $\leftarrow$  a + 1
12:            valor1  $\leftarrow$  lista_examenes[a]
13:            while b < contador do
14:              valor2  $\leftarrow$  lista_examenes[b]
15:              MatrizConflicto[valor1][valor2]  $\leftarrow$  1
16:              MatrizConflicto[valor2][valor1]  $\leftarrow$  1
17:              b++ = 1
18:            end while
19:            a++ = 1
20:          end while
21:          lista_examenes  $\leftarrow$  [ ]
22:          contador_examenes  $\leftarrow$  0
23:        else
24:          lista_examenes  $\leftarrow$  [ ]
25:          contador_examenes  $\leftarrow$  0
26:        end if
27:      end if
28:      ultimo_visitado  $\leftarrow$  student
29:      lista_examenes[−1]  $\leftarrow$  exam
30:      contador_examenes++ = 1
31:    else
32:      lista_examenes[−1]  $\leftarrow$  exam
33:      contador_examenes++ = 1
34:    end if
35:  end for
36:  return MatrizConflicto
37: end procedure

```

Algorithm 2 Algoritmo Greedy

```
1: procedure GREEDY(MatrizConflicto, CantExamenes)
2:   MaxTimeSlot  $\leftarrow$  EncontrarTSM(MatrizConflicto, CantExamenes)  $\triangleright$  Encuentra
   la cantidad M nima de TimeSlot
3:   lista_solucion  $\leftarrow$  [MaxTimeSlot]
4:   matriz_solucion  $\leftarrow$  [MaxTimeSlot][CantExamenes]
5:   i  $\leftarrow$  0
6:   y  $\leftarrow$  0
7:   x  $\leftarrow$  0
8:   while i < MaxTimeSlot do
9:     lista_timeslot  $\leftarrow$  []
10:    lista_solucion[i]  $\leftarrow$  lista_timeslot
11:  end while
12:  while x < CantExamenes do
13:    conflictos  $\leftarrow$  GenerarConflictos(x, CantExamenes)
14:    while y < MaxTimeSlot do
15:      if x + 1 not in lista_soluciones[y] then
16:        if CompararListas(lista_soluciones[y], conflictos) then
17:          lista_soluciones[y]  $\leftarrow$  x + 1
18:          matriz_solucion[y][x]  $\leftarrow$  1
19:        end if
20:      end if
21:    end while
22:  end while
23:  return matriz_solucion
24: end procedure
```

Algorithm 3 Algoritmo Simulated Annealing

```
1: procedure SIMULATEDANNEALING(MatrizGreedy, MatrizConflictos alpha  
   ,CantExámenes, timeslots, mejoras)  
2:   MejorCalidad  $\leftarrow$  evaluarCalidad(MatrizGreedy)  
3:   actual  $\leftarrow$  MatrizGreedy  
4:   mejor_solucion  $\leftarrow$  MatrizGreedy  
5:   t  $\leftarrow$  0  
6:   T  $\leftarrow$  24  
7:   while x = 0 < CantExámenes do  
8:     while y = 0 < mejoras do  
9:       NewMatriz  $\leftarrow$  Movimiento(actual)  
10:      calidad_actual  $\leftarrow$  evaluarCalidad(NewMatriz)  
11:      if calidad_actual > MejorCalidad then  
12:        actual  $\leftarrow$  NewMatriz  
13:      end if  
14:      if Random(0,1) <  $\exp^{(\text{deltaCalidad}/T)}$  then  
15:        actual  $\leftarrow$  NewMatriz  
16:      end if  
17:      if calidad_actual > MejorCalidad then  
18:        MejorCalidad  $\leftarrow$  calidad_actual  
19:        mejor_solucion  $\leftarrow$  actual  
20:      end if  
21:    end while  
22:    T  $\leftarrow$  Enfriamiento(T, alpha)  
23:    t + = 1  
24:  end while  
25:  return mejor_solucion  
26: end procedure
```

7 Conclusiones

Luego de explorar las numerosas investigaciones más relevantes para tratar los Examination Timetabling Problems es posible destacar el interés creciente desde la década de los 60 hasta la actualidad, esto se ve directamente reflejado en la cantidad de enfoques y técnicas que han surgido a través de los años, muchas de ellas surgen como una versión mejorada de las que ya existen, el éxito de la hibridación de heurísticas, las metaheurísticas y las hiperheurísticas, para encontrar soluciones de alta calidad da indicios de las futuras tendencias a seguir en próximas investigaciones. Teniendo en cuenta la gran diversidad de algoritmos y técnicas existentes para resolver problemas de Timetabling, es fácil notar que cada implementación esta basada en alguna variante en particular, si bien todas las técnicas y metodologías apuntan al mismo objetivo de organizar de la manera más óptima y eficiente los exámenes en los diversos bloques horarios, estas se diferencian en tamaño, función objetivo, restricciones, variables, enfoques, implementación computacional, heurísticas aplicadas, por lo que no todas técnicas vistas en este informe resuelven el mismo problema. En particular, para la instancia estudiada en este informe se aplican dos algoritmos para encontrar soluciones de calidad, la combinación entre Greedy y Simulated Annealing puede funcionar bien, sin embargo, existen otras combinaciones de algoritmos para encontrar soluciones del mismo nivel de calidad, esta comparación es interesante y queda propuesta para un siguiente estudio. Por otra parte, es posible destacar que, actualmente existen demasiadas limitaciones en cuanto a los ETP, una de las más destacadas es la necesidad de avanzar en la búsqueda de técnicas para lograr una solución generalizada, pues las configuraciones específicas de cada problema son complejas de tratar en problemas con parámetros distintos.

Con los años se han logrado grandes avances en los problemas relacionados al Timetabling, esto permite a los nuevos investigadores tener bases más solidas para proponer ideas innovadoras y potenciar el avance en esta área de estudio tan compleja y necesaria para las organizaciones educacionales al rededor del mundo.

8 Bibliografía

References

- [1] Leena Ahmed, Ender Özcan, and Ahmed Kheiri. Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications*, 42:5463–5471, 08 2015.
- [2] Masri Ayob, Ariff Malik, Salwani Abdullah, Abdul Hamdan, Graham Kendall, and Rong Qu. Solving a practical examination timetabling problem: A case study. pages 611–624, 08 2007.
- [3] Eva Barrena, David Canca, Leandro Coelho, and Gilbert Laporte. Exact formulations and algorithm for the train timetabling problem with dynamic demand. *Computers and Operations Research*, 44:66–74, 04 2014.
- [4] S.C. Brailsford, C.N. Potts, and B.M. Smith. Constraint satisfaction problems: algorithms and applications. Discussion paper, 1998.
- [5] Sol Broder. Final examination scheduling. *Commun. ACM*, 7(8):494–498, August 1964.
- [6] Edmund Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and John Woodward. *A Classification of Hyper-heuristic Approaches*, volume 146, pages 449–468. 09 2010.
- [7] Edmund Burke, Barry Mccollum, Amnon Meisels, Sanja Petrovic, and Rong Qu. A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176:177–192, 01 2007.
- [8] Edmund Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140:266–280, 07 2002.
- [9] Michael W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2):193–202, 1986.
- [10] Michael W. Carter and Gilbert Laporte. Recent developments in practical examination timetabling. In *Selected Papers from the First International Conference on Practice and Theory of Automated Timetabling*, page 3–21, Berlin, Heidelberg, 1995. Springer-Verlag.
- [11] Michael W. Carter, Gilbert Laporte, and Sau Yan Lee. Examination timetabling: Algorithmic strategies and applications. *The Journal of the Operational Research Society*, 47(3):373–383, 1996.
- [12] Manuel Chica, Oscar Cordon, S. Damas, Joaquín Bautista-Valhondo, and Jordi Pereira. Heurísticas constructivas multiobjetivo para el problema de equilibrado de líneas de montaje considerando tiempo y espacio. pages 649–656, 02 2009.
- [13] A. J. Cole. The preparation of examination time-tables using a small-store computer. *The Computer Journal*, 7(2):117–121, 01 1964.

- [14] D.B. Leake. Problem solving and reasoning: Case-based. In Neil J. Smelser and Paul B. Baltes, editors, *International Encyclopedia of the Social and Behavioral Sciences*, pages 12117–12120. Pergamon, Oxford, 2001.
- [15] Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30:167–190, 01 2008.
- [16] Belen Melian, José Moreno-Pérez, and J. Moreno-Vega. Metaheuristics: A global view. *Inteligencia Artificial, revista Iberoamericana De Inteligencia Artificial - AEPIA*, 7, 07 2003.
- [17] Sanja Petrovic and Edmund Burke. University timetabling. In *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 45. Chapman Hall/CRC Press, 2004.
- [18] Sanja Petrovic, Yong Yang, and Moshe Dror. Use of case based reasoning in solving examination timetabling problems. *European Journal of Operational Research - EJOR*, 01 2004.
- [19] Rong Qu, E. Burke, Barry Mccollum, Liam Merlot, and S. Lee. A survey of search methodologies and automated approaches for examination timetabling. *Journal of Scheduling - SCHEDULING*, 01 2006.
- [20] Mario César Velez and José Alejandro Montoya. METAHEURÍSTICOS: UNA ALTERNATIVA PARA LA SOLUCIÓN DE PROBLEMAS COMBINATORIOS EN ADMINISTRACIÓN DE OPERACIONES. *Revista EIA*, pages 99 – 115, 12 2007.
- [21] Ender Özcan, Mustafa Misir, Gabriela Ochoa, and Edmund Burke. A reinforcement learning - great-deluge hyper-heuristic for examination timetabling. *International Journal of Applied Metaheuristic Computing*, 1:39–59, 01 2010.