

# Inteligencia Artificial

## Informe Final: Examination Timetabling Problem

Clemente Donoso Krauss

1 de agosto de 2021

### Evaluación

Mejoras 1ra Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (15 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
<b>Nota Final (100):</b>	_____

### Resumen

El Examination Timetabling Problem (ETP), es un problema que busca dar solución a la asignación de exámenes sobre un conjunto de horarios bajo ciertas restricciones que se planteen. Los alcances y variaciones de éste problema que van desde asignar exámenes en una universidad, hasta asignar las fechas de los partidos de fútbol u otros deportes, hacen del ETP un problema bastante estudiado en la historia y sumamente importante trabajarlo. El documento presenta el problema desde sus orígenes, la necesidad de estudiarlo, un recorrido sobre alguna de las soluciones propuestas en el último tiempo. También se presenta un modelo matemático que busca solucionar el ETP de manera novedosa con las conclusiones pertinentes. Finalmente se invita al lector a involucrarse en el estudio y presentar sus propias conclusiones sobre el ETP y la solución presentada.

## 1. Introducción

En la actualidad, gran parte de las universidades y centros de estudios han optado por evaluar el conocimiento de los estudiantes a partir de evaluaciones y exámenes que engloban toda la materia vista en un semestre, es por ello, que cobra importancia para estos centros educativos que aseguren las condiciones para realizar una buena evaluación, ya que mientras mejores condiciones se brinden para su rendición, mejores resultados podrán esperarse [15]. Los factores que se pueden nombrar que influyen en estas condiciones son múltiples, entre ellos se pueden nombrar, que los horarios asignados a los estudiantes no se solapen (ya que un estudiante no puede rendir más de un examen al mismo tiempo), y a su vez, cada estudiante necesitará de

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

Cuadro 1: Penalización por estudiante [15]

un tiempo de descanso o preparación entre exámenes [10]. Toda institución educacional debe realizar esta asignación de horarios, por lo que dado la cantidad de exámenes y alumnos que dichas instituciones tienen, es una tarea que consume bastante tiempo y hacerlo de manera manual es contraproducente. Por lo que se debe contar con alguna técnica que automatice dicho proceso, sin embargo, dada las variaciones de curriculum, de estudiantes e incluso el atraso o adelantos de estos en la malla curricular, hacen de éste problema uno complejo y de gran relevancia para las universidades o entidades educativas, tanto para tenerlo encuenta, como también resolverlo.

El problema descrito anteriormente se le conoce comúnmente como Examination Timetabling Problem, el cual es una variante del problema Timetabling Problem (TP), dicho problema, considera el organizar un conjunto de actividades en bloques de horario, por lo que el ETP se define, como dice su nombre, el problema que tiene como objetivo organizar o calendarizar un conjunto de exámenes de cursos universitarios, evitando a toda costa el traslape de aquellos exámenes de los cursos que tengan estudiantes en común, y de esta manera, lograr distribuir de la mejor forma posible los exámenes [19]. Los TP, como los ETP forman parte del conjunto de problemas NP-completos, esto quiere decir, que no son posibles de resolver en un tiempo polinomial [22], por otra parte, dado la diversidad de instituciones, planes de estudios y estudiantes que existen, los requerimientos a la hora de asignar los exámenes a sus horarios serán múltiples, sin embargo, éste grupo de requerimientos se puede separar en 2 tipos de restricciones [18], tendremos las restricciones duras, las cuales siempre deben ser satisfechas (ver Cuadro 2), y por otro lado, tendremos las restricciones blandas, las cuales deben satisfacerse en el mayor grado posible para que de esta manera se logre dar con una mejor solución (ver Cuadro 3).

En el presente documento, se trabajará con una variante del ETP, la cual tiene como restricción dura que no están permitidos los toques de horario, como también, existirá una penalización por estudiante la cual dependerá de la cantidad de bloques que separan a los exámenes de un mismo estudiante (ver Cuadro 1), también para simplificar los cálculos tanto las salas como su capacidad será infinitas. Por lo que el problema ETP presentado tiene los siguientes dos objetivos

1. Minimizar los bloques de horario o tener la cantidad mínima de timeslot para los exámenes (**objetivo principal**)
2. Minimizar la penalización o mejorar la distribución de los exámenes (**objetivo secundario**)

Se ha mostrado con el estudio del problema ETP, que el minimizar los timeslot, por su parte también minimiza, en gran parte de los casos, la penalización promedio por estudiante, por lo que el desarrollo del problema, busca dar solución al objetivo principal.

Dado que el ETP se trata de un problema NP-Completo, el impacto que éste tiene no solo en las instituciones educativas, sino en gran parte de las organizaciones deportivas, hacen de éste problema uno interesante de estudiar y sobretodo, fundamental de resolver universalmente.

	<b>Restricción</b>
1.-	Los exámenes no deben compartir alumnos de manera simultánea.
2.-	Las salas deben ser suficientes, tanto en cantidad como en capacidad

Cuadro 2: Restricciones duras en ETP [18]

	<b>Restricción</b>
1.-	Los exámenes que tienen mismo horario, deben distribuirse de manera uniforme
2.-	Existen grupos de exámenes que deben tomarse al mismo tiempo
3.-	Existen exámenes que deben ser consecutivos
4.-	Exámenes extensos deben tomarse al inicio del día
5.-	Exámenes que se solapan deben rendirse en horarios consecutivos
6.-	Exámenes iguales pueden rendirse en ubicaciones distintas
7.-	Exámenes distintos solo si tienen misma duración, pueden rendirse en la misma ubicación

Cuadro 3: Restricciones blandas en ETP [18]

Finalmente, la estructuración del presente documento está bajo la siguiente forma: En primera instancia se define el problema ET y sus variantes, y a su vez, se precisa y detalla el problema ETP en la sección 2; luego se habla sobre la historia del problema, desde sus inicios y como se han planteado diversas soluciones en la sección 3; por otro lado, una vez presentada la historia, se determina un modelo matemático, con sus variables y restricciones que permita entregar una solución al problema ETP en la sección 4; por último se presenta alguna de las conclusiones desarrolladas sobre el trabajo realizado, como también se deja una invitación al lector para reflexionar sobre el futuro del problema ETP en la sección 9

## 2. Definición del Problema

Como se dijo anteriormente, el Examination Timetabling Problem es un tipo de Timetabling Problem el cual está centrado en la calendarización de exámenes. Los TP tienen 4 componentes principales: Un conjunto de bloques de tiempos, un conjunto de recursos, un conjunto de actividades y un conjunto de restricciones [18], en el cual para un ETP, se tiene que el conjunto de recursos son los estudiantes, el conjunto de actividades son los exámenes y las restricciones son que un alumno no puede dar más de un examen al mismo tiempo, de esta manera generar y organizar un calendario de exámenes donde las restricciones se satisfagan. Modelando de esta manera el problema, se puede plantear el ETP como un problema de búsqueda, satisfaciendo siempre las restricciones del problema, sin embargo, esta optimización de restricciones y de la función objetivo, hacen del problema ETP un problema de optimización, más que un problema de búsqueda.

Existen variaciones del problema ETP como instituciones educativas en el mundo, sin embargo, todas estas buscan satisfacer el mismo objetivo principal, el cual es minimizar los bloques de horario en los cuales se deberán rendir los exámenes, por lo que también se han planteado diversas alternativas para solucionar este problema, algunas de ellas son la realización de diversas versiones de examen [15], también se puede encontrar en la literatura, alternativas donde se considera que existen un conjunto de ubicaciones para rendir el examen y los alumnos pueden agendar sus preferencias [7], entre otros ejemplos. Estos casos muestran la diversidad de versiones que se pueden encontrar en la literatura y las aplicaciones en la actualidad que comprende éste problema.

También dentro de los estudios realizados, se puede encontrar una cantidad no menor de Timetabling Problem, sin embargo, dentro de estos, existen dos que están directamente relacionadas con el ETP: Timetabling Course Problem (Se debe organizar un conjunto de cursos en salas y horarios) y el School Timetabling Problem (Se debe organizar un conjunto de profesores, horarios, estudiantes, entre otros a un conjunto de actividades) que no solo se asemejan al Examination Timetabling Problem, sino también, han aportado importantes avances y novedosos puntos de vista para enfrentar y dar solución al ETP.

El problema que se plantea en el documento, es aquel ETP que cuenta con un conjunto de exámenes finitos, y una cantidad finita de alumnos que deben rendir dichos exámenes. Para facilitar el desarrollo del problema se supone que las salas y la capacidad de éstas son infinitas, para así, buscar minimizar la cantidad de timeslot necesarios para que todos los alumnos puedan rendir sus exámenes correctamente (que ningún estudiante tenga dos exámenes al mismo tiempo) y, siguiendo la línea anterior, también se debe disminuir la penalización por estudiante, es decir, aumentar el tiempo entre exámenes por alumno, siempre teniendo en cuenta que el objetivo principal es el minimizar la cantidad de bloques requeridos para rendir todos los exámenes.

### 3. Estado del Arte

A lo largo de la historia, las primeras apariciones del problema ETP se pueden encontrar en los años 60's donde el autor Broder [1] en el año 1964, por primera vez, presenta y modela el problema dada la necesidad de organizar los exámenes finales de una universidad, donde la solución propuesta era un modelo probabilístico, el cual minimiza los conflictos entre los exámenes, sin embargo, con el tiempo se demostró que si bien era óptima la solución encontrada, no era la mejor. Aquel año, también se propuso una matriz, en la cual se representa los conflictos entre los exámenes y las restricciones pertinentes, para que de esta manera en conjunto con un algoritmo de coloreo de grafos ("largest degree first: fill from top") se genere una solución también óptima [5]

Dos años más tarde, en el año 1966 se propone un nuevo algoritmo de coloreo de grafos, el cual plantea organizar de mejor manera los exámenes, sin embargo, esta solución dejaba totalmente de lado el objetivo principal del ETP [17]. Es por ello, que en el año 1968, se plantea un nuevo algoritmo, que a partir de la matriz generada por el algoritmo de Cole [5], realizaba una organización óptima para gran parte de los exámenes, luego aquellos exámenes que no se pudiesen agendar con el algoritmo, eran reagendados utilizando la técnica "look ahead" para determinar la mejor asignación, es decir, el algoritmo se encargaba de organizar la mayor parte de los exámenes, y aquellos con problemas, eran calendarizados manualmente siguiendo esta técnica. Debido a la complejidad que tenía éste algoritmo y al crecimiento exponencial que ocurría al agregar más exámenes, no se podía utilizar para toda la universidad, sino que, se debía utilizar para grupos de exámenes más pequeños [24].

Años más tarde, en los años 80's se desarrolla el algoritmo HOREX, el cual utiliza tanto las técnicas de coloreo de grafo desarrollado anteriormente, como también, una nueva forma de enfrentar el problema utilizando variables de decisión enteras o binarias. Éste algoritmo, en comparación a los presentados en años anteriores, entregaba la mejor solución cuando era aplicado a grupos pequeño de exámenes [8]. Tiempo después, se desarrolla a partir del algoritmo presentado por Desroches (HOREX), un nuevo algoritmo que buscaba dar mayor solución a las restricciones blandas del problema, logrando su objetivo, sin embargo, al igual que HOREX, su buen desempeño solo se debía al utilizar pequeñas cantidades de exámenes. [23].

Ya a mediados de los años 80's, en 1984, por primera vez se habló sobre la preferencia que tienen los estudiantes a la hora de elegir sus cursos (exámenes), y también de la penalización por tener exámenes en bloques seguidos o costos de proximidad, es por ello, que a las restricciones duras

que se tenían anteriormente, se le sumarían estas dos. El algoritmo presentado para dar solución a esta nueva variante, fue propuesto por Laporte y lo llamaron HORHEC (una variante de HOREX), el cual dejó de lado las heurísticas de coloreo de grafo, y planteó una nueva forma de enfrentar el problema utilizando la técnica "polynomially bounded backtracking", de esta manera al aplicarlo, mejoraba las soluciones encontradas anteriormente [11].

Si bien todos estos modelos y planteamientos del problema ETP entregaban una solución óptima y fueron mejorando con el paso del tiempo, estuvieron siempre centrados en dar solución a organizaciones en específico, es decir, a los exámenes de instituciones en específica, sin tomar en cuenta los avances en paralelo que estaba ocurriendo, por lo que era imposible extrapolar las soluciones encontradas e incluso, comparar los distintos algoritmos realizados a la fecha [3]. Por lo que el aporte de Carter en 1996 fue fundamental en la creación de un algoritmo único para resolver los ETP, ya que estudió y analizó los puntos de vista mencionados (obteniendo sus ventajas y desventajas entre sí), y que si bien no podían ser utilizados en términos generales, introdujeron el problema y de esta manera se generaron las bases para poder comparar futuros algoritmos y analizar las soluciones propuestas [4].

Dentro de las técnicas utilizadas para resolver los ETP, se pueden destacar las siguientes 6 estrategias:

### **3.1. Estrategias basadas en Grafos:**

Diversos autores han utilizado la teoría de grafos y las técnicas de coloreo de grafos para presentar una resolución al problema ETP o versiones del TP como la calendarización de ligas deportivas (recordemos que la asignación de estudiante con exámenes, se puede modelar usando grafos bipartitos) [21], las cuales utilizando también algoritmos de inteligencia artificial como machine learning, han permitido optimizar las soluciones encontradas por el algoritmo. Éste conjunto de estrategias para abordar el problema ETP han generado una de las mejores soluciones actuales a dicho problema al utilizar grandes volúmenes de exámenes.

### **3.2. Estrategias Multi-Criterio:**

Un punto de vista novedoso y alternativo a la hora de enfrentar el problema ETP fue la implementación de las estrategias Multi-Criterio, las cuales optaban por dejar de lado la optimización de la función objetivo, mientras que se centraban en calcular la importancia de cada restricción. De esta manera el problema ETP se pudo enfrentar con mayor flexibilidad, sin embargo, solo en algunos casos fueron buenos o decentes los resultados entregados por esta estrategia [2].

### **3.3. Estrategias basados en Población:**

Los principales algoritmos basados en población, fueron los algoritmos de colonia de hormigas, estos algoritmos parten como un planteamiento desde la teoría de grafos, el cual utilizando técnicas de probabilidad deberían permitir encontrar la mejor solución [9], sin embargo, no se obtuvieron resultados significativos inicialmente, es por ello, que se crearon diversas versiones como algoritmos genéticos [16] para mejorar la optimización de esta técnica, pero no se logró el objetivo. Independiente de los resultados obtenidos, al implementar los algoritmos basados en población se pudo no solo generar la mejor representación del problema ETP, sino también en avanzar en crear la mejor heurística para dar solución al problema.

### **3.4. Estrategias de Satisfacción de Restricciones:**

Es considerada una de las clásicas estrategias a la hora de resolver el problema ETP debido a la flexibilidad que entregaba el enfrentar dicho problema con esta técnica, sin embargo, seguía

existiendo el principal problema, es decir, el crecimiento exponencial que tenía el problema a la hora de aumentar el grupo de exámenes. Por lo que esta técnica fue desechada rápidamente para entregar una solución favorable por si sola, no obstante, ésto no fue impedimento para que se pudiesen utilizar en conjunto con otras técnicas y así mejorar su rendimiento [6].

### 3.5. Estrategias de Búsqueda Local:

Las técnicas de búsqueda local, utilizan la función objetivo para verificar la calidad de las soluciones generadas, por lo que tienen facilidades a la hora de trabajar con las restricciones del problema. Dentro de las técnicas de búsqueda local más utilizadas se pueden encontrar:

- **Búsqueda por fases**, la cual a medida que se obtiene una solución, agrega una restricción que no haya sido utilizada anteriormente.
- **Tabu Search**, en donde la función objetivo se modifica de tal manera que se adapte al problema.
- **Asignación de prioridades**, la cual a partir de las restricciones que tenía el problema se le asignaba una prioridad.

Si bien la implementación de estas técnicas no entregó mejores resultados, sentó las bases para las técnicas híbridas, en las cuales se puede destacar la técnica híbrida entre Simulated Annealing y Backtracking, la cual mejoró notablemente los resultados obtenidos [20]. Otros autores, utilizaron esta idea para implementar sus versiones de algoritmos híbridos, por lo que en el año 2003, Merlot junto a Boland, Hughes y Stuckey utilizaron dicha técnica en conjunto con las técnicas de Satisfacción de Restricciones, obteniendo los mejores resultados (hasta esa fecha) a la hora de enfrentar el problema ETP [13].

### 3.6. Estrategias de Hiperheurísticas:

Las heurísticas de un problema son definidas como las bases o lineamientos en que se debe realizar una solución, sin embargo, dada la gran variedad de ETP que se pueden encontrar en la literatura, el planteamiento de heurísticas o meta-heurísticas se tornó difícil, por lo que fue necesario plantear hiperheurísticas para resolver el problema [12]. Es por ello, que en las últimas dos décadas se ha trabajado incesantemente en dar con una hiperheurística para el problema, donde el mayor avance se tuvo el año 2019, en que a partir del algoritmo Great Deluge se obtuvieron resultados impecables con respecto a las demás técnicas utilizadas, dando como consecuencia que las hiperheurísticas se establecieran como una de las mejores técnicas para enfrentar al ETP [15].

## 4. Modelo Matemático

Para poder presentar una solución al problema EPT con las especificaciones nombradas, se requiere realizar un modelo matemático tal que en primera instancia busque minimizar la cantidad de bloques necesarios para agendar la totalidad de exámenes, y que a su vez considere la penalización por estudiantes (el costo asociado por proximidad entre los exámenes). Para ello se utilizarán diversas herramientas como matriz de conflicto entre los exámenes y vectores correspondientes a los timeslot a utilizar.

### 4.1. Función Objetivo:

$$\min F : \sum_{m=1}^M t_m + \sum_{p=0}^5 \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^N (X_{im}X_{i(j+p)} + X_{im}X_{i(j-p)})C_{ij}W_p \quad (1)$$

Ecuación 1: Consta de dos partes, la primera sumatoria toma en cuenta la totalidad de timeslot utilizados (objetivo principal) , y la segunda parte toma en cuenta la penalización de los estudiantes según lo establecido (objetivo secundario).

#### 4.2. Variables:

- $x_{im} = \begin{cases} 1 & \text{si el examen } i \text{ se rinde en el horario } m \\ 0 & \text{caso contrario} \end{cases}$
- $t_m = \begin{cases} 1 & \text{si el horario } m \text{ es utilizado en la solución} \\ 0 & \text{caso contrario} \end{cases}$

#### 4.3. Constantes:

- $N$ : Número de exámenes
- $C_{ij}$ : Número de conflictos entre el examen  $i$  y  $j$
- $E_i$ : Número de estudiantes que darán el examen  $i$
- $W_p$ : Penalización por estudiantes
- $M$ : Cantidad máxima de horarios

#### 4.4. Restricciones:

$$\sum_{i=1}^N \sum_{m=1}^M x_{im} h_m \leq N \quad (2)$$

Ecuación 2: Restricción que considera que la cantidad de exámenes rendidos no puede superar la totalidad de exámenes.

$$\sum_{m=1}^M x_{im} t_m = 1 \quad \forall i \in \{1, \dots, N\} \quad (3)$$

Ecuación 3: Restricción donde el examen solo puede estar asignado a uno y solo un horario.

$$x_{im} + x_{jm} \leq 1 \quad \forall i, j \setminus i < j \wedge i, j \in \{1, \dots, E\} \wedge m \in \{1, \dots, M\} \quad (4)$$

Ecuación 4: Restricción la cual hace alusión a que dos exámenes no pueden compartir el mismo timeslot, es decir, no puede existir conflicto entre ambos exámenes.

#### 4.5. Consideraciones:

Es importante realizar las siguientes consideraciones al momento de plantear el modelo:

- $1 \leq i \leq N$
- $1 \leq m \leq M$
- $N \geq 1$
- $L \geq 1$
- $W_0 = 16, W_1 = 8, W_2 = 4, W_3 = 2, W_4 = 1, W_{5+} = 0$

## 5. Representación

La técnica asignada para resolver el problema fue utilizar un algoritmo Greedy para obtener una solución inicial, y luego utilizar un algoritmo HillClimbing con mejor mejora para mejorar la solución encontrada anteriormente. Para resolver esto, se utilizaron diversas técnicas matemáticas como las siguientes:

- **Matriz de Conflictos:** Tanto para encontrar la solución inicial, como para mejorar dicha solución, se utilizó una representación del problema en una matriz de conflictos, donde las columnas y filas son los exámenes. En caso de que un alumno debiese rendir dos o más exámenes, eso implica que dichos exámenes están en conflicto y no pueden ser agendados en el mismo horario, por lo que, en la matriz de conflictos, dicha intersección se marcó con un 1, en caso de no existir conflictos se marcó con un 0.
- **Vectores Auxiliares:** A lo largo del problema se puede observar que se utilizaron diversos vectores para facilitar los cálculos, por ejemplo, vectores que almacenan el cruce de los estudiantes con sus exámenes, vectores que almacenan los exámenes agendados, entre otras utilidades.

**La solución:** El formato de la solución, más allá de que el algoritmo genera archivos los archivos correspondientes, está dado por un vector bidimensional, cuyas filas corresponden a los timeslot, y las columnas los exámenes agendados en dicho timeslot.

## 6. Descripción del algoritmo

Para resolver el problema del Examination Timetabling Problem, se utilizaron además de las funciones que se detallan a continuación, una serie de funciones auxiliares para facilitar el trabajo con los vectores y matrices correspondientes.

Los pasos en como el problema es resuelto son los siguientes:

- Se inicializa generando una matriz de conflictos entre los exámenes, a partir de que si un alumno debe realizar X exámenes, entonces dichos exámenes no pueden estar ninguno compartiendo el timeslot.
- La solución inicial se utiliza usando un algoritmo Greedy, el cual a partir de la matriz de conflictos, va instanciando los exámenes a medida que va recorriendo, teniendo en cuenta que no puede instanciar en un mismo timeslot exámenes que tengan conflicto.
- El algoritmo HillClimbing, utiliza tanto la información de la matriz de conflictos, como la del algoritmo Greedy, como también la información de funciones auxiliares para estimar cual es el mejor examen que debe ser asignado nuevamente. En base a una serie de resultados se observó que mientras antes se instancien los exámenes más conflictivos, mejor será el resultado final.
- Para calcular la penalización promedio por estudiante, solo se tomaron en cuenta aquellos alumnos que debían rendir 2 o más exámenes, y utilizando la tabla 1 para encontrar la penalización correspondiente por estudiante. Luego la suma de dichas penalizaciones se divide por la cantidad de estudiantes para obtener la penalización promedio total.

Recordemos que HillClimbing propiamente tal no tiene un componente de diversificación, por lo que la representación y como se elijan que movimientos hacer será primordial para obtener la solución con mejor calidad posible, como también la solución inicial, es por ello, que durante la ejecución del algoritmo se puso un fuerte énfasis en como iniciar el recorrido del vecindario.



---

**Algorithm 1** Algoritmo Generar Matriz de Conflictos

---

Básicamente el algoritmo que genera la matriz de conflicto, recorre a cada estudiante, y por cada estudiante genera un vector con sus exámenes. Luego recorre dicho vector dos veces, para generar poblar la matriz de conflictos con 1 o 0 según corresponda.

**Input:** nombre del archivo de texto, cantidad examenes

**Output:** Matriz de conflictos

```
E ← cantidad de examenes
matrizConflictos[ExE] ← Vacía
while no se ha terminado el archivo do
  for linea del archivo do
    vectorExamenes ← Vacío
    if Alumno no ha cambiado then
      agregar examenes a vectorExamenes
    end if
    if Alumno cambio then
      for examen in vectorExamenes do
        poblar matriz con 1 en las intersecciones
        poblar matriz con 0 donde no hay intersecciones
      end for
    end if
  end for
end while
```

---

---

**Algorithm 2** Algoritmo Greedy

---

El algoritmo Greedy corresponde a tomar la información de la matriz de conflictos, e ir poblando el vector de timeslot según corresponda, esto quiere decir, que solo puede agendar en un mismo timeslot aquellos exámenes cuya intersección en la matriz de conflictos tengan un 0.

**Input:** matrizConflictos, cantidadExamenes

**Output:** cantidad de timeslot

```
E ← cantidad de examenes
vectorSolución[E'xE] ← vacío
for examen in matrizConflictos do
  tSlot ← 0
  for conflicto in matrizConflictos[examen] do
    verificar que el examen no haya sido agregado
    if examen presenta conflicto then
      tSlot ← ++
    end if
  end for
  Agregar examen al vector solución según corresponda
end for
```

---

---

**Algorithm 3** Algoritmo HillClimbing - Mejor Mejora

---

El algoritmo que se presenta, tiene como objetivo mejorar la solución encontrada por el algoritmo Greedy. Para ello utilizando una función auxiliar la cual retorna un vector con el examen y la cantidad de conflictos, los cuales están ordenados de menor a mayor cantidad de conflictos. Con la combinación de la matriz de conflictos, el vector generado por el algoritmo greedy y el vector generado por la función auxiliar, se plantea ir tomando los exámenes más conflictivos y agregándolos a nuevos timeslot de tal manera de disminuir los timeslot finales.

**Input:** lista conflictos, matriz de conflictos, cantidad de exámenes, nombre del archivo a generar y nombre del archivo de estudiantes,

**Output:** Cantidad de timeslot utilizados.

```
solucionMejor  $\leftarrow$  True
timeSlotMax  $\leftarrow$  0
vectorSolucion  $\leftarrow$  vacío
ultimoValor  $\leftarrow$  vacío

while no se vacía la lista conflictos do
    ultimoValor  $\leftarrow$  ultimo valor lista conflictos
    for i = cantidad de exámenes do
        agregarEx  $\leftarrow$  True
        for j = cantidad de exámenes do
            if Presenta conflicto esos exámenes then
                agregarEx  $\leftarrow$  False
                break
            end if
        end for
        if agregarExamen then
            se agrega el examen más conflictivo en el timeslot correspondiente.
        end if
    end for
    comparar la cantidad de timeslot utilizados
end while
```

---

## 7. Experimentos

Durante la construcción y ejecución del código para resolver el problema se encontraron bastantes dificultades. En primera instancias, los problemas entregados al tener una gran cantidad de estudiantes / exámenes, hacía imposible ir testeando el código, por lo que se decidió crear instancias propias con una menor cantidad de variables (como máximo 10) y de esta manera darle solución al problema. Algo sumamente interesante que ocurrió, es que por el hecho de trabajar con una matriz de conflictos cuadrada, cuya dimensión depende únicamente de la cantidad de exámenes que se requieren organizar, solo se permite dar solución a los problemas que tengan como máximo 720 exámenes, ya que una matriz de 721 o más filas/columnas genera un error de memoria. Por otra parte, se observa tal y como estudios lo contemplaron, que el hecho de minimizar los timeslot, conlleva una minimización en la penalización promedio de los estudiantes. Otro dato interesante, es que aquellos problemas que tienen menos estudiantes, estadísticamente, conllevan una mayor penalización.

Los problemas que se utilizaron para experimentar, además de los contruidos personalmente, fueron los siguientes:

**Carleton91, Carleton92, EarlHaig83, EdHEC92, LSE91, St.Andrews83. TorontoAS92, TorontoE92, Trent92 y YorkMills83**

Durante la ejecución del código y la experimentación, se observó que al momento de mejorar la

solución, primer modificar aquellos exámenes que tienen mayor cantidad de conflictos, conlleva a una mejor solución (en términos de timeslot), por lo que el algoritmo HillClimbing utiliza dicha información para testear la solución con ellos inicialmente.

Otro punto clave es que los movimientos utilizados para generar una nueva solución, corresponden a buscar en el vecindario de la solución inicial (y las soluciones que se van generando), sin embargo, el vecindario de las soluciones se obtienen de tal manera de mover un examen a cualquier timeslot, tal que disminuya la cantidad de timeslot totales (esto ya que se produce un efecto cascada al momento de cambiar un examen de lugar).

## 8. Resultados

A partir de los experimentos realizados, se encontró que utilizar el algoritmo HillClimbing mejora bastante la solución encontrada por el algoritmo Greedy, los resultados por cada experimento se observan en la siguiente tabla:

Nombre del archivo	Total timeslot Greedy	Total timeslot HC	Penalización promedio
<b>Carleton91</b>	168	33	2
<b>Carleton92</b>	177	32	1
<b>EarlHaig83</b>	67	25	6
<b>EdHEC92</b>	24	19	3
<b>LSE91</b>	126	18	4
<b>St.Andrews83</b>	57	12	11
<b>TorontoAS92</b>	215	35	1
<b>TorontoE92</b>	30	10	7
<b>Trent92</b>	47	21	2
<b>YorkMills83</b>	62	22	5

## 9. Conclusiones

Debido a la importancia que tiene el problema TP y los alcances tanto en las instituciones educativas, como también en las ligas deportivas y centros de salud, es un problema que ha sido bastante estudiado en la literatura y en todas sus versiones. Específicamente el ETP desde que fue propuesto por primera vez, se han planteado un centenar de soluciones, sin embargo, gran parte de las soluciones planteadas son propuestas para instituciones en específico o en grupos pequeños de exámenes, por lo que no son aplicables universalmente. Por otra parte, la representación del problema es fundamental a la hora resolverlo, esto ya que, el espacio de búsqueda está relacionado directamente con la representación a utilizar. Por otra parte, HillClimbing al tratarse de un algoritmo reparador, su espacio de búsqueda son las soluciones candidatas, por lo que la solución encontrada si bien será rápida, no se asegura que sea de buena calidad.

Cabe destacar, que el aporte que han hecho dichas alternativas, ha aportado nuevos puntos de vista y avances a la hora de tratar con el ETP, de esta manera, las mejores soluciones se pueden llegar a ajustar realizando algunas modificaciones a los algoritmos presentados. Si bien actualmente no se tiene una solución estándar, se invita al lector a profundizar en las técnicas de hiperheurísticas en conjunto con otras estrategias, debido al alto grado de eficacia que presentan, y a utilizar las nuevas tecnologías que se van creando, por ejemplo, la computación en paralelo para optimizar la entrega de la solución [14].

## 10. Bibliografía

### Referencias

- [1] Sol Broder. Final examination scheduling. *Commun. ACM*, 7(8):494 – 498, 1964.
- [2] Erben Wilhelm Burke, Edmund. Practice and theory of automated timetabling iii. *Springer*, 1:18 – 27, 2000.
- [3] Michael W. Carter. A survey of practical applications of examination timetabling algorithms. *Computers & Operations Research*, 34(2):193 – 202, 1986.
- [4] 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.
- [5] A. J. Cole. The preparation of examination time-tables using a small-store computer. *The Computer Journal*, 7(2):117 – 121, 1964.
- [6] Philippe David. Practice and theory of automated timetabling ii. *Springer*, 1:169 – 186, 2003.
- [7] Sylvain Desroches, Gilbert Laporte, and Jean-Marc Rousseau. Horex: A computer program for the construction of examination schedules. *INFOR: Information Systems and Operational Research*, 16(3):294 – 298, 1978.
- [8] Sylvain Desroches, Gilbert Laporte, and Jean-Marc Rousseau. Horex: A computer program for the construction of examination schedules. *Information Systems and Operational Research*, 16(3):294 – 298, 1978.
- [9] K. A. Dowsland and J. M. Thompson. Ant colony optimization for the examination scheduling problem. *Operational Research Society Palgrave Macmillan Journals*, 56(4):426 – 438, 2015.
- [10] Jeroen Beliën Stefan Creemers Gert Woumans, Liesje De Boeck. A column generation approach for solving the examination-timetabling problem. *European Journal of Operational Research*, 253:178 – 194, 2016.
- [11] Gilbert Laporte and Sylvain Desroches. Examination timetabling by computer. *Computers & Operations Research*, 11(4):351 – 360, 1984.
- [12] Ahmed Kheiri Leena N. Ahmed, Ender Özcan. Solving high school timetabling problems worldwide using selection hyper-heuristics. *ELSEVIER*, 42:5463 – 5471, 2015.
- [13] Natasha Merlot, Liam T. G. and Boland, Barry D. Hughes, and Peter J. Stuckey. Practice and theory of automated timetabling iv. *Springer*, 1(0):207 – 231, 2003.
- [14] John Nickolls Joshua Anderson Jim Hardwick Scott Morton Everett Phillips Yao Zhang Vasily Volkov Michael Garland, Scott Le Grand. Parallel computing experiences with cuda. *IEEE*, 28(4):13 – 27, 2008.
- [15] Ahmad Muklason, Gusti Bagus Syahrani, and Ahsanul Marom. Great deluge based hyper-heuristics for solving real-world university examination timetabling problem: New data set and approach. *Procedia Computer Science*, 161:647 – 655, 2019.
- [16] W. Banzhaf N. Pillay. An informed genetic algorithm for the examination timetabling problem. *ELSEVIER*, 10:457 – 467, 2009.

- [17] J. E. L. Peck and M. R. Williams. Algorithm 286: Examination scheduling. *Commun. ACM*, 9(6):433 – 434, 1966.
- [18] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1):55 – 89, 2009.
- [19] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87 – 127, 1999.
- [20] Jonathan M. Thompson and Kathryn A. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25(7):637 – 648, 1998.
- [21] Celso C. Ribeiro Dominique de Werra Tiago Januario, Sebastián Urrutia. Edge coloring: A natural model for sports scheduling. *European Journal of Operational Research*, 254:1 – 8, 2016.
- [22] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85 – 86, 1967.
- [23] George White and Pak-Wah Chan. Towards the construction of optimal examination schedules. *Information Systems and Operational Research*, 17(3):219 – 229, 1979.
- [24] D. C. Wood. A system for computing university examination timetables. *The Computer Journal*, 11(1):41 – 47, 1968.