\documentclass[letter, 10pt]{article}

\usepackage[latin1]{inputenc}

%%\usepackage[spanish]{babel}

\usepackage{amsfonts}

\usepackage{amsmath}

\usepackage[dvips]{graphicx}

\usepackage{url}

\usepackage[dvipsnames]{xcolor}

\usepackage[top=3cm,bottom=3cm,left=3.5cm,right=3.5cm,footskip=1.5cm,headheight=1.5cm,headsep=.5cm,textheight=3cm]{geometry}

\usepackage[spanish,es-tabla]{babel}

\usepackage{amsthm}

\usepackage{pgfplots}

\usepackage{pgfplotstable}

\usepackage{filecontents}

\usepackage[utf8]{inputenc}

\usepackage{tikz}

\usepackage{listings}

\usepackage{algorithm}

\usepackage[noend]{algpseudocode}

\makeatletter

\def\BState{\State\hskip-\ALG@thistlm}

\makeatother

\begin{filecontents}{testdata.csv}

category, value

Carter~\cite{carter},13

Merlot~\cite{merlot},2

Dowsland~\cite{dowsland},2

\end{filecontents}

\begin{filecontents}{testdata2.csv}

category2, value2

Carter~\cite{carter}, 1

Merlot~\cite{merlot}, 5

Di Gaspero~\cite{gaspero}, 0

Burke~\cite{annealing}, 6

\end{filecontents}

\begin{filecontents}{testdata3.csv}

category3, value3

Merlot~\cite{merlot},5

Burke~\cite{memetic},0

\end{filecontents}

\begin{filecontents}{testdata4.csv}

category4, value4

Merlot~\cite{merlot},1

Burke~\cite{annealing},1

\end{filecontents}

\begin{filecontents}{testdata5.csv}

category5, value5

Terashima-Marin~\cite{marin},12

Ross~\cite{ross},0

\end{filecontents}

\pgfplotstableread[col sep=comma]{testdata5.csv}\datatable

\pgfplotstableread[col sep=comma]{testdata4.csv}\datatable

\pgfplotstableread[col sep=comma]{testdata3.csv}\datatable

\pgfplotstableread[col sep=comma]{testdata2.csv}\datatable

\pgfplotstableread[col sep=comma]{testdata.csv}\datatable

\makeatletter

\pgfplotsset{

/pgfplots/flexible xticklabels from table/.code n args={3}{%

\pgfplotstableread[#3]{#1}\coordinate@table

\pgfplotstablegetcolumn{#2}\of{\coordinate@table}\to\pgfplots@xticklabels

\let\pgfplots@xticklabel=\pgfplots@user@ticklabel@list@x

}

}

\theoremstyle{definition}

\newtheorem{definition}{Definici\'on}[section]

\theoremstyle{remark}

\newtheorem\*{remark}{Remark}

\begin{document}

\title{Inteligencia Artificial \\ \begin{Large}**Estado del Arte:** \\Examination Timetabling Problem (ETP)\end{Large}}

\author{Rodrigo Cayazaya Mar\'in}

\date{\today}

\maketitle

%--------------------No borrar esta secci\'on--------------------------------%

\section\*{Evaluaci\'on}

\begin{tabular}{ll}

Resumen (5\%): & \underline{\hspace{2cm}} \\

Introducci\'on (5\%): & \underline{\hspace{2cm}} \\

Definici\'on del Problema (10\%): & \underline{\hspace{2cm}} \\

Estado del Arte (35\%): & \underline{\hspace{2cm}} \\

Modelo Matem\'atico (20\%): & \underline{\hspace{2cm}}\\

Conclusiones (20\%): & \underline{\hspace{2cm}}\\

Bibliograf\'ia (5\%): & \underline{\hspace{2cm}}\\

& \\

\textbf{Nota Final (100\%)}: & \underline{\hspace{2cm}}

\end{tabular}

%---------------------------------------------------------------------------%

\vspace{2cm}

\begin{abstract}

Las instituciones acad\'emicas deben organizar el calendario de evaluaciones de modo que no se asignen 2 o m\'as ex\'amenes a un estudiante en el mismo bloque horario, es por esto que surgen los problemas de calendarizaci\'on de ex\'amenes (ETP). Sin embargo, esta no es la \'unica variante que hay que tener en consideraci\'on, ya que, \¿Qu\'e ocurre con los aforos limitados por sala? Y \¿Descanso entre evaluaciones?, etc. Es por esto que existen distintos tipos de ETP para la resoluci\'on de estas variadas problem\'aticas.

\end{abstract}

\section{Introducci\'on}

Los problemas de calendarizaci\'on de ex\'amenes (ETP) buscan calendarizar los ex\'amenes de forma que ning\'un estudiante deba rendir m\'as de 2 ex\'amenes en un mismo periodo de tiempo, asignando la menor cantidad de bloques de horarios o mejor conocidos como timeslots (para efectos pr\'acticos se utilizar\'a la palabra ``timeslots") a los ex\'amenes. Este es un problema recurrente en las instituciones acad\'emicas, ya que realizan evaluaciones semestralmente o anualmente.\\

En este documento se investigar\'a el estado del arte del problema ETP, comparando resultados de diferentes m\'etodos existentes bajo un mismo dataset de la Universidad de Toronto. Tambi\'en se definir\'a y modelar\'a una variante del problema al agregar la minimizaci\'on de una penalizaci\'on ligada a la holgura de tiempo que existe entre ex\'amenes compartidos por estudiantes.

\section{Definici\'on del Problema}

\textit{Examination Timetabling Problem} es un problema de la categor\'ia \textit{Education Timetabling Problems}, que consiste en organizar las evaluaciones semestrales tipo ex\'amenes a trav\'es de timeslots, este tipo de problemas es recurrente en las instituciones acad\'emicas, debido a las restricciones que conlleva y a la gran cantidad de estudiantes con las que tratan estos establecimientos.

El objetivo principal es \textbf{organizar los ex\'amenes de manera que se minimicen los timeslots utilizados}, esto significa que se distribuir\'an los ex\'amenes utilizando la menor cantidad de timeslots posibles. En la tabla \ref{table:timeslot1} se puede observar c\'omo se organizarian 5 ex\'amenes utilizando 5 timeslots.

\begin{table}[hbt!]

\centering

\begin{tabular}{| c | c |}

\hline

Timeslot 1 & Ex\'amen 1 \\ \hline

Timeslot 2 & Ex\'amen 2 \\ \hline

Timeslot 3 & Ex\'amen 3 \\ \hline

Timeslot 4 & Ex\'amen 4 \\ \hline

Timeslot 5 & Ex\'amen 5 \\ \hline

\end{tabular}

\caption{Posible soluci\'on con 5 ex\'amenes y 5 timeslots.}

\label{table:timeslot1}

\end{table}

Al observar la tabla \ref{table:timeslot1} se puede preguntar, ?`por qu\'e no colocar los 5 ex\'amenes en el primer timeslot y as\'i utilizar solamente 1 timeslot? (tal como se muestra en la tabla \ref{table:timeslot2}). Esto no es factible, ya que puede causar problemas si al menos un estudiante debe rendir m\'as de un ex\'amen, ya que no podr\'ia rendirlos simult\'aneamente.

\begin{table}[hbt!]

\centering

\begin{tabular}{| c | c | c | c | c | c |}

\hline

Timeslot 1 & Ex\'amen 1 & Ex\'amen 2 & Ex\'amen 3 & Ex\'amen 4 & Ex\'amen 5 \\ \hline

\end{tabular}

\caption{Posible soluci\'on con 5 ex\'amenes y 1 timeslot.}

\label{table:timeslot2}

\end{table}

De esta premisa nace la restricci\'on principal del problema: \textbf{A ning\'un estudiante se le asignar\'an 2 o m\'as ex\'amenes en un mismo timeslot}, esta restricci\'on es conocida coloquialmente como ``tope de horario".\\

Para el siguiente objetivo se debe definir los siguientes conceptos:

\begin{definition}[Penalizaci\'on]

La penalizaci\'on es el valor asociado a la cantidad de timeslots que existe entre un ex\'amen y otro. Este valor asociado se puede observar en la tabla \ref{table:timeslot3}. Asimismo, la penalizaci\'on de un estudiante es la suma de las penalizaciones de todos sus ex\'amenes.

\end{definition}

\begin{definition}[Penalizaci\'on promedio por estudiante]

La penalizaci\'on promedio por estudiante es la suma de las penalizaciones de todos estudiantes, dividido en la cantidad estudiantes.

\end{definition}

\begin{table}[hbt!]

\centering

\begin{tabular}{| c | c |}

\hline

\textbf{Cantidad de timeslots} & \textbf{Penalizaci\'on} \\ \hline

0 & 16 \\ \hline

1 & 8 \\ \hline

2 & 4 \\ \hline

3 & 2 \\ \hline

4 & 1 \\ \hline

5 o m\'as & 0 \\ \hline

\end{tabular}

\caption{Penalizaciones dependiendo de la cantidad de timeslots entre ex\'amenes.}

\label{table:timeslot3}

\end{table}

Con estas definiciones ahora se puede detallar el objetivo secundario de este problema, el cual consiste en \textbf{minimizar la penalizaci\'on promedio por estudiante}. Este objetivo no es estrictamente necesario llevarlo a cabo para resolver el problema, pero s\'i es un objetivo deseable.\\

El problema que se acaba de definir consiste en uno de la categor\'ia \textit{Education Timetabling Problems}, sin embargo no es el \'unico, ya que existen otros 2 problemas m\'as.\\

\textbf{\textit{High Schools Timetabling Problem (HSTP)}} y \textbf{\textit{University Course Timetabling Problem (UCTP)}} son problemas similares al problema anteriormente planteado, ya que los 3 problemas deben asignar eventos en timeslots, sin embargo tienen peque\~{n}as variaciones.

El objetivo de HSTP es asignar los horarios de clases anuales a los cursos y a los profesores de manera que no exista un ``tope de horario", adem\'as existe un objetivo secundario que consta de minimizar los timeslots donde (despu\'es de dictar una clase) los profesores part-time se encuentren libres, pero tengan que esperar para dictar una clase (coloquialmente llamado ``ventana"). Las restricciones tambi\'en cambian, ya que se agrega la restricci\'on donde no se debe exceder la capacidad m\'axima de la sala~\cite{EducationalTimetabling}.

El objetivo de UCTP es similar al de HSTP, pero no se asignan horarios de manera anual, sino que de manera semestral, y se aplica individualmente a alumnos y no a un curso de alumnos~\cite{EducationalTimetabling}.\\

Otro problema similar, pero alejado de la categor\'ia \textit{Education Timetabling Problems} es el \textbf{\textit{Nurse-Scheduling Problem (NCP)}}, el cual trata sobre crear calendarios semanales para los turnos que deben tomar las enfermeras. El objetivo principal es simplemente asignar un horario a las enfermeras sin romper las restricciones y su objetivo secundario es minimizar el coste asignado a las preferencias de todas las enfermeras. Sus restricciones son varias: se debe satisfacer la cantidad de turnos por contrato, tener una cierta cantidad de enfermeras de un grado mayor por cada turno, una enfermera de un grado mayor puede reemplazar a una de un grado menor y una enfermera no puede trabajar de d\'ia y de noche la misma semana~\cite{nurseScheduling}.

Como se puede observar, existe una similitud entre el problema ETP anteriormente definido y el problema NSP, ya que ambos tratan con el problema de asignar eventos en horarios bajo ciertas restricciones.\\

No solamente existen problemas de calendarizaci\'on en los trabajos e instituciones, tambi\'en suceden en las actividades, tal como es el caso del \textbf{\textit{Sports Timetabling Problem (STP)}}. Su objetivo principal consta de organizar partidos de un torneo en timeslots o ``jornadas deportivas" que representan un periodo de tiempo donde todos los equipos juegan (por ejemplo, un fin de semana). Las restricciones vienen dadas en que durante este timeslot todos los equipos deben jugar solamente 1 vez, en cada timeslot los equipos deben jugar con un equipo distinto hasta que todos los equipos hayan jugado entre s\'i, adem\'as los equipos deben jugar una cantidad equitativa de veces en su estadio (conocido coloquialmente como ``jugar de local")~\cite{sportTimetabling}.

Es clara la similitud entre problemas, ya que, a pesar de que no es necesario minimizar una cantidad de un evento, s\'i es necesario organizar una calendarizaci\'on de estos, tal como se ha visto en los problemas anteriores.\\

Por \'ultimo se explicar\'a el problema \textbf{\textit{Train Timetabling Problem (TTP)}}, de la categor\'ia \textit{Transportation Timetabling Problem}. El objetivo principal de este problema es planificar el calendario para cada tren, minimizando la distancia entre trenes de un mismo recorrido, cabe destacar que los trenes llevan una prioridad asignada. La restricci\'on viene dada por las reparaciones que puedan existir en las v\'ias, por lo que el tren deber\'a desviarse (equivalente a una penalizaci\'on). As\'i surgiendo un objetivo secundario: maximizar la diferencia entre el valor de la prioridad del tren y la penalizaci\'on~\cite{trainTimetabling}.

El objetivo del problema TTP es bastante similar al que se revis\'o de ETP, ya que ambos comparten la creaci\'on de un calendario intentando minimizar un evento.\\

Tanto ETP, TTP, STP, NSP, UCTP y HSTP son problemas que se encuentran relacionados, ya que todos intentan crear un calendario en base a restricciones, intentando de alguna u otra manera minimizar o maximizar alg\'un objetivo.

\section{Estado del Arte}

\subsection{Origen}

Los problemas de calendarizaci\'on (\textit{Timetabling Problems}) no son algo nuevo, el primer registro de este tipo de problemas aplicado a una computadora dicta del \textbf{1961} por Appleby sobre la categor\'ia \textit{Education Timetabling Problems}\cite{1961}. Tres a\~{n}os m\'as tarde en \textbf{1964} Sol Broder implementa un programa para \textit{Examination Timetabling Problem}\cite{firstExamination}. Posteriormente, aparecieron las dem\'as categor\'ias como \textit{Transportation Timetabling Problem}, que fue introducido por Pullen en \textbf{1967}~\cite{1967}.

\subsection{Diferentes t\'ecnicas}

Han existido diferentes t\'ecnicas para resolver el problema de calendarizaci\'on de ex\'amenes, el primero fue utilizado en 1964 llamada la t\'ecnica de \textbf{``grafo basado en t\'ecnicas secuenciales"}~\cite{survey}. Esta t\'ecnica consiste en utilizar el algoritmo de coloreo de grafo: con el objetivo de minimizar la cantidad de colores, cada nodo del grafo lleva un color y no pueden existir nodos conectados que contengan un mismo color. Para resolver un ETP, solamente se deben reemplazar los nodos por ex\'amenes, las aristas (conexiones) por las restricciones duras (aquellas restricciones que se deben cumplir para satisfacer el problema) y los colores asignados a los nodos representan el timeslot asociado al ex\'amen. Cabe destacar que esta t\'ecnica no sirve para un problema ETP con restricciones blandas (aquellas restricciones que son deseables, pero no obligatorias)~\cite{grafo}.\\

Otra t\'ecnica utilizada es la \textbf{t\'ecnica basada en restricciones}. Consiste en modelar los ex\'amenes como variables con un dominio finito, donde los valores del dominio representan los timeslots y salas donde se dictar\'an los ex\'amenes, las variables son asignadas de forma secuencial para construir las soluciones del problema. Si alg\'un valor del dominio no puede ser instanciado, se emplea backtracking para encontrar una soluci\'on posible~\cite{survey}.\\

Tambi\'en existen categor\'ias de t\'ecnicas, tal como es el caso de las \textbf{t\'ecnicas basadas en b\'usqueda local}. Estas buscan soluciones dentro de un vecindario, la estructura de este vecindario depende de la t\'ecnica que se utilice. \textbf{Tabu Search} es un de estas t\'ecnicas, la cual consiste en explorar el espacio de b\'usqueda, pero sin re-visitar los recientes movimientos que lleva en un lista (lista tab\'u). El algoritmo se detiene si encuentra una soluci\'on que cumpla con los criterios de detenci\'on, si esto no ocurre, continua buscando en otro vecindario, aunque este vecindario sea peor que el anterior~\cite{tabu}.

Tabu Search no es la \'unica t\'ecnica basada en b\'usqueda local, tambi\'en existe la t\'ecnica \textbf{Simulated Annealing}, que utiliza un \'area mayor del espacio de b\'usqueda al principio del proceso, para as\'i aceptar peores movimiento con mayor probabilidad, la cual gradualmente va decreciendo a medida que la b\'usqueda continua.

Hoy en d\'ia no se utilizan estas t\'ecnicas con un vecindario de forma aleatoria, sino que existen diferentes dise\~{n}os de vecindarios, ayudando a escapar de que el \'optimo sea solamente local, una de estas t\'ecnicas es usar una estructura de m\'ultiples vecindarios, lo cual otorga m\'as flexibilidad de navegaci\'on del espacio de b\'usqueda~\cite{simulated}.\\

Las t\'ecnicas basadas en algoritmos tambi\'en son una forma de afrontar este tipo de problemas, uno de estos son los \textbf{algoritmos basado en la poblaci\'on}. Uno de estos es llamado \textbf{Genetic Algorithms}, los cuales simulan el proceso evolutivo de la naturaleza manipulando y evolucionando poblaciones de soluciones dentro del espacio de b\'usqueda, apuntando a obtener mejores resultados al pasar las generaciones. Estos algoritmos trabajan con m\'as de 1 soluci\'on a la vez, a diferencia de la categor\'ia anterior que solo encontraba 1 soluci\'on y la mejoraba con el tiempo~\cite{genetic}.

Una variaci\'on de los Genetic Algorithms son los \textbf{Memetic Algorithms}, estos se diferencian en que los individuos de una poblaci\'on mejoran durante su vida dentro de una sola generaci\'on, sin embargo esto requiere de un gran tiempo de c\'omputo~\cite{memetic2}.

El \'ultimo algoritmo basado en la poblaci\'on es el \textbf{Ant Algorithm} que consiste en replicar la forma en que las hormigas encuentran la ruta m\'as corta hacia la comida a trav\'es de la feromona que dejan en el camino. En este algoritmo, cada hormiga es usada para construir una soluci\'on y la informaci\'on se mantiene como la feromona, la cual sirve para ayudar a generar soluciones para la siguiente etapa~\cite{ant}.\\

Una t\'ecnica un poco alejada de lo normal, es la \textbf{t\'ecnica de m\'ultiples criterios}, la cual consiste en utilizar distintas restricciones como criterios, en vez de utilizar la suma de los pesos de una restricci\'on. Esta t\'ecnica se aleja de las dem\'as porque apunta en distintas direcciones, es por ello que no se puede comparar con el resto de las t\'ecnicas~\cite{multi}.\\

La \'ultima t\'ecnica llamada \textbf{hyper-heuristics} consiste en desarrollar un enfoque m\'as general y no uno tan dependiente y espec\'ifico de los par\'ametros, as\'i se logra desarrollar soluciones que no son para problemas espec\'ificos~\cite{survey}.\\

\subsection{Comparaci\'on entre t\'ecnicas}

A continuaci\'on se realizar\'a una comparaci\'on entre las distintas t\'ecnicas antes descritas, cabe destacar que la implementaci\'on de las t\'ecnicas no representan un resultado representativo (pero s\'i aproximado), ya que una t\'ecnica se puede representar por distintos algoritmos, obteniendo resultados distintos. Tambi\'en cabe destacar que la \textit{t\'ecnica de m\'ultiples criterios} no ser\'a evaluada, ya que su enfoque es distinto al de las dem\'as t\'ecnicas.\\

Los datos utilizados son los famosos datasets de Toronto, estos se pueden visualizar con m\'as detalles en la p\'agina: $http://www.cs.nott.ac.uk/pszrq/data.htm$.\\

\textit{Toronto a} tiene como objetivo minimizar la cantidad de timeslots, \textit{Toronto b} tiene como objetivo minimizar el costo promedio por estudiante, \textit{Toronto c} tiene como objetivo minimizar la cantidad de estudiantes que tengan 2 ex\'amenes el mismo d\'ia, \textit{Toronto d} tiene como objetivo minimizar la cantidad de estudiantes que tengan 2 ex\'amenes el mismo d\'ia y 2 ex\'amenes de noche, por \'ultimo \textit{Toronto e} tiene como objetivo minimizar la cantidad de estudiantes con 2 ex\'amenes en timeslots consecutivos.

\begin{table}[hbt!]

\centering

\begin{tabular}{| c | c | c |}

\hline

\textbf{T\'ecnica} & \textbf{Implementaci\'on} & \textbf{Dataset}\\ \hline

Grafo basado en t\'ecnicas secuenciales & Carter~\cite{carter} & \textit{Toronto a y b}\\ \hline

T\'ecnica basada en restricciones & Merlot~\cite{merlot} & \textit{Toronto a,b,c y d} \\ \hline

Tabu Search & Di Gaspero~\cite{gaspero} & \textit{Toronto b} \\ \hline

Simulated Annealing & Burke~\cite{annealing} & \textit{Toronto b y d} \\ \hline

Genetic Algorithm & Terashima-Marin~\cite{marin} & \textit{Toronto e} \\ \hline

Memetic Algorithm & Burke~\cite{memetic} & \textit{Toronto c} \\ \hline

Ant Algorithm & Dowsland~\cite{dowsland} & \textit{Toronto a}\\ \hline

Hyper-heuristics & Ross~\cite{ross} & \textit{Toronto e} \\ \hline

\end{tabular}

\caption{Diferentes implementaciones.}

\label{table:tabla4}

\end{table}

En la tabla \ref{table:tabla4} se pueden observar las implementaciones que se usar\'an para representar a las diferentes t\'ecnicas y los diferentes datasets con que se probaron.\\

En las tablas \ref{table:tabla5}, \ref{table:tabla6} y \ref{table:tabla7} se encuentran los resultados de las implementaciones en los diferentes datasets.\\

\begin{table}[hbt!]

\centering

\begin{tabular}{| c || c | c | c |}

\hline

\textbf{Dataset} & \textbf{Carter~\cite{carter}} & \textbf{Merlot~\cite{merlot}} & \textbf{Dowsland~\cite{dowsland}}\\ \hline

car91 & \textbf{28} & 30 & 35 \\ \hline

car92 & \textbf{28} & 31 & 35 \\ \hline

ear83 & \textbf{22} & 24 & 24 \\ \hline

hec92 & \textbf{17} & 18 & 18 \\ \hline

kfu93 & \textbf{19} & 21 & 20 \\ \hline

lse91 & \textbf{17} & 18 & 18 \\ \hline

rye92 & \textbf{21} & 22 & 23 \\ \hline

pur93 & \textbf{35} & - & - \\ \hline

sta83 & \textbf{13} & \textbf{13} & \textbf{13} \\ \hline

tre92 & \textbf{20} & 21 & 23 \\ \hline

uta92 & \textbf{32} & \textbf{32} & 35 \\ \hline

ute92 & \textbf{10} & 11 & \textbf{10} \\ \hline

yor83 & \textbf{19} & 23 & 21 \\ \hline

\end{tabular}

\caption{Resultados con dataset \textit{Toronto a}.}

\label{table:tabla5}

\end{table}

\begin{table}[hbt!]

\centering

\begin{tabular}{| c || c | c | c | c |}

\hline

\textbf{Dataset} & \textbf{Carter~\cite{carter}} & \textbf{Merlot~\cite{merlot}} & \textbf{Di Gaspero~\cite{gaspero}} & \textbf{Burke~\cite{annealing}}\\ \hline

car91 & 7.1 & 5.1 & 5.7 & \textbf{4.8}\\ \hline

car92 & 6.2 & 4.3 & - & \textbf{4.2}\\ \hline

ear83 & 36.4 & \textbf{35.1} & 39.4 & 35.4\\ \hline

hec92 & 10.8 & \textbf{10.6} & 10.9 & 10.8\\ \hline

kfu93 & 14.0 & \textbf{13.5} & - & 13.7\\ \hline

lse91 & 10.5 & 10.5 & 12.6 & \textbf{10.4}\\ \hline

rye92 & \textbf{7.3} & - & - & 8.9\\ \hline

pur93 & - & - & - & -\\ \hline

sta83 & 161.5 & \textbf{157.3} & 157.4 & 159.1\\ \hline

tre92 & 9.6 & 8.4 & - & \textbf{8.3}\\ \hline

uta92 & 3.5 & 3.5 & 4.1 & \textbf{3.4}\\ \hline

ute92 & 25.8 & \textbf{25.1} & - & 25.7\\ \hline

yor83 & 41.7 & 37.4 & 39.7 & \textbf{36.7}\\ \hline

\end{tabular}

\caption{Resultados con dataset \textit{Toronto b}.}

\label{table:tabla6}

\end{table}

\begin{table}[hbt!]

\centering

\begin{tabular}{| c || c | c || c | c || c | c |}

\hline

\textbf{Dataset} & \textbf{Merlot~\cite{merlot}} & \textbf{Burke~\cite{memetic}} & \textbf{Merlot~\cite{merlot}} & \textbf{Burke~\cite{annealing}} & \textbf{Terashima-Marin~\cite{marin}} & \textbf{Ross~\cite{ross}}\\ \hline

car91 & \textbf{158} & 331 & - & - & \textbf{130} & 283 \\ \hline

car92 & \textbf{31} & 81 & 1744 & \textbf{1506} & \textbf{285} & 542\\ \hline

ear83 & - & - & - & - & \textbf{723} & 958\\ \hline

hec92 & - & - & - & - & \textbf{154} & 224 \\ \hline

kfu93 & \textbf{247} & 974 & \textbf{1082} & 1321 & \textbf{223} & 226\\ \hline

lse91 & - & - & - & - & \textbf{221} & 263\\ \hline

rye92 & - & - & - & - & \textbf{671} & 832\\ \hline

pur93 & - & - & - & - & - & - \\ \hline

sta83 & - & - & - & - & \textbf{821} & 1058\\ \hline

tre92 & \textbf{0} & 3 & - & - & \textbf{586} & 604 \\ \hline

uta92 & \textbf{334} & 772 & - & - & \textbf{594} & 855\\ \hline

ute92 & - & - & - & - & \textbf{902} & 967\\ \hline

yor83 & - & - & - & - & \textbf{708} & 758\\ \hline

\end{tabular}

\caption{Resultados con dataset \textit{Toronto c, d y e}, de izquierda a derecha.}

\label{table:tabla7}

\end{table}

\subsection{ETP Hoy}

Hoy en d\'ia existe tecnolog\'ia que en 1964 no exist\'ia, es por ello que tanto los problemas como las soluciones han evolucionado. El usuario ya no necesita escribir a mano los inputs, sino que ahora todo es digital. Es por ello que una tendencia actual es utilizar \textit{Mixed Integer Programming Models} y el framework \textit{Django} para as\'i desarrollar un \textit{Web Based Decision support system}. As\'i el usuario solo debe ingresar al sitio web, subir los inputs en un archivo, estos se ingresa a una base de datos para correr los modelos a trav\'es de servidores y en menos de 2 minutos desplegar el resultado para el usuario~\cite{2021}.

\section{Modelo Matem\'atico}

A continuaci\'on se modelar\'a el problema de ETP explicado en la secci\'on \textit{Definici\'on del Problema}.\\

\textbf{Variables:}

\begin{equation}

X\_{e,t} =

\begin{cases}

1,& \text{Si el ex\'amen \textit{e} se dicta en el timeslot \textit{t}.}\\

0,& \text{Caso contrario.}

\end{cases}

\end{equation}

$X\_{e,p}$ es la variable que se utilizar\'a en la funci\'on objetivo principal, el objetivo es asignar la mayor cantidad de ex\'amenes \textit{e} a un timeslot \textit{t}.

\begin{equation}

penalty(i) =

\begin{cases}

16,& \text{Si penalty(i)\*=0}\\

8,& \text{Si penalty(i)\*=1}\\

4,& \text{Si penalty(i)\*=2}\\

2,& \text{Si penalty(i)\*=3}\\

1,& \text{Si penalty(i)\*=4}\\

0,& \text{Caso contrario.}\\

\end{cases}

\end{equation}

\begin{equation}

\textit{penalty(i)\*} = C\_{e,f,i} \*( |p\_1 \* X\_{e,p\_1} - p\_2 \* X\_{f,p\_2}| - 1) \, \, \, \forall p\_1 \neq p\_2 \in P \land \forall e \neq f \in E

\end{equation}

penalty(i)\* representa la cantidad de timeslots que existen entre los diferentes timeslots asignados a los ex\'amenes que debe rendir el estudiante \textit{i}.\\

\textbf{Constantes:}

\begin{itemize}

\item A = Conjunto de alumnos.

\item E = Conjunto de ex\'amenes.

\item T = Conjunto de timeslots.

\item P = Penalizaci\'on promedio por estudiante.

\end{itemize}

\begin{equation}

C\_{e,f,i} =

\begin{cases}

1,& \text{Si el ex\'amen \textit{e} comparte al estudiante \textit{i} con el ex\'amen \textit{f}.}\\

0,& \text{Caso contrario.}

\end{cases} \, C\_{e,f,i} \in R^{ExE}

\end{equation}

\begin{equation}

S\_{i,e} =

\begin{cases}

1,& \text{Si el estudiante \textit{i} debe rendir el ex\'amen \textit{e}.}\\

0,& \text{Caso contrario.}

\end{cases}

\end{equation}\\

\textbf{Dominios:}

\begin{itemize}

\item $e \in [1,\#E]$

\item $i \in [1,\#A]$

\item $t \in [1,\#T]$

\end{itemize}\\

\smallskip

\textbf{Funci\'on objetivo:}

\begin{itemize}

\item Principal\\

Min T: m\'ax $\vec{X}$

El objetivo principal es minimizar la cantidad de timeslots (T) a la vez que se maximizan la cantidad de ex\'amenes asignados a timeslots.

\item Secundaria\\

Min P: $\frac{\sum\_{i \in A}\text{penalty(i)}}{\#A}$

El objetivo secundario es minimizar la penalizaci\'on promedio por estudiante (P).

\end{itemize}

\smallskip

\textbf{Restricciones:}

\begin{enumerate}

\item Ning\'un estudiante rinde 2 o m\'as ex\'amenes en el mismo timeslot.

\smallskip\\

$\sum\_{e \in E} S\_{i,e}\*X\_{e,t} \leq 1$ \, $\forall i \in A \land \forall t \in T$

\item Cada ex\'amen est\'a asignado a al menos 1 periodo.

\smallskip\\

$\sum\_{t \in T} X\_{e,t} \geq 1$ \, $\forall e \in E$

\end{enumerate}

\section{Representaci\'on}

\textcolor{red}{Representaci\'on de \textbf{soluciones} (arreglos, matrices, etc.). En caso de t\'ecnicas completas indicar variables y dominios. Incluir justificaci\'on y ejemplos para mayor claridad.}

El problema se represent\'o utilizando las siguientes variables, constantes, dominios y restricciones:\\

\textbf{Variables:}

\begin{equation}

X\_{e,t} =

\begin{cases}

1,& \text{Si el ex\'amen \textit{e} se dicta en el timeslot \textit{t}.}\\

0,& \text{Caso contrario.}

\end{cases}

\end{equation}

$X\_{e,p}$ es la variable que se utilizar\'a en la funci\'on objetivo principal, el objetivo es asignar la mayor cantidad de ex\'amenes \textit{e} a un timeslot \textit{t}.

\begin{equation}

penalty(i) =

\begin{cases}

16,& \text{Si penalty(i)\*=0}\\

8,& \text{Si penalty(i)\*=1}\\

4,& \text{Si penalty(i)\*=2}\\

2,& \text{Si penalty(i)\*=3}\\

1,& \text{Si penalty(i)\*=4}\\

0,& \text{Caso contrario.}\\

\end{cases}

\end{equation}

\begin{equation}

\textit{penalty(i)\*} = C\_{e,f,i} \*( |p\_1 \* X\_{e,p\_1} - p\_2 \* X\_{f,p\_2}| - 1) \, \, \, \forall p\_1 \neq p\_2 \in P \land \forall e \neq f \in E

\end{equation}

penalty(i)\* representa la cantidad de timeslots que existen entre los diferentes timeslots asignados a los ex\'amenes que debe rendir el estudiante \textit{i}.\\

\begin{equation}

Conf\_{e,f,i,t} =

\begin{cases}

1,& \text{Si C\_{e,f,i} = 1 \land el ex\'amen \textit{e} comparte al timeslot \textit{t} con el con el ex\'amen \textit{f}.}\\

0,& \text{Caso contrario.}

\end{cases} \, Conf\_{e,f,i,t} \in R^{ExE}

\end{equation}

\begin{equation}

Res\_{T,X,E} =

\begin{cases}

Min T: m\'ax $\vec{X}$ ,& \text{Si existe una solución para el conjunto de exámenes E.}\\

vacío,& \text{Caso contrario.}

\end{cases} \, Res\_{E} \in R^{T^ExT\*E}

\end{equation}

\textbf{Constantes:}

\begin{itemize}

\item A = Conjunto de alumnos.

\item E = Conjunto de ex\'amenes.

\item T = Conjunto de timeslots.

\item P = Penalizaci\'on promedio por estudiante.

\end{itemize}

\begin{equation}

C\_{e,f,i} =

\begin{cases}

1,& \text{Si el ex\'amen \textit{e} comparte al estudiante \textit{i} con el ex\'amen \textit{f}.}\\

0,& \text{Caso contrario.}

\end{cases} \, C\_{e,f,i} \in R^{ExE}

\end{equation}

\begin{equation}

S\_{i,e} =

\begin{cases}

1,& \text{Si el estudiante \textit{i} debe rendir el ex\'amen \textit{e}.}\\

0,& \text{Caso contrario.}

\end{cases}

\end{equation}\\

\textbf{Dominios:}

\begin{itemize}

\item $e \in [1,\#E]$

\item $i \in [1,\#A]$

\item $t \in [1,\#T]$

\end{itemize}\\

\smallskip

\textbf{Restricciones:}

\begin{enumerate}

\item Ning\'un estudiante rinde 2 o m\'as ex\'amenes en el mismo timeslot.

\smallskip\\

$\sum\_{e \in E} S\_{i,e}\*X\_{e,t} \leq 1$ \, $\forall i \in A \land \forall t \in T$

\item Cada ex\'amen est\'a asignado a al menos 1 periodo.

\smallskip\\

$\sum\_{t \in T} X\_{e,t} \geq 1$ \, $\forall e \in E$

\end{enumerate}

\smallskip

Esta representaci\'on es similar a la presentada en el Modelo Matem\'atico, con la diferencia en que se agregaron dos matrices:

\begin{itemize}

\item Matriz \textbf{Res} de soluciones factibles: De estas soluciones se escoger\'a la que tenga la menor penalizaci\'on promedio por estudiante como soluci\'on final, su tamaño viene definido por un límite superior como la cantidad máxima de soluciones posibles.

\item Matriz \textbf{Conj} de conflictos: Este ser\'a utilizado por el algoritmo \textit{Conflict-based backjumping} explicado m\'as adelante, su tamaño representa una relación 1:1 entre los exámenes.

\end{itemize}\\

Las soluciones se representaron en forma de \'arbol, donde cada nodo representa una instanciaci\'on de un examen \textit{e} a un timeslot \textit{t}, parecido al \'arbol de la Figura \ref{tree:arbol}.\\ Sin embargo, esta representaci\'on representa tanto las soluciones factibles como las infactibles, por eso que decidi\'o representar las soluciones como un \'arbol \textit{unario}, parecido a un grafo, donde se realiza una b\'usqueda en profundidad de las soluciones factibles hasta encontrar una soluci\'on, la cual es almacenada en el conjunto R. En la Figura \ref{tree:arboles\_unarios} se observa como ser\'ia una instanciaci\'on de 3 ex\'amenes que se encuentran en conflicto.\\

\begin{figure}[hbt!]

\hfil

\begin{tikzpicture}

\node {inicio} [sibling distance = 3cm]

child {node {e1=t1} [sibling distance = 1.5cm]

child {node {e2=t1}}

child {node {e2=t2}}}

child {node {e1=t2} [sibling distance = 1.5cm]

child {node {e2=t1}}

child {node {e2=t2}}};

\end{tikzpicture}

\caption{Posible representaci\'on en forma de \'arbol.}

\label{tree:arbol}

\end{figure}

\begin{figure}[hbt!]

\begin{minipage}[b]{0.10\linewidth}

\begin{tikzpicture} [

tlabel/.style={pos=0.4,right=-1pt},

baseline=(current bounding box.center)

]

\node {inicio}

child {node {e1=t1}

child [red]{node {e2=t1} edge from parent [dashed]}};

\end{tikzpicture}

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

$\rightarrow$

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

\begin{tikzpicture} [

tlabel/.style={pos=0.4,right=-1pt},

baseline=(current bounding box.center)

]

\node {inicio}

child {node {e1=t1}

child {node {e2=t2}}};

\end{tikzpicture}

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

$\rightarrow$

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

\begin{tikzpicture} [

tlabel/.style={pos=0.4,right=-1pt},

baseline=(current bounding box.center)

]

\node {inicio}

child {node {e1=t1}

child {node {e2=t2}

child [red]{node {e3=t1} edge from parent [dashed]}}};

\end{tikzpicture}

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

$\rightarrow$

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

\begin{tikzpicture} [

tlabel/.style={pos=0.4,right=-1pt},

baseline=(current bounding box.center)

]

\node {inicio}

child {node {e1=t1}

child {node {e2=t2}

child [red]{node {e3=t2} edge from parent [dashed]}}};

\end{tikzpicture}

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

$\rightarrow$

\end{minipage}

\begin{minipage}[b]{0.10\linewidth}

\begin{tikzpicture} [

tlabel/.style={pos=0.4,right=-1pt},

baseline=(current bounding box.center)

]

\node {inicio}

child {node {e1=t1}

child {node {e2=t2}

child {node {e3=t3}}}};

\end{tikzpicture}

\end{minipage}

\caption{Instanciaci\'on de 3 ex\'amenes en conflicto.}

\label{tree:arboles\_unarios}

\end{figure}

Esta representaci\'on resulta ser eficiente, ya que reduce el espacio de b\'usqueda al no buscar entre soluciones infactibles, por lo que ahorra tiempo y almacenamiento. Una vez encontrada una soluci\'on, verifica en la siguiente instanciaci\'on factible para as\'i no perder soluciones. Esto se realiza para todos los ex\'amenes, as\'i cada soluci\'on representa una soluci\'on factible del problema original.\\

Concretamente hablando, las representaci\'on del grafo en el lenguaje C se realizaron con una estructura que contiene el id del ex\'amen, el timeslot asociado, un puntero al nodo hijo y uno para el padre. Para el conjunto R se cre\'o una matriz de una estructura que contiene un ex\'amen y su timeslot asociado, as\'i una entrada de la matriz corresponde a un arreglo que contiene una soluci\'on (osea, un arreglo de dicha estructura).\\

\section{Descripci\'on del algoritmo}

\textcolor{red}{C\'omo fue implementada la soluci\'on. Interesa la implementaci\'on particular m\'as que el algoritmo gen\'erico, es decir, si se tiene que implementar SA, lo que se espera es que se explique en pseudoc\'odigo la estructura

general y en p\'arrafos explicativos c\'omo fue implementada cada parte para su problema particular. Si

se utilizan operadores/movimientos se debe justificar por qu\'e se utilizaron dichos operadores/movimientos.

En caso de una t\'ecnica completa, describir detalles relevantes del proceso, si se utiliza recursi\'on o no, explicar c\'omo se van construyendo soluciones, cu\'ando se revisan restricciones, c\'omo se registran conflictos, etc. En este punto no se espera que se incluya c\'odigo, eso va aparte.}

Los algoritmos utilizados fueron \textit{Backtracking} y \textit{Conflict-based backjumping (CBJ)}, ambos son algoritmos tipo \textit{Look-Back}. Primero se ingres\'o una instanciaci\'on al \'arbol para luego comprobar si es factible, en un, si es factible se queda, si no lo es, se elimina y se realiza un salto atr\'as. Este salto es llevado a cabo por \textit{Backtracking} o \textit{CBJ}.

El primer acercamiento que se realiz\'o fue solamente realizar \textit{Backtracking}, es decir, realizar un salto de tama\~no 1 hacia atr\'as en todas las ocasiones necesarias, tal como se puede apreciar en el Algorithm \ref{code:backtracking}. Una vez realizado esto, se agreg\'o el salto \textit{CBJ} cuando ning\'un timeslot pod\'ia ser instanciado de forma factible en un ex\'amen (mejor conocido como un \textit{fallo}). Una vez realizado \textit{CBJ}, este se desactiva por toda la rama (hasta la ra\'iz) y se realiza solamente \textit{Backtracking}, esto se realiz\'o para poder encontrar todas las soluciones. Esta implementaci\'on se puede apreciar en la parte azul del Algorithm \ref{code:cbj&backtracking}.

\begin{algorithm}[hbt!]

\caption{Implementaci\'on con Backtracking}\label{code:backtracking}

\begin{algorithmic}[1]

\Procedure{ETP}{}

\State $\textit{R} \gets \text{Conjunto de soluciones}$

\While{existan soluciones factibles}

\State $\textit{Arbol} \gets \text{\'Arbol vacio}$

\While{existan ex\'amenes no instanciados en \textit{Arbol}}

\State $\textit{IdExamen} \gets \text{Id de un examen no instanciado}$

\While{\textit{IdExamen} no se encuentre en \textit{Arbol}}

\State $\textit{Timeslot} \gets \text{Timeslot no instanciado con \textit{IdExamen}}$

\State $\textit{Nodo} \gets \text{\textit{instanciacion}(\textit{IdExamen},\textit{Timeslot})}$

\State $\textit{agregarNodo}\text{(\textit{Arbol},\textit{Nodo})}$

\If {existe tope de horario en \textit{Arbol}}

\State $\textit{backtracking}\text{(\textit{Nodo})}$

\EndIf

\EndWhile

\EndWhile

\State $\textit{agregarSolucion}\text{(\textit{R},\textit{Arbol})}$

\EndWhile

\EndProcedure

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[hbt!]

\caption{Implementaci\'on con Backtracking y CBJ}\label{code:cbj&backtracking}

\begin{algorithmic}[1]

\Procedure{ETP}{}

\State $\textit{R} \gets \text{Conjunto de soluciones}$

\While{existan soluciones factibles}

\State $\textit{Arbol} \gets \text{\'Arbol vacio}$

\While{existan ex\'amenes no instanciados en \textit{Arbol}}

\If {\textcolor{blue}{es la primera iteraci\'on de la rama}}

\State $\textcolor{blue}{\textit{Conf} \gets \text{Conjunto de conflicto vac\'io}}$

\EndIf

\State $\textit{IdExamen} \gets \text{Id de un examen no instanciado}$

\While{\textit{IdExamen} no se encuentre en \textit{Arbol}}

\State $\textit{Timeslot} \gets \text{Timeslot no instanciado con \textit{IdExamen}}$

\State $\textit{Nodo} \gets \text{\textit{instanciacion}(\textit{IdExamen},\textit{Timeslot})}$

\State $\textit{agregarNodo}\text{(\textit{Arbol},\textit{Nodo})}$

\If {existe tope de horario en \textit{Arbol}}

\If {\textcolor{blue}{no se ha aplicado CBJ en esta rama}}

\State $\textcolor{blue}{\textit{ExmConf} \gets \text{Ex\'amen m\'as prematuramente instanciado y en conflicto con \textit{Nodo}}}$

\State $\textcolor{blue}{\textit{agregarConflicto}\text{(\textit{Conf},\textit{ExmConf})}}$

\If {\textcolor{blue}{hay fallo}}

\State $\textcolor{blue}{\textit{CBJ}\text{(\textit{Arbol},\textit{Conf})}}$

\EndIf

\EndIf

\If {\textcolor{blue}{ya se aplic\'o \textit{CBJ} en esta rama \textbf{\'o} no hay fallo}}

\State $\textit{backtracking}\text{(\textit{Arbol})}$

\EndIf

\EndIf

\EndWhile

\EndWhile

\State $\textit{agregarSolucion}\text{(\textit{R},\textit{Arbol})}$

\EndWhile

\EndProcedure

\end{algorithmic}

\end{algorithm}

\section{Conclusiones}

De la vasta mayor\'ia de m\'etodos y algoritmos que se presentaron, todos intentan resolver el problema de ETP, pero desde distintas perspectivas. Si bien se puede concluir que algunos m\'etodos son m\'as efectivos que otros, los resultados pueden variar dependiendo del objetivo que se quiere satisfacer o cu\'antos objetivos se quieren satisfacer (como en el caso de la \textit{t\'ecnica de m\'ultiples criterios}). Sin embargo, los resultados obtenidos en las tablas \ref{table:tabla5}, \ref{table:tabla6} y \ref{table:tabla7} reflejan las t\'ecnicas m\'as prometedoras seg\'un sea el objetivo principal:

\begin{itemize}

\item Para minimizar la cantidad de timeslots: Grafo basado en t\'ecnicas secuenciales.

\item Para minimizar el costo promedio por estudiante: T\'ecnica basada en restricciones o Simulated Annealing.

\item Para minimizar la cantidad de estudiantes con 2 ex\'amenes el mismo d\'ia: T\'ecnica basada en restricciones.

\item Para minimizar la cantidad de estudiantes con 2 ex\'amenes el mismo d\'ia y 2 ex\'amenes de noche: T\'ecnica basada en restricciones o Simulated Annealing.

\item Para minimizar la cantidad de estudiantes con 2 ex\'amenes en timeslots consecutivos: Genetic Algorithm.

\end{itemize}

Este, sin emabrgo, no es el final para los problemas de ETP. Se siguen desarrollando diferentes implementaciones a las t\'ecnicas existentes, tal como es el caso de modificar el vecindario de b\'usqueda para las \textit{t\'ecnicas basadas en b\'usqueda local}.\\

Debido a la velocidad en que la tecnolog\'ia avanza, las soluciones tienen que modernizarse o adaptarse a los nuevos requerimientos. Se propone utilizar la paralelizaci\'on (como ``cloud computing'' o ``CUDA'') para as\'i, gracias a la disminuci\'on del trabajo de los servidores y el tiempo de c\'omputo, poder realizar una \textit{t\'ecnicas basadas en b\'usqueda local} con distintos vecindarios y quedarse con la que obtenga mejores resultados.

\section{Bibliograf\'ia}

\bibliographystyle{plain}

\bibliography{Referencias}

\end{document}