Asignación de Horarios Universitario: un enfoque determinístico

Genaro Alfonso Ramón Rodríguez[1], María del Pilar Pozos Parra[2]

División Académica de Informática y Sistemas,
Universidad Juárez Autónoma de Tabasco,
Carretera Cunduacán – Jalpa Km. 1, Cunduacán, Tabasco, México.
[1] genalfa@gmail.com
[2] pilar.pozos@dais.ujat.mx

Resumen En inteligencia artificial, la fusión de creencias se define como un procedimiento que involucra creencias de un conjunto de individuos, posiblemente incompatibles entre sí, y devuelve una sola creencia consistente. Existen varios métodos que fusionan creencias, sin embargo éstos, aunque tienen una base formal robusta, solo han sido comprobados con pocos datos de entrada, por lo que para problemas más complejos que no tienen una solución en un tiempo polinomial (NP-Completos) como lo es el problema de asignación de horarios, todavía no es factible una implementación de solución. Por ello en este artículo se propone la aplicación de la fusión de creencias a este problema, mediante una modificación que permita encontrar una solución factible, en particular para los casos planteados en el concurso organizado por la PATAT.

Palabras clave: Fusión de Creencias, Operador de Fusión, *Timetabling*, Heurística, PATAT.

1. Introducción

En cualquier problema cotidiano se busca siempre la manera de encontrar la solución óptima, la cual se considera como la que consume menos cantidad de recursos (tiempo y espacio) satisfaciendo las expectativas del interesado. Uno de estos problemas es la asignación de horarios escolares, el cual se caracteriza por acomodar sesiones de asignaturas en períodos de tiempo determinados cumpliendo ciertas restricciones para generar un horario válido. Dentro de este ámbito se encuentran los problemas de asignación de horarios universitarios, los cuales poseen además la dificultad de tratar a los alumnos individualmente y se añaden restricciones que cada institución de educación superior considera conveniente incluir para obtener un horario válido.

En el presente artículo se estudia el caso particular de la PATAT (International Conference on the Practice and Theory of Automated Timetabling), la cual organiza un concurso internacional donde se proponen diversos problemas de asignación de horarios y que, entre una de las competiciones, se presenta una variante del problema de asignación de horarios universitarios, en el cual diferentes investigadores y científicos participan para encontrar una solución cercana a la óptima, la cual se sabe que existe [8] y es la que satisface todas las restricciones. Sin embargo, todos los algoritmos propuestos para buscar la solución a dicho problema utilizan métodos no determinísticos, los cuales generan soluciones diferentes a cada iteración, y en consecuencia incertidumbre entre los interesados.

Es por ello que se propone el uso de la fusión de creencias, el cual es un método determinístico, para encontrar una solución que sea invariante durante las ejecuciones del método, haciendo uso de una heurística que posiblemente sacrifique la eficiencia de la solución pero la encuentre en un tiempo que cumpla con los parámetros establecidos por la PATAT. Sin embargo, una de las dificultades es que la fusión de creencias solo se ha llevado a cabo con pocos datos de entrada, y solo se conoce un caso en el que ha sido probado con cantidades "modestas" de información [6], obteniendo resultados con pocos recursos de tiempo y espacio, lo que motiva la búsqueda de una heurística que permita mejorar estos resultados con cantidades mayores de datos.

A continuación se describe el contenido del resto del artículo: en la Sección 2 se introducen las nociones de complejidad computacional y NP-Completo. En la Sección 3 se muestran los posibles enfoques para solucionar problemas NP-Completo y la noción de heurística. En la Sección 4 se describe el problema de asignación de horarios universitario y específicamente el caso de la PATAT. En la Sección 5 se define la fusión de creencias y se muestran ejemplos de operadores de fusión, abordando en especial el operador *PS-merge*. En la Sección 6 se plantea el método a utilizar para el desarrollo de la investigación, además de un primer acercamiento al primer paso, y finalmente en la Sección 7 se da una conclusión sobre la factibilidad y beneficios de esta propuesta de investigación.

2. Complejidad Computacional

La complejidad (o costo) de un algoritmo es una medida de la cantidad de recursos que el algoritmo necesita [29]. Los recursos que se estudian comúnmente son:

- **Tiempo:** mediante una aproximación al número de pasos de ejecución que un algoritmo emplea para resolver un problema.
- **Espacio:** mediante una aproximación a la cantidad de memoria utilizada para resolver el problema.

De acuerdo a esto se establecen las siguientes clases de complejidad [25] [12]:

- **Complejidad P:** es la clase de lenguajes reconocibles por una máquina de Turing determinista de una cinta que opera en tiempo polinomial, lo que corresponde intuitivamente a problemas que pueden ser resueltos aún en el peor de sus casos.
- **Complejidad NP:** es la clase de lenguajes reconocibles por una máquina de Turing no determinista de una cinta que opera en tiempo polinomial.

- **Complejidad NP-Completo:** es el subconjunto de NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-Completo [5]. Se consideran como los problemas más difíciles de NP y también son conocidos como *intratables*. Ejemplos de éstos son el problema de asignación de horarios, buscaminas y tetris, entre otros más.

3. Soluciones aproximadas a problemas NP-Completo

Para resolver un problema NP-Completo de tamaño arbitrario, se puede utilizar uno de los siguientes enfoques [3]:

- **Aproximación:** Un algoritmo que rápidamente encuentra una solución que puede no ser la óptima, pero dentro de un cierto rango de error.
- **Probabilístico:** Un algoritmo probabilístico utiliza aleatoriedad para obtener en promedio una buena solución al problema planteado con una pequeña probabilidad de fallar, para una distribución dada de los datos de entrada.
- **Heurísticas:** Un algoritmo que trabaja razonablemente bien en muchos casos. En general son rápidos, pero no existe medida de la calidad de la respuesta.

3.1. Heurísticas

Las heurísticas se pueden definir como técnicas de búsqueda basadas en un supuesto, es decir sin bases formales, que permiten encontrar una solución aceptable pero no necesariamente la óptima, posiblemente a costo de menor eficiencia o eficacia. El uso de heurísticas adecuadas pueden permitir la reducción del tiempo de cómputo guiando la búsqueda de forma que evitemos recorrer el máximo número posible de estados que no conducen a solución alguna.

Una clasificación propuesta en [24] considera a las siguientes heurísticas:

1. **Heurísticas de construcción:** son las que se utilizan para encontrar una solución del problema, tratando desde el inicio de que ésta sea lo más próxima al óptimo que se pueda.
2. **Heurísticas de mejoramiento:** son las que parten de una solución ya conocida y tratan de mejorarla para que se aproxime al óptimo.

4. Problema de asignación de horarios universitario

El problema de asignación de horarios escolar (*Timetabling*) consiste en asignar las sesiones de las asignaturas a los períodos de tiempo por otorgarse a profesores en bloques semanales, de tal manera que, con algunas restricciones básicas a cumplirse, ningún profesor o asignatura tenga más de una sesión en el mismo período y que todas las sesiones de la asignatura estén presentes en el horario semanal [11].

Una variante de este problema es la asignación de horarios universitario, el cual se distingue por considerar a los alumnos individualmente al momento de asignarle sus materias, además de adecuarse a las politicas institucionales que cada organismo de educación superior tiene, lo cual supone una dificultad mayor para generar un horario válido.

En este tipo de problema se pueden identificar dos tipos de restricciones [26]:

1. **Restricciones primarias**: definen la factibilidad o validez de un horario de clases y, por lo general, expresan limitaciones físicas.
2. **Restricciones secundarias**: se refieren a restricciones que no son obligatorias, pero se desean satisfacer al máximo.

De acuerdo a esto se puede llegar a una clasificación del problema de *Timetabling* [23]:

- Cuando se desea obtener un horario que satisfaga todas las restricciones, entonces se clasifica como un *problema de búsqueda.* En esta situación lo fundamental es decidir si existe o no una solución al problema.
- En otro caso, cuando se requiere un horario que satisfaga todas las restricciones primarias y minimice (o maximice según sea el caso) una función que contenga las restricciones secundarias, se considera un *problema de optimización* . El objetivo es encontrar una respuesta con un valor dado de la función objetivo.

Para el caso general del problema de asignación de horarios, éste se define mediante los siguientes elementos:

- conjunto de m clases $c_1, \dots, c_m$
- conjunto de n maestros $t_1, \dots, t_n$
- conjunto de p periodos de tiempo 1,...,p
- matriz R_{m*n}, denominada *matriz de requerimientos*, donde r_{ij} es el número de sesiones dadas por un maestro t_j a una clase c_i.

Para esto, la formulación matemática es como sigue [4]:

Encontrar $x_{ijk}(i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, p)$ sujeto a las siguientes restricciones:

$$\sum_{k=1}^{p} x_{ijk} = r_{ij} \quad (i = 1, \dots, m; j = 1, \dots, n) \tag{1}$$

$$\sum_{j=1}^{n} x_{ijk} \leq 1 \quad (i = 1, \dots, m; k = 1, \dots, p) \tag{2}$$

$$\sum_{i=1}^{m} x_{ijk} \leq 1 \quad (j = 1, \dots, n; k = 1, \dots, p) \tag{3}$$

$$x_{ijk} \in \{0, 1\} \quad (i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, p) \tag{4}$$

Donde $x_{ijk} = 1$ si la clase c_1 y el maestro t_j están disponibles en el periodo k, y $x_{ijk} = 0$ en caso contrario.

4.1. Problema de asignación de horarios universitario de la PATAT

La conferencia internacional PATAT [21], se lleva a cabo por lo regular cada dos años y consiste en tratar problemas relacionados a los diferentes tipos de asignaciones de horarios, ya sea educacional, de transporte, empleados o de deportes. En dichas conferencias concursan investigadores que tienen como objetivo comparar y analizar el desempeño de las diversas metaheurísticas en los diferentes problemas de generación de horarios.

En el concurso de la PATAT, se propone un problema simplificado de asignación de horarios universitario, y se consideran las siguientes restricciones primarias y secundarias para generar un horario válido [18]:

- **Restricciones primarias**
  - El salón asignado debe cumplir con requisitos apropiados para desarrollar el evento tales como espacio, laboratorio, computadores, medios audiovisuales, entre otros.
  - Los horarios entregados a los estudiantes no deben presentar conflictos de horarios, es decir, que un estudiante precise asistir a dos eventos diferentes a la misma hora.
  - Asimismo no es conveniente que en un salón se presenten dos eventos diferentes a la misma hora.
- **Restricciones secundarias**
  - La asistencia continuada a varios eventos.
  - Asistir en un día a tan solo un evento.
  - Asistir a un evento que se dicte en la última hora del día.

Para obtener una solución factible a este problema es necesario generar un horario que cumpla con las restricciones primarias en cierto tiempo. Cabe destacar que en el concurso se llevan a cabo varias instancias de la ejecución del algoritmo propuesto para la solución del problema, y los problemas siempre tienen una solución perfecta, es decir un resultado sin violaciones de restricciones, tanto primarias como secundarias.

El ganador del concurso es el que obtenga los mejores resultados a través de todas las instancias, es decir que obtenga menores violaciones de restricciones primarias y secundarias en el tiempo otorgado. En la Tabla 1 se muestran los 5 mejores algoritmos que participaron en el último concurso de la PATAT celebrado en el año 2007 [9] y donde además se puede observar que el ganador desarrolló un algoritmo de búsqueda local con un enfoque de programación restrictiva, el cual es un método no determinístico y tiene como desventaja que se obtienen diferentes resultados en cada iteración.

Posición Oficial en la Competencia	Algoritmo	Autor(es)
1	Hibrido de Búsqueda de Gran Vecindad y Programación Constrictiva	Cambazard et al. [7]
2	Hibrido de Búsqueda Tabú y Búsqueda Local Iterativa	Atsuta et al. [17]
3	Hibrido de Búsqueda Local Estocástica y Búsqueda Tabú	Chiarandini et al. [16]
4	Optimización de Colonia de Hormigas	Nothegger et al. [1]
5	Hibrido de Búsqueda Avante Iterativa, Escalada de la Colina y Gran Diluvio	Müller [27]

Tabla 1. 5 mejores algoritmos participantes del concurso de la PATAT 2007

En la Tabla 2 se puede observar las ejecuciones en su primera instancia de los 5 primeros lugares del concurso de la PATAT 2007, y donde se muestran las restricciones primarias y secundarias no satisfechas, con lo que se muestra la irregularidad de los resultados obtenidos.

SCORES ACHIEVED BY THE ALGORITHMS (Distance to Feasibility, Soft Cost) (10 runs per instance, sorted in ascending order)									
ATSUTA ET AL.		CAMBAZARD ET AL.		CHIARANDINI ET AL.		NOTHEGGER ET AL.		MULLER	
DtF	Soft Cost	DtF	Soft Cost	DtF	Soft Cost	DtF	Soft Cost	DtF	Soft Cost
0	61	0	571	0	1482	0	15	0	1861
0	121	0	689	0	1486	0	29	0	1893
0	708	0	696	0	1609	0	410	0	2054
0	927	0	760	0	1676	295	1121	19	1747
0	985	0	821	0	1686	572	1513	20	2298
5	2	0	932	0	1705	583	1406	25	1865
5	826	0	934	0	1736	674	1685	26	1639
5	906	0	1056	0	1904	741	1456	31	1824
16	1059	0	1127	0	1916	790	1346	50	2084
52	881	0	1248	0	2105	800	1731	51	2005

Tabla 2. Resultados ganadores del concurso de la PATAT en su primera instancia

Cabe destacar que en cada competencia realizada por la PATAT se añaden nuevas restricciones al problema de asignación de horarios así como variantes del mismo problema [22], lo que ocasionaría que algoritmos como los anteriormente analizados queden mal situados ante las modificaciones del problema.

De igual manera las instancias del problema del concurso de la PATAT pueden ser consideradas para realizar una evaluación comparativa y medir la calidad de los resultados obtenidos por los algoritmos desarrollados [2], lo cual permite corroborar que los algoritmos cada vez son más efectivos en la solución de problemas de generación de horarios universitarios.

5. Fusión de creencias

La fusión de creencias se define como la operación de combinar información contenida en un conjunto de bases de conocimiento o creencias (posiblemente incompatibles entre sí) obtenidas de diferentes fuentes para producir una única base de creencias consistente [19]. Esta operación es de gran interés cuando se desconoce el estado general de un entorno, ya que puede determinar el estado "verdadero" del entorno a partir de las diferentes creencias de los agentes.

En este procedimiento importa lo que opine la mayoría de los individuos involucrados, por lo que siempre se obtendrá mayor satisfacción para el entorno multiagente.

Para realizar la fusión de creencias se hace uso de la terminología estándar de la lógica proposicional, y se añaden los siguientes conceptos:

- **Base de creencias:** es un conjunto finito de fórmulas proposicionales del lenguaje que representa las creencias de un agente.
- **Estado o Interpretación:** es una función que mapea los átomos del lenguaje hacia los valores veritativos: verdadero o falso. Se utiliza el valor numérico 1 para denotar verdadero y el valor numérico 0 para denotar falso.
- **Modelo:** una vez evaluados los conectivos de una fórmula proposicional bajo una interpretación, si la fórmula tiene un valor veritativo verdadero (1), se dice que la interpretación es un modelo de dicha fórmula.
- **Perfil de creencias:** denota las creencias del grupo de agentes que están involucrados en el proceso de fusión, donde dos agentes pueden tener la misma creencia.

5.1. Operadores de fusión

Para llevar a cabo la fusión de creencias se requiere lo que se conoce como operador de fusión, el cual es el encargado de devolver una base de conocimiento sin contradicciones, que represente al conjunto de bases involucradas. En el caso de que cada creencia corresponda al conocimiento de un agente, al fusionar estas bases se obtendrá una base que represente el conocimiento del grupo [10].

Se consideran dos tipos de operadores de fusión los cuales son [13]:

1. **De mayoría (Majority):** Se esfuerza en satisfacer la mayoría de las creencias de los agentes.
2. **Arbitrario (Arbitration):** Intenta satisfacer las creencias de cada agente al mejor grado posible.

Ejemplos de operadores de fusión mayoritarios son *CMerge* [15], $\triangle_{\Sigma}$ [13], *PS-merge* [19] y de uno arbitrario es $\triangle_{GMax}$ [14]. Una comparación se hace en [10] donde se muestra que se obtienen resultados similares entre ellos y que los operadores de fusión mayoritarios como el *PS-merge* mediante una modificación se pueden definir también como operadores arbitrarios.

5.2. *PS-merge*

Este operador de fusión maneja la noción de Satisfacción Parcial (Partial Satisfiability), el cual tiene como características principales las siguientes [28]:

- Es un operador de fusión de mayoría.
- Considera bases de conocimiento inconsistentes.
- Extrae información de aquellas piezas que no causaron la inconsistencia.
- Es sensible a la frecuencia de creencias de un agente, es decir, toma en cuenta la repetición de ellas.
- No está basado en una medida de distancia entre modelos.
- Emplea el mismo operador para todos los contextos, lo que facilita la automatización del proceso.

Para la definición de satisfacción parcial, se considera un lenguaje normalizado, de tal manera que cada base es tomada como la forma normal disyuntiva (DNF) de la conjunción de sus elementos, es decir si $K = \{Q_1, \ldots, Q_n\}$ es una base de creencias, ésta se identificará como: $Q_K = DNF(Q_1 \wedge \ldots \wedge Q_n)$.

Mediante la Fórmula 5 es posible medir la satisfacción parcial de una base de creencias para un estado w, si la base es una conjunción, es decir $Q_K = C_1 \wedge \ldots \wedge C_s$

$$w_{ps}(Q_K) = max\{\sum_{i=1}^{s} \frac{w(C_i)}{s}, \frac{n - |P(\bigwedge_{i=1}^{s} C_i)|}{2n}\} \tag{5}$$

Donde:

- $w \in W$, y W es el conjunto de todos los posibles estados.
- Q_K es forma normal disyuntiva de una base de creencias.
- P es una función que toma como parámetro una fórmula y devuelve el número de átomos que aparecen (ocurren) en dicha fórmula.
- n es el número de átomos del lenguaje considerado.
- s es el número de literales que pertenecen en la conjuncion.

Si Q_K es una disyunción $D_1 \vee \ldots \vee D_r$ donde cada D_i es una conjunción de literales, entonces la satisfacción parcial se obtiene usando la Fórmula 6:

$$w_{ps}(Q_K) = max\{w_{ps}(D_1), \ldots, w_{ps}(D_r)\} \tag{6}$$

6. Método propuesto

En la Figura 1 se muestra en forma de diagrama los pasos del método propuesto para solucionar el problema de la generación de horarios formulado por la PATAT utilizando un operador de fusión de creencias.

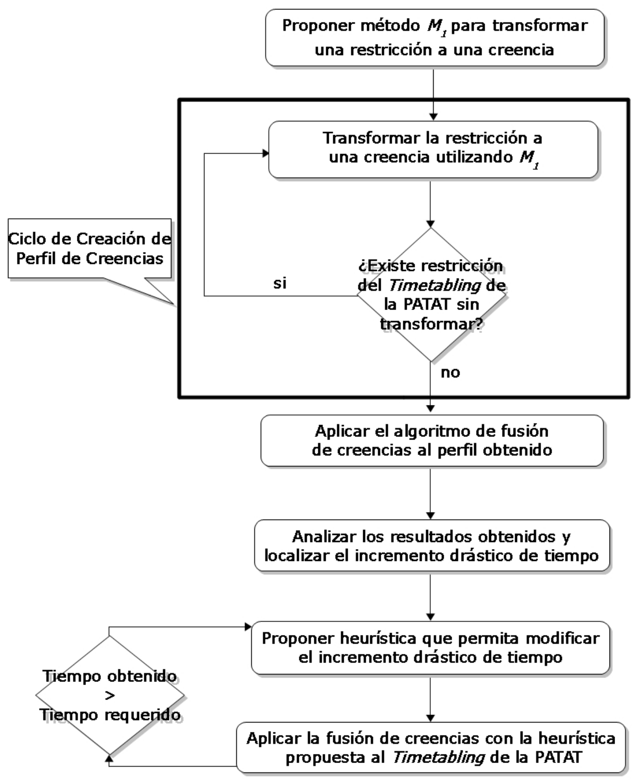

Figura 1. Diagrama del método propuesto

A continuación se describen los avances en la implementación del método mencionado:

En el algoritmo 1 se muestra el pseudocódigo de *PS-merge*, del cual se pretende partir para realizar las modificaciones que conllevarán a encontrar la heurística que permita reducir el tiempo de cómputo a problemas de mayor cantidad de datos de entrada.

Algoritmo 1: Algoritmo *PS-merge*

Entrada:
E: Perfil de Creencias
V: Número de variables del perfil E
B: Número de bases de E
D: Vector de números de disyunciones de cada base en E
L: Matriz de ocurrencias de literales para cada disyunción de cada base de E
Resultado:
$Solution\_Set$: El conjunto de estados en $PS\text{-}Merge(E)$

begin
    $Solution \leftarrow \emptyset$
    $Max\text{-}Sum \leftarrow 0$
    $W \leftarrow$ Matriz cuyas filas son todos posibles estados para V variables.
    for $s = 1 \dots B$ **do**
        $ID_s \leftarrow \sum_{k=1}^{s-1} D_k + 1$
    for $i = 1 \dots 2^V$ **do**
        $Sum \leftarrow 0$
        for $s = 1 \dots B$ **do**
            $ps\text{-}disjunct \leftarrow \emptyset$
            for $d = ID_s \dots ID_s + D_s$ **do**
                $satisfied \leftarrow 0$
                $conjuncts \leftarrow 0$
                $vars\text{-}not\text{-}appearing \leftarrow 0$
                for $j = 1 \dots V$ **do**
                    if $W_{i,j} = 1$ **then**
                        $satisfied \leftarrow satisfied + L_{d,2j-1}$
                    if $W_{i,j} = 0$ **then**
                        $satisfied \leftarrow satisfied + L_{d,2j}$
                    $conjuncts \leftarrow conjuncts + L_{d,2j-1} + L_{d,2j}$
                    if $L_{d,2j} = 0$ ***and*** $L_{d,2j-1} = 0$ **then**
                        $vars\text{-}not\text{-}appearing \leftarrow vars\text{-}not\text{-}appearing + 1$
                $ps\text{-}disjunct \leftarrow ps\text{-}disjunct \cup \{max(\frac{satisfied}{conjuncts}, \frac{vars-not-appearing}{2V})\}$
            $PS \leftarrow max(ps - disjunct)$
            $Sum \leftarrow Sum + PS$
        if $Sum > MaxSum$ **then**
            $Solution \leftarrow \{i\}$
            $MaxSum \leftarrow Sum$
        else if $Sum = MaxSum$ **then**
            $Solution \leftarrow Solution \cup \{i\}$
    $Solution_Set = \{i$esima fila de $W | i \in Solution\}$
end

El algoritmo ha sido implementado en Matlab[3] y probado con todos los ejemplos presentados en [19] con rendimiento promedio de 0.4 segundos, y en el caso de otros con mayor número de variables, por ejemplo 15 variables con 35 bases y un total de 130 disyunciones, se obtuvó un rendimiento promedio de 1400 segundos. Los resultados indican que los problemas de modesto tamaño pueden ser tratados usando hardware convencional y tiempos cortos de cómputo.

El primer paso del método contempla transformar las restricciones del problema de asignación de horarios a creencias del método de fusión de creencias. Para esto el problema puede ser formulado como sigue, si:

- $E = \{E_1, E_2, ..., E_{n_E}\}$ es el conjunto de eventos por agendarse en un conjunto de salones $R = \{R_1, R_2, ..., R_{n_R}\}$.
- $S = \{S_1, S_2, ..., S_{n_S}\}$ es el conjunto de estudiantes que asisten a los eventos.
- $\{F_1, F_2, ..., F_{n_F}\}$ es el conjunto de variables que representan las características de los salones, tales como: poseer cañón, poseer pintarrón, etc.
- $R_i = \{F_1^{'}, F_2^{'}, ..., F_{n_{R_i}}^{'}\}$ donde $F_j^{'} \in F$ para $j = 1, ..., n_{R_i}$ y n_{R_i} es el número de características satisfechas por cada salón R_i.
- $\{N_1, ..., N_{n_R}\}$ es el conjunto de números enteros indicando el máximo de estudiantes que cada salón puede acomodar.
- $E_i = \{F_1^{''}, F_2^{''}, ..., F_{n_{E_i}}^{''}\}$ donde $F_j^{''} \in F$ para $j = 1, ..., n_{E_i}$ y donde n_{E_i} es el número de la característica requerida por el evento E_i.
- $S_i = \{E_1^{'}, E_2^{'}, ..., E_{n_{S_i}}^{'}\}$ donde $E_j^{'} \in E$ para $j = 1, ..., n_{S_i}$ y donde n_{S_i} es el número de eventos a los que el estudiante S_i asiste.
- $T = \{T_1, ..., T_{45}\}$ es el conjunto de bloques de tiempo correspondientes de lunes a viernes divididos en 9 periodos de tiempo al día.

Una solución factible es un conjunto de pares $\{(T'_1, R'_1), ..., (T'_{n_E}, R'_{n_E})\}$ tales que:

$$T'_i \in T \text{ y } R'_i \in R \tag{7}$$

$$\neg(T'_i = T'_j) \text{ si } E_i \in S_k \text{ y } E_j \in S_k \text{ y } \neg(i = j) \tag{8}$$

$$E_i \subseteq R'_i \text{ y } |\{S_j | j = 1, ..., n_S \text{ y } E_i \in S_j\}| \leq N_k \text{ y } R'_i = R_k \tag{9}$$

$$\neg\exists i \exists j \ (\neg(i = j) \wedge T'_i = T'_j \wedge R'_i = R'_j) \tag{10}$$

La descripción intuitiva de la Fórmula 8 es: Si existen los eventos E_i y E_j a los que un estudiante S_k debe asistir, entonces no puede asignarse el mismo bloque de tiempo a esos dos eventos, puesto que el estudiante S_k no podrá asistir a ambos eventos E_i y E_j debido a que se realizan al mismo tiempo. La Fórmula 10 se deduce de que no existen eventos i y j tales que sean diferentes y que tengan asignado el mismo salón al mismo tiempo.

Además las siguientes acciones se realizan:

[3] El código y ejemplos pueden ser descargados desde: http://www.utm.mx/~vero0304/PSMerge.htm

- el número de estudiantes para cada evento es calculado y almacenado, $n_i = |\{S_j | j = 1, ..., n_S \text{ y } E_i \in S_j\}|$
- una lista de posibles salones es creado para cada evento, $e_i = \{R_j | E_i \subseteq R_j \text{ y } n_i \leq N_j\}$

Una forma de reducir el problema podría ser eliminando el conjunto de características, de eventos y de requerimientos de capacidad, esto definiendo el nuevo conjunto $\{e_1, ..., e_{n_E}\}$ que incluye la información eliminada. Este nuevo conjunto de eventos será usado para satisfacer las restricciones primarias.

De esta manera se pueden crear los parámetros que utilizará el operador PS-$Merge$, los cuales pueden establecerse como sigue:

- crear el conjunto de variables proposicionales V=$\{R_{i,s,r} | R_r \in e_i,\ s = 1, ..., 45$ y $i = 1, ..., n_E\}$,
- para cada evento e_i crear la base de creencias $K_i = \bigvee_{R_{i,s,r} \in V} R_{i,s,r}$,
- crear la base $K_{n_E+1} = \bigwedge_{R_{i_1,s,r}, R_{i_2,s,r} \in V} R_{i_1,s,r} \wedge \neg(i_1 = i_2) \rightarrow \neg R_{i_2,s,r}$. Por ejemplo, si el evento 5 puede ser colocado solamente en el salón 2 y 4, y el evento 8 puede ser colocado solamente en el salón 1 y 2, algunos conjuntandos de esta fórmula son: $\neg R_{5,1,2} \vee \neg R_{8,1,2}, \neg R_{5,2,2} \vee \neg R_{8,2,2}, ..., \neg R_{5,45,2} \vee \neg R_{8,45,2}$,
- crear la base $K_{n_E+2} = \bigwedge_{R_{i_1,s,r_1}, R_{i_2,s,r_2} \in V, j \in \{1,...,n_S\}} R_{i_1,s,r_1} \wedge \neg(i_1 = i_2) \wedge E_{i_1}, E_{i_2} \in S_j \rightarrow \neg R_{i_2,s,r_2}$. Por ejemplo, si el salón 2 puede ser usado para alojar el evento 5, el salón 3 puede ser usado para alojar el evento 3, y el estudiante 90 asiste al evento 5 y 3, para esta fórmula algunas de los conjuntandos son: $\neg R_{3,1,3} \vee \neg R_{5,1,2}, \neg R_{3,2,3} \vee \neg R_{5,2,2}, ..., \neg R_{3,45,3} \vee \neg R_{5,45,2}$.

De esta forma se puede realizar la operación PS-$Merge(\{K_1, ..., K_{n_E+2}\})$ y así obtener las interpretaciones que maximicen la satisfacción de las bases de creencias.

Esta sería una forma tentativa de resolver el *Timetabling* universitario usando la fusión de creencias, específicamente mediante el operador PS-$Merge$, a cuya implementación se puede adaptar una heurística para reducir el tiempo de búsqueda.

7. Conclusión

No existen antecedentes de la aplicación de la fusión de creencias a problemas NP-Completos, incluyendo el problema de asignación de horarios universitario, por lo que soluciones a estos problemas mediante este método se consideran aún imposibles en tiempos que satisfagan las expectativas, en este caso las de la PATAT. Por ello se considera un reto interesante la modificación de este procedimiento mediante una heurística adecuada con la finalidad de dar solución al *Timetabling* universitario mediante un algoritmo determinístico, lo que resultaría innovador y permitiría uniformidad en los resultados.

Se pretende utilizar el operador de fusión *PS-merge* para resolver el problema de asignación de horarios universitario, ya que debido a las características de considerar bases inconsistentes y a la extracción de información de creencias que no causan la inconsistencia, se puede llevar a cabo una mayor interpretación de posibles estados en los que puede estar una base de creencias, y al ser también un operador de fusión mayoritario permitirá resolver los conflictos entre las creencias de manera natural. Además tiene la ventaja que ya se ha implementado este operador [20], con lo que se tiene un referente base para continuar con la investigación.

Mediante el método propuesto se pretende obtener un primer acercamiento a la solución del *Timetabling* universitario usando la fusión de creencias, con lo que la primera intención es obtener un algoritmo determinístico sin considerar el tiempo de cómputo necesario para realizar la operación, permitiendo de esta manera la posible aplicación de otros operadores de fusión.

De resultar verídica la hipótesis de investigación se podrá participar en el concurso de la PATAT, y así mostrar a escala internacional la potencial utilidad que tiene la fusión de creencias en la solución a problemas en contextos complejos, además de beneficiar a las instituciones de educación superior que asemejan su sistema de asignación de horarios escolares a los que se presentan en dicho concurso. De igual forma al concluir este trabajo, se podrá comparar la eficacia de los métodos deterministas con los no deterministas, además contribuiría a la obtención de soluciones cada vez mejores a los problemas de optimización, lo que motivaría la aplicación de la fusión de creencias a otros problemas de este tipo.

Referencias

1. Alfred Mayer, Clemens Nothegger, Andreas Chwatal, and Günther Raidl. Solving the Post Enrolment Course Timetabling Problem by Ant Colony Optimization. 2008.
2. Oscar A. Chávez Bosquez. Búsqueda tabú aplicada a un problema NP-Completo: *Timetabling* en la DAIS. Master's thesis, División Académica de Informática y Sistemas - Universidad Juárez Autonóma de Tabasco, 2009.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.
4. D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, Febrero 1985.
5. M. D. Johnson Garey. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Bell Telephone Laboratories, 1979.
6. Nikos Gorogiannis and Anthony Hunter. Implementing semantic merging operators using binary decision diagrams. *Int. J. Approx. Reasoning*, 49(1):234–251, 2008.
7. Hadrien Cambazard, Emmanuel Hebrard, Barry OSullivan, and Alexandre Papadopoulos. Submission to ICT: Track 2. 2008.
8. IDSIA. International Timetabling Competition. 2003. Obtenido el 7 de marzo del 2009 desde `http://www.idsia.ch/Files/ttcomp2002/oldindex.html`.
9. ITC. International Timetabling Competition - Finalists. 2008. Obtenido el 4 de julio de 2009 desde `http://www.cs.qub.ac.uk/itc2007/winner/finalorder.htm`.
10. Jesús Alejandro Hernández Tello. Fusión de creencias basándose en Satisfacción Parcial. 2008.

11. John Fredy Franco Baquero, Eliana Mirledy Toro Ocampo, and Ramón Alfonso Gallego Rendón. Problema de asignación óptima de salones resuelto con búsqueda tabú. 2008.
12. Richard M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
13. Sébastien Konieczny and Ramón Pino Pérez. Merging with integrity constraints, 1999.
14. Sébastien Konieczny and Ramón Pino-Pérez. On the logic of merging, 1998.
15. Jinxin Lin and Alberto O. Mendelzon. Knowledge base merging by majority. In *In Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer, 1994.
16. Marco Chiarandini, Chris Fawcett, and Holger H. Hoos. A Multiphase Modular Heuristic Solver for Post Enrollment Course Timetabling. 2008.
17. Mitsunori Atsuta, Koji Nonobe, and Toshihide Ibaraki. ITC-2007 Track2: An Approach using General CSP Solver. 2008.
18. Metaheuristics Network. International timetabling competition. 2004. Obtenido el 20 de mayo de 2009 desde http://www.idsia.ch/Files/ttcomp2002/IC_Problem/node1.html.
19. Pilar Pozos Parra and Verónica Borja Macías. Partial Satisfiability-Based Merging. In *MICAI*, pages 225–235, 2007.
20. Pilar Pozos Parra and Verónica Borja Macías. PSMerge. 2009. Obtenido el 27 de mayo de 2009 desde http://www.utm.mx/~vero0304/PSMerge.htm.
21. PATAT. Practice and Theory in Automated Timetabling. 2004. Obtenido el 8 de marzo de 2009 desde http://www.asap.cs.nott.ac.uk/patat/patat-index.shtml.
22. PATAT. Practice and Theory in Automated Timetabling. 2007. Obtenido el 31 de marzo de 2009 desde http://www.cs.qub.ac.uk/itc2007/postenrolcourse/course_post_index_files/Inputformat.htm.
23. Andrea Schaerf. A Survey of Automated Timetabling. *Artificial Intelligence Review*, 13:87–127, 1995.
24. Silvia A. Ramos. Heurísticas y Problemas Combinatorios. In *Modelos y Optimización I*, 2007.
25. Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.
26. Marcos Gil Tallavó and Amadís Antonio Martínez. Algoritmo basado en tabu search para el problema de asignación de horarios de clases. 2006.
27. Tomáš Müller. ITC2007: Solver Description. 2008.
28. Verónica Borja Macías. Belief Merging basado en Partial Satisfiability. 2007.
29. Víctor Valenzuela Ruz. *Manual de Análisis y Diseño de Algoritmos*. 2003.