

Problema del School Timetabling y algoritmos genéticos: una revisión

Timetabling School problem and genetic algorithms: a review

Mauricio Andres Guerra Cubillos*

Erwin Hamid Pardo Quiroga**

Roberto Emilio Salas Ruiz***

Fecha de recepción: 15 de mayo de 2013

Fecha de aceptación: 15 de junio de 2013

Resumen

En este artículo se presenta de manera general el problema “School Timetabling”, se parte de una definición del mismo, su clasificación, su complejidad computacional, para luego entrar a revisar las diferentes técnicas con las cuales se puede solucionar el mismo y como último se entra a revisar una de estas técnicas como son los algoritmos genéticos (AG) que fue la escogida para darle solución.

Palabras claves: Timetabling, Complejidad computacional, algoritmos metaheurísticos, Algoritmos Genéticos.

* Universidad Distrital Francisco José de Caldas. Correo electrónico: maoguerra007@hotmail.com

** Universidad Distrital Francisco José de Caldas. Correo electrónico: erwinpardoq@hotmail.com

*** Universidad Distrital Francisco José de Caldas. Correo electrónico: resalasr@udistrital.edu.co

Abstract

This paper shows an overview of the problem, "School Timetabling", it treats first a definition of this problem, their classification, their computational complexity, and then it makes a review the different techniques which can be solved and the last one review one of these techniques such as genetic algorithms (GA), which was chosen to solution it.

Key words: Timetabling, computational complexity, metaheuristics algorithms, genetic algorithms.

1. Introducción

Timetabling, hace referencia a la calendarización, a la organización específicamente de horarios de distintas organizaciones tales como, hospitales, colegios, centros de transporte, universidades, etc., y es uno de los problemas más abarcados en la actualidad, pero también está situado dentro de uno de los problemas más difíciles de tratar ya que varios autores afirman que el problema del Timetabling está situado dentro de la complejidad computacional en los problemas NP-Complejos.

En este caso, se tratara a fondo el problema específico de la organización de los horarios en los colegios, dentro de los cuales para la asignación de cursos, profesores y periodos de tiempo correctos en la realización de horarios académicos, existe la necesidad de tener en cuenta una gran variedad de restricciones, tales como la disponibilidad de los profesores en periodos de tiempo, la asignación correcta de un profesor para dictar un curso, el cruce de entre cursos en un mismo salón, entre otras, y es por esto que el problema va tomando mayor tamaño.

1.1. ¿Por qué se hace referencia a los colegios en general y no a uno en específico?

La organización de los colegios en todas partes es bastante similar, es decir, el modo en que se distribuyen los salones, los cursos que se dictan, la distribución de los grupos de estudiantes en los distintos salones, y por supuesto, la disponibilidad horaria y académica de los maestros, no distan mucho dentro de cada colegio, pero el punto principal para hacerlo de este modo, y teniendo en cuenta lo anterior, es que se considera que si la solución que se dará puede ser aplicada para un colegio, entonces lo más probable y conveniente es que esta servirá también para los demás colegios similares.

Hoy en día, existen bastantes instituciones que realizan el trabajo de la organización de horarios de manera manual, ya sea por cuestiones de desconocimiento o simplemente de no querer dar un salto tecnológico en ese sentido, en cualquiera de los casos, el trabajo manual se convierte en un problema de tiempo, personal y costos, es por eso, y porque en la actualidad la tecnología está a la vanguardia, que cada organización, en este caso, cada colegio, deberían contar con herramien-

tas que le permitan optimizar sus procesos y realizarlos más eficientemente.

Por lo expuesto, el presente artículo presenta una revisión del problema del timetabling y de los algoritmos genéticos como técnica para solucionar al mismo.

2. Problema del Timetabling

Conocido en la literatura de habla hispana, como un problema de Programación horaria o Calendarización; en él existen recursos que deben ser asignados, en instantes o bloques de tiempo determinados, teniendo en cuenta requisitos y condiciones ("restricciones"). En aras a que se comprenda mejor este problema se presentan algunas definiciones:

- **Definición 1.** Zhipeng Lu y Jin-Kao Hao, definen timetabling como: "Asignar un número de eventos, cada uno con ciertas características, a un número limitado de recursos sujeto a restricciones" [1]. Anterior a ellos Anthony Wren en 1996, determina el timetabling, como un caso especial de Programación (scheduling), a esta la define como:
- **Definición 2:** "la asignación, sujeta a restricciones, de un grupo de recursos a objetos ubicados en tiempo y espacio, de tal manera que se satisfagan un conjunto de objetivos deseados"[2]

Este particular problema se puede apreciar claramente en diferentes escenarios en el mundo, por lo que es objeto de estudio por parte de grupos de investigación, para optimizar los resultados y lograr soluciones automatizadas de alta calidad. En los diversos sectores donde se puede ver la problemática de la asignación de recursos y la calendarización se encuentran: los deportes, empresas, transporte aéreo, educación, entre otros; de tal manera que la serie de congresos interna-

cionales PATAT² determina las grandes áreas de estudio de investigación actualmente, al igual que su coorganizador el grupo de investigación ASAP [3] da a conocer algunos tipos de timetabling, los más conocidos:

- **Transport Timetabling:** aquí se encuentran las diferentes problemáticas en la asignación de rutas de los conductores de buses de transporte público o privado, trenes (por ej., [6, 12]) y/o aviones (conocido como Airline Schedule en [7]); además con la programación y previsibilidad de horarios de salida y la optimización de operaciones de llegada principalmente en aeropuertos, donde los controladores de las pistas de aterrizaje de aeropuertos modernos, se realizan de manera manual y requieren de un alto grado de precisión en poco instantes de tiempo.
- **Sports Timetabling:** (también conocido como Sports scheduling, en [3]). En esta interesante variante del problema mencionado, los deportes en especial el fútbol, cuenta con una característica como son los diferentes tipos de enfrentamientos entre equipos, ya sea de uno contra uno, ida y vuelta, o torneos de todos contra todos; los cuales requieren un tipo de programación de encuentros distinta, otra inconveniente por ejemplo, en épocas de Navidad y Año nuevo se realizan encuentros muy seguidos, y los simpatizantes deben desplazarse por largas distancias, lo cual requiere otro tipo de optimización que beneficie al espectáculo, taquillas, transmisión por cable, etc. Un ejemplo se puede ver en [5].
- **Employee Timetabling and Rostering:** primordialmente esta modalidad trata de los turnos particularmente en el sector de la salud (Nurse Rostering en [9]), puesto

2 Practice and Theory of Automated Timetabling - (Serie de Congresos Internacionales) <http://www.patat2012.com/index.html>

que, enfermeras y médicos deben cumplir diferentes turnos de trabajo, se deben equilibrar las cargas de trabajo, teniendo en cuenta restricciones duras y blandas, tales como, exigencia mínima de enfermeras, días de descanso, etc. [9]. También algunas empresas, imparten horarios de trabajo a sus empleados, por lo tanto, en turnos y/o horarios, se deben tener en cuenta distintas restricciones, en [10] encontramos algunas como: límite máximo de horas o turnos, existencia de interrupciones, existencia de trabajadores temporales y de tiempo completo, periodos de planificación, disponibilidad y preferencias del trabajador, entre otras. (Véase un ejemplo de un Call Center en [8]).

- **Educational Timetabling:** en el ámbito educativo, los principales problemas son los de programación de horarios tanto en colegios (School Timetabling) como en universidades (University or Course Timetabling), este tipo de programación de materias es subdividida por Cabezas J. [4] en "EB-CTT"³ y "CB-CTT"⁴; todos los anteriores requieren una eficiente asignación de recursos respetando instantes de tiempo establecidos, esto implica una serie de restricciones y preferencias derivadas de personas, instituciones, reglamentos u otras. Este tipo de programación tiene una gran complejidad, por la cantidad de variables y limitaciones, de tal manera que existe un espacio para la investigación y el desarrollo [3]. Otro campo subsecuente de la educación tiene que ver con la carga de exámenes y su calendarización (Exam o Examination Timetabling).

Generalmente, para problemas en la educación, se manejan dos tipos de restricciones, Larrosa [11] las describe:

- **Restricciones Duras (Obligatorias):** son condiciones de obligatorio cumplimiento, de tal manera que la violación a alguna origina un horario no valido. Son espaciales (p. ej.: la cantidad de estudiantes no debe superar la capacidad de un aula) o temporales (p. ej.: un docente no debe tener asignado dos o más cursos en un mismo bloque de tiempo), de esta manera se dice que toda restricción dura se debe satisfacer.
- **Restricciones Blandas (Deseadas):** son restricciones que denotan preferencias del usuario, se busca que se cumplan en la medida de lo posible (p. ej.: no se desea que un profesor se traslade a diferentes aulas cuando se tiene una clase de dos periodos consecutivos). La violación de alguna de estas seguirá ocasionando un horario factible, pero no de la calidad deseada.

3. Complejidad computacional

En la búsqueda de soluciones a una gran cantidad de problemas en el área de la computación, se notó que hay algunos más difíciles de resolver que otros, teniendo en cuenta principalmente el tiempo de procesamiento y la cantidad de espacio en memoria que se requiere para resolver el problema, sabiendo esto, la complejidad del problema se puede clasificar en 3 tipos principales: P, NP y NP-COMPLETOS. [33]

- **Problemas P:** los problemas de tipo P son aquellos que son solucionables en tiempo polinomial, es decir problemas sencillos que se pueden resolver fácilmente de forma práctica tales como multiplicaciones, funciones lineales, cuadráticas, etc. Todo problema que se encuentra en

3 Enrollment-Based Course Timetabling – (Calendarización por Materias Basado en Inscripciones)

4 Curriculum-Based Course Timetabling – (Calendarización por Materias Basado en Plan de Estudios)

P hace parte de los problemas situados en NP [33].

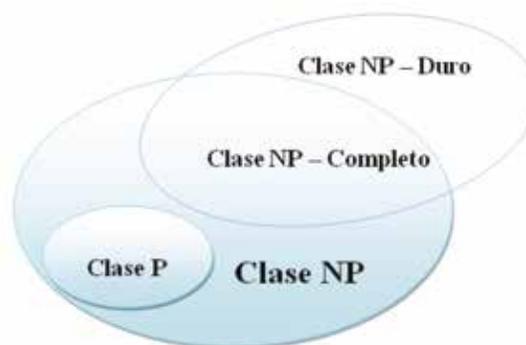
- **Problemas NP:** los problemas de tipo NP tienen un concepto similar al de los problemas P, ya que son resueltos en un tiempo polinomial, la diferencia es que son problemas NO DETERMINISTICOS, es decir, suelen ser resueltos mediante el uso de una maquina de turing no determinista, con esto se dice que no se sabe cuál es el resultado que se va a dar, y el tiempo de procesamiento depende de la cantidad de datos de entrada. Este tipo de problema contiene los problemas que también son contenidos dentro de las otras clases (P, NP-c).

Se dice que contiene los problemas P, porque es posible la aplicación de un algoritmo polinomio que compruebe que la solución dada es válida o no, entonces, en P los problemas se resuelven en tiempo polinómico y en NP los problemas se comprueban en tiempo polinómico.

Principalmente esta clase abarca problemas de búsqueda y optimización como la utilización de grafos [34].

- **Problemas NP-Completo:** los problemas NP-COMPLETOS, son también problemas NP, es decir, los problemas NP pueden ser reducidos a problemas NP-COMPLETOS, y el tiempo computacional requerido aumenta exponencialmente con el tamaño que tenga el problema [35]. Como se dijo antes NP abarca el conjunto completo de problemas (figura 1), entonces se puede decir que los problemas NP-COMPLETOS son los más difíciles de resolver dentro del conjunto NP, y no están presentes dentro de los problemas P.

Figura 1. Complejidad Computacional



Fuente: elaboración propia.

Los problemas NP-COMPLETOS podrían parecer tan complejos que algunos dirían que son intratables, pero en realidad no se ha podido comprobar esto.

También se puede decir que este tipo de problemas son equivalentes entre sí. Si existe una solución para un problema NP-COMPLETO, entonces existe para cualquier problema de este tipo, y si por el contrario se comprobara que un problema NP-COMPLETO no tiene solución, entonces ninguno la tendría [34].

Para el caso específico del problema de estudio Timetabling, la gran mayoría de autores coinciden que está ubicado en la clase de problemas NP-COMPLETOS [33] [35] [36], por su gran dificultad de resolución.

4. Solución al problema del Timetabling

4.1. Técnicas tradicionales

Son métodos que recorren todo el espacio de búsqueda, por tanto se dice que encuentran todas las soluciones al problema, se les considera como algoritmos completos. Sin embargo estos métodos, dependen del número de variables que intervienen en el problema [13].

En este Grupo se encuentran: Programación Lineal, Programación entera, backtracking, entre otras.

4.2. Técnicas no tradicionales

En contraste a las anteriores, estas no encuentran todas las posibles soluciones a un problema, solo acotan o reducen el espacio de búsqueda, por lo tanto se dice que son métodos incompletos [15].

Dentro de este grupo están: Recocido Simulado (SimulatedAnnealing), Algoritmos Evolutivos (EvolutionaryAlgorithms), búsqueda tabú (TabuSearch), algoritmos voraces (GRASP), redes neuronales (Neuronal Networks), entre otras. Este tipo de métodos son conocidas como "metaheurísticos", miremos en detalle algunas técnicas más usadas:

- **Templado o recocido Simulado (SimulatedAnnealing):** el recocido simulado o "SimulatedAnnealing" en inglés, fue propuesto y estudiado en primera instancia por Metrópolis en el año de 1953 [23]. Es un método heurístico que tiene más relación con la termodinámica (similar al proceso de enfriamiento del metal), que con la optimización. Como es una variante de la búsqueda local, puede quedar atrapado prematuramente en un óptimo local [16]. En cada iteración una

vecindad es generada (Un horario factible se modifica ligeramente de forma aleatoria para crear uno nuevo también factible). Este vecino es aceptado como el actual horario si se considera que tiene baja penalidad. Por el contrario, si este nuevo vecino presenta alta penalidad, se considera para ser aceptada como la actual solución, es decir, como un calendario (horario) acorde a una probabilidad relacionada con un parámetro de control denominado temperatura.

"Las soluciones obtenidas por estas estrategias descendentes, dependen fuertemente de las soluciones iniciales consideradas" [18]. A mayor temperatura, mayor probabilidad de aceptación de soluciones peores, de tal manera que el algoritmo acepta soluciones mucho peores al principio de la ejecución (exploración) pero no al final (explotación).

"Finalmente, cuando la temperatura es tan baja que ningún cambio se acepta, el algoritmo se detiene con el objeto inicial profundamente alterado, de hecho con el objeto que probablemente pueda encontrarse para la propiedad de interés. Esta situación inicial es el cero absoluto del objeto respecto a la propiedad" [17].

Gómez en [18] comenta que las principales desventajas se presentan por el tiempo computacional y la dificultad para ajustar adecuadamente los parámetros que controlan el algoritmo.

- **Aplicaciones en Timetabling:** se pueden ver en [24] y [25].
- **Búsqueda Tabú (TabuSearch):** la metaheurística búsqueda Tabú es introducida y desarrollada por Fred Glover en 1986 [19]. Este método está diseñado para salir del óptimo local, "La filosofía de esta técnica es la creencia de que la elección de una mala estrategia sistemática de búsqueda es mejor que una buena elegida al azar" [4].

Su funcionamiento se ve expuesto en [26] de la siguiente manera: “Una búsqueda con lista tabú o TabooSearch consiste en partir de un candidato al azar (o generado con alguna otra heurística) y modificarlo progresivamente (mediante un segundo algoritmo) hasta que no sea posible obtener mejoras haciendo esa modificación”.

La principal característica de la búsqueda tabú, utiliza una memoria flexible por medio de estructuras simples, de tal manera que dirige la búsqueda de acuerdo a la historia que lleva, es así como el escape de óptimos locales se realiza de manera sistemática y no aleatoria.

“...desde el punto de vista de la Búsqueda Tabú, la memoria flexible envuelve el proceso dual de crear y explotar estructuras para tomar ventaja mediante la combinación de actividades de adquisición, evaluación y mejoramiento de la información de manera histórica...” Glover y Laguna [14].

Restrepo y Velásquez [20] expresan que la memoria es representada mediante una lista tabú, la cual contiene para las mejores soluciones o en su defecto, los movimientos realizados para obtener dicha solución, de esa forma no serán tenidos en cuenta en futuras iteraciones, lo que beneficia a tener un reducido número de soluciones elegibles. Para el problema específico del Timetabling la búsqueda tabú existen implementaciones como en [27] y [28].

- **Colonia de Hormigas (AntColony):** una de las metas-heurísticas más empleadas recientemente para enfrentar problemas de optimización, desde su inicio por Dorigo, Maniezzo y Colomi en la primera mitad de la década de los 90' [21]. Cada hormiga en la colonia realiza inicialmente trayectorias aleatorias en búsqueda de su alimento, al hallarlo estudia

la cantidad y la calidad según Gómez [18] y regresa a su colonia depositando una feromona, que permitirá a otras hormigas seguir el rastro reforzando la intensidad de la feromona y evitando su evaporación, de manera análoga con los problemas de optimización el concepto de evaporación de la feromona es utilizado para evitar que el algoritmo converja a un óptimo local. [20] En contraste si no existiese la evaporación de la feromona, cualquier trayectoria sería igual de atractiva para las hormigas lo que se traduciría en una exploración muy amplia de soluciones.

En general, lo que se pretendió y se pretende aun con el método de la colonia de hormigas se expone en [18] de la siguiente manera:

- “La idea primordial al poner en práctica la Metodología de Colonia de Hormigas es intentar obtener una alta organización y distribución entre las hormigas artificiales para utilizarlas en la administración de la población de agentes artificiales, obteniendo como resultado las mejoras en la solución de problemas de optimización combinatoria.”
- Algunas aplicaciones en Timetabling [18] y [29].

- **GRASP (Greedy Randomize Adaptive Search Procedure):** esta surgió en 1989 gracias a Feo y Resende, y según su autor fue desarrollada para resolver problemas difíciles en el campo de la optimización combinatoria [30]. Esta metodología se desarrolla mediante un proceso iterativo, dividido en dos fases principales, la fase de construcción y la fase de mejoramiento, explicado en [31] de la siguiente manera:

- En la fase de construcción, entra la función greedy o miope, que determina el añadido de un elemento a una solución parcial, es si, la función miope consiste en elegir el mejor camino o la mejor opción para un elemento, y luego de que el elemento es añadido a la solución parcial, se recalculan los valores de la función, lo cual hace que este procedimiento sea adaptativo.
 - Pero en realidad no se garantiza una solución óptima y es acá donde entra a trabajar la fase de mejoramiento, es por ello que en la fase anterior se habla de una solución parcial y no final.
 - En esta segunda fase lo que se realiza, es un procedimiento de búsqueda, que a partir de la solución parcial dada, busca una solución mejor.
 - Podemos encontrar algunas implementaciones para el problema de Timetabling aplicando esta metodología en [27] y [32].
- **Algoritmos Genéticos (Genetic Algorithms):** Cabezas [4] hace referencia a Díaz [22] para presentar este método meta-heurístico:
 - Definición (Algoritmo Genético, según Díaz [22]). Un algoritmo Genético es una estructura de control que organiza o dirige un conjunto de transformaciones y operaciones diseñadas para simular los procesos de evolución.

Estos algoritmos están inspirados en la teoría de evolución de Darwin en 1859 (evolución por selección na-

tural), donde los individuos con mas aptitudes para sobrevivir y dejar un mayor número de descendientes, son los más favorecidos (aptos) y transmiten a sus hijos los caracteres favorables de manera hereditaria.

El funcionamiento de un AG (sigla que se utilizara para referirse a Algoritmo Genético), parte de crear un cromosoma o cadena de información, conocida como genotipo, la cual establece la relación entre un conjunto de soluciones de un problema (fenotipo) y el conjunto de individuos de una población inicial.

Varios individuos se agrupan formando una población, aquellos que mejor se adaptan son los que tienen mayor probabilidad sobrevivir y reproducirse

Los nuevos cromosomas se forman seleccionando algunos individuos, utilizando operadores genéticos de cruzamiento y mutación y serán evaluados en cada nueva iteración (generación) mediante una medida de aptitud, originándose así, una nueva descendencia.

4.3. Comparación de técnicas metaheurísticas para resolver el problema del timetabling

De acuerdo a las técnicas meta-heurísticas propuestas anteriormente, se ha diseñado la tabla 1 para mostrar características de cada método expuesto.

Tabla 1. Características de las técnicas meta-heurísticas estudiadas

Meta-Heurística	Características
Grasp	<ul style="list-style-type: none"> • 1989 • Adaptativa de acuerdo a las condiciones del problema • Metodología de las más recientes • Requiere alto tiempo para hallar solución • Búsqueda aleatoria • Dificultad de adecuar los parámetros Soluciones que maneja: 1
Búsqueda Tabú	<ul style="list-style-type: none"> • 1986 • Requiere solución inicial • Más complejo de implementar • Buenos resultados en poco tiempo • No garantiza optimalidad Soluciones que maneja: 1
Recocido simulado	<ul style="list-style-type: none"> • 1983 • Facilidad de implementación • Complejo para problemas muy grandes • No garantiza optimalidad • Facilidad para combinar con otras técnicas heurísticas, para obtener sistemas híbridos • Dependiendo de los parámetros elegidos, las soluciones que se van encontrando pueden ser poco estables. Soluciones que maneja: 1
Colonia de hormigas	<ul style="list-style-type: none"> • 1996 • La más novedosa • Tiempo para encontrar solución de calidad es alto • Ofrece buenas soluciones • No garantiza optimalidad Soluciones que maneja: N, determinadas por la naturaleza del problema.

Meta-Heurística	Características
Algoritmos genéticos	<ul style="list-style-type: none"> • 1975 • Método más antiguo de los estudiados en este artículo • Implementación relativamente simple • No necesitan conocimientos específicos sobre el problema a resolver. • Altamente estudiados • Utilizan operadores probabilísticos (otras técnicas utilizan determinísticos), sin embargo recorren el espacio de soluciones en forma "más inteligentes" que la búsqueda aleatoria. • Mucha información y soporte • Útiles en casos donde no es necesario obtener una solución óptima al problema, sino que una buena solución aproximada sería suficiente. • Utilizan una población de soluciones, siendo menos sensibles a quedar atrapadas en óptimos locales que las técnicas que utilizan una solución única. <p>Soluciones que maneja: N, determinadas por la naturaleza del problema.</p>

Fuente: elaboración propia.

En [33] se realizó un estudio comparativo entre las técnicas meta-heurísticas de Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos y Algoritmos Meméticos, teniendo en cuenta una serie de características como simplicidad, efectividad, adaptabilidad, autonomía, ente otras, a las cuales se les dio un

puntaje entre 0 y 1, siendo 0 el puntaje más bajo y 1 el puntaje más alto. De esta comparación se dedujo que el uso de Algoritmos Genéticos es el más óptimo en comparación con los otros ubicados en este estudio.

En la tabla 2 se muestra de manera resumida el estudio elaborado por [33].

Tabla 2. Calificación de metaheurísticas

Características	Alg. Memét	Alg. Gen	Búsqueda tabú	Rec. Simul.
Simplicidad	0.5	1	0.75	0.75
Independencia	1	1	1	1
Coherencia	0.75	1	0.75	1
Efectividad	1	1	1	0.75
Eficacia	0.75	0.75	0.75	0.5
Eficiencia	1	1	0.75	0.5
Generalidad	0.5	0.75	1	1

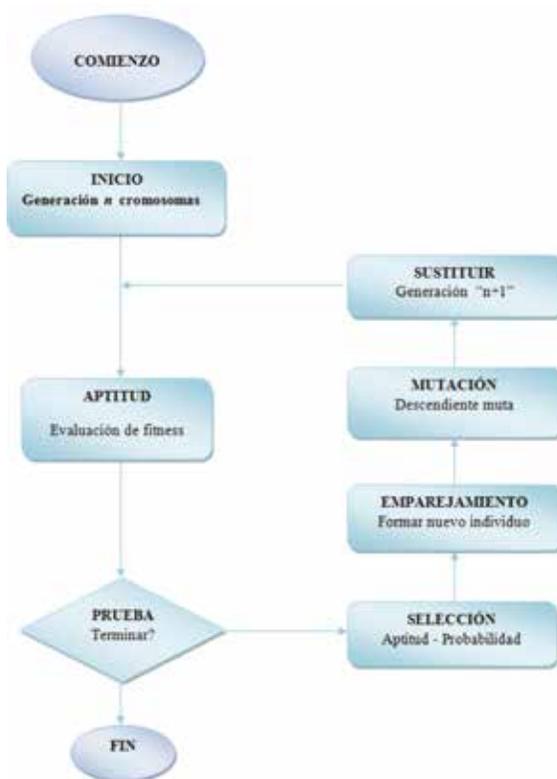
Características	Alg. Memét	Alg. Gen	Búsqueda tabú	Rec. Simul.
Adaptabilidad	1	1	1	0.75
Robustez	0.75	0.75	0.75	0.75
Interactividad	0.75	0.75	0.75	0.75
Diversidad	1	1	0.5	0.5
Autonomía	1	1	1	1
Puntaje Final	10	11	10	9.25

Fuente: elaboración propia.

5. Algoritmos genéticos

En [33], describen el proceso básico de un AG, la codificación del cromosoma, operadores genéticos y parámetros de configuración. El funcionamiento del mismo se muestra en la figura 2.

Figura 2. Funcionamiento de un algoritmo genético



Fuente: elaboración propia.

- I. [Inicio] se genera la población aleatoria de n cromosomas (soluciones posibles para el problema).
- II. [Aptitud] se evalúa la aptitud $f(x)$ de cada cromosoma x de la población.
- III. [Prueba] si la condición de término está satisfecha, se para el algoritmo, se devuelve la mejor solución de la población actual y se va al paso 7.
- IV. [Nueva población] se crea una nueva población repitiendo los siguientes pasos, hasta que se cumpla la condición de parada.

- a) [Selección] se selecciona dos cromosomas padres, de una población, según su aptitud (cuanto mejor es la aptitud, mayor es la probabilidad de ser seleccionado).
- b) [Emparejamiento] con una probabilidad de emparejamiento, los padres se

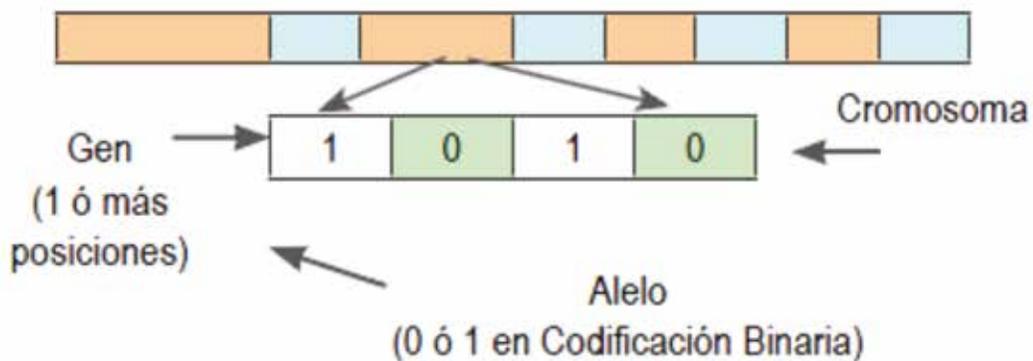
emparejan para formar a un nuevo descendiente (hijos). Si no se realiza emparejamiento alguno, el descendiente es la copia exacta de los padres.

- c) [Mutación] con una probabilidad de mutación, el nuevo descendiente muta (en alguna posición de su cromosoma).
- V. [Sustituir] la nueva población generada es aplicada para otra iteración del algoritmo.
- VI. [Bucle] se va al paso 2.
- VII. Fin del algoritmo.

5.1. Codificación del genotipo

Como se ha señalado los AG están compuestos de una población, que contiene diversos individuos (cromosomas), estos están conformados por un número determinado de genes.

Figura 3. Individuo Binario de un AG



Fuente: elaboración propia.

En la Figura 3, se observa que un gen contiene una información valiosa para solución de problema, esta información debe manejar una codificación, generalmente es binaria, sin embargo dependiendo de la naturaleza del problema se determina cual se implementará.

Naupari y Rosales [33], dan a conocer dos (2) tipos de codificación utilizadas en los algoritmos genéticos:

- • Codificación Indirecta: en este tipo de codificación se halla la binaria, representada por cadenas de "1" y "0"; es la más común, por ventajas de cómputo y de programación.
- • Codificación Directa: en contraste con la anterior, aquí se sitúan las cadenas de números reales, principalmente genes compuestos por números enteros o decimales.

En la codificación directa, también se tienen a las cadenas de letras alfabéticas.

5.2. Operadores genéticos

Consisten en los métodos que se pueden ejecutar sobre una población, son 4:

- **Selección:** proceso que escoge los miembros de la población que serán utilizados en la reproducción (padres). Se eligen los más aptos. Existen varios métodos de selección la más conocida es la rueda de ruleta (Roulette Wheel); en [32, 36] dan a conocer algunos como: elitista, por Estado Estacionario, por torneo, escalada, entre otras.
- Reproducción, Emparejamiento o Cruce (Crossover): Consiste de alguna manera los cromosomas de dos padres, para formar dos descendientes (hijos).
- Algunas variaciones son: cruce de n puntos, uniforme, segmentada, aritmético, etc.[32,36]

- **Mutación:** es encargada de modificar uno o más genes del descendiente, para buscar un factor de diversificación. Se realiza de manera aleatoria, siguiendo a la probabilidad de mutación establecida. Según [33] existen algunas técnicas, algunas como: mutación de bit, de gen, multi-bit, multigen, de intercambio.
- **Reemplazo o Sustitución:** es el método por el cual se insertan los hijos en la población; por ejemplo, mediante la eliminación del individuo más débil o al azar. [33]

5.3. Parámetros de un AG

Estos parámetros deben ser establecidos para cada ejecución que se haga, según [33,37] definen:

5.3.1. Tamaño de la Población

Este parámetro es muy importante determinarlo, dependiendo del problema y las variables a manejar, estamos tratando de la cantidad de individuos que conformara la población en el transcurso de la ejecución. De tal manera, que para un número insuficiente de cromosomas, el AG tiene pocas posibilidades de diversidad, lo que afecta la reproducción y realizará una búsqueda escasa y poco optima. Por otro lado, si la población es excesiva, el algoritmo genético será excesivamente lento [37].

5.3.2. Probabilidad o porcentaje de cruce (crossover)

Determina con qué frecuencia se cruzan los individuos; si es 0% los hijos serán como los padres y solo se afectaran por la mutación. Si este es 100% todos los nuevos individuos son creados mediante reproducción de los padres de la generación previa.

Cuanto más se emparejen los individuos, se supone que los hijos serán mejores; sin embargo, se recomienda por la naturaleza de un AG, que algunos individuos pasen a la siguiente generación sin modificarse [33].

5.3.3. Probabilidad o porcentaje de mutación

Indica la probabilidad en qué deben ser mutados los individuos; si es 0% los descendientes son los mismos que había tras la reproducción. En caso de que haya mutaciones, parte del cromosoma descendiente se modifica; si es de 100%, la totalidad del cromosoma se cambia. [37]. La mutación trata de impedir que la búsqueda caiga en óptimos locales, por eso es conveniente que ocurra de vez en cuando; por el contrario, si ocurre continuamente, se convierte en una búsqueda aleatoria [33].

6. Conclusiones

En conclusión se puede observar que el problema de Timetabling abarca gran variedad de ramas de estudio, incluyendo el caso del SchoolTimetabling, pero así como es bastante estudiado, también posee una gran complejidad, hasta el punto de pertenecer al tipo de problemas NP-Complejos, y es precisamente esto lo que lleva a profundizar en el mismo y dar una solución a este problema.

Teniendo en cuenta todo lo expuesto en este artículo, se puede concluir también que el uso de los Algoritmos Genéticos con sus pros y contras, es la mejor opción para el desarrollo de la solución al problema propuesto de SchoolTimetabling, por su facilidad de uso y la diversidad de soluciones que puede generar.

A pesar de que los Algoritmos Genéticos son un método bastante antiguo, así mismo

existe bastante documentación que soporta el uso de este método, además que gran parte de este soporte está guiado hacia el problema de SchoolTimetabling a resolver. También, como se puede ver en la tabla propuesta por [33] es el método en general más versátil y que puede ofrecer más características positivas para el desarrollo. Así como también ofrece grandes facilidades a la hora de la implementación.

7. Referencias

- [1] Lü, Z. & Hao, J.-K.: Adaptive tabu search for course timetabling. In: *European Journal of Operational Research* 200 (2010), Nr. 1, S. 235–244
- [2] Wren, A.: Scheduling, timetabling and rostering --- A special relationship?. In: Burke, E. & Ross, P. (Hrsg.): *Springer Berlin Heidelberg*. 1153: Practice and Theory of Automated Timetabling., 1996, S. 46-75
- [3] Automated Scheduling, Optimisation and Planning (ASAP) Research Group. http://www.asap.cs.nott.ac.uk/sites/default/files/ASAP_Brochure_20092011.pdf
- [4] Cabezas Garcia, J. X.: Diseño e implementación de una heurística para resolver el problema de calendarización de horarios para universidades, Dissertation (), Escuela Superior Politécnica Del Litoral, 2009
- [5] Schönberger, J.; Mattfeld, D. & Kopfer, H.: Memetic Algorithm timetabling for non-commercial sport leagues. In: *European Journal of Operational Research* 153 (2004), Nr. 1, S. 102 – 116
- [6] Leone, R.; Festa, P. & Marchitto, E.: A Bus Driver Scheduling Problem: a new mathematical model and a GRASP approximate solution. In: *Journal of Heuristics* 17 (2011), S. 441-466

- [7] Barnhart, C.: *Airline Schedule Optimization*: John Wiley & Sons, Ltd. :The Global Airline Industry., 2009, S. 183–211
- [8] Rangel-Valdez, N. & Torres-Jimenez, J.: Solving Employee Timetabling in a Call Center of a Telecommunications Company in Mexico with Simulated Annealing. In: . : Artificial Intelligence, 2009. MICAI 2009. Eighth Mexican International Conference on., 2009, S. 170 -175
- [9] Bai, R.; Burke, E.; Kendall, G.; Li, J. & McCollum, B.: A Hybrid Evolutionary Approach to the Nurse Rostering Problem. In: *Evolutionary Computation*, IEEE Transactions on 14 (2010), Nr. 4, S. 580 -590
- [10] Adamuthe, A. & Bichkar, R.: Tabu search for solving personnel scheduling problem. In: *Communication, Information Computing Technology (ICCICT)*, 2012 International Conference on, 2012, S. 1 -6
- [11] Larrosa J, Meseguer P. Restricciones Blandas: Modelos y Algoritmos. *Inteligencia Artificial*. *Revista Iberoamericana de Inteligencia Artificial* 2003;7 Disponible en: <http://redalyc.uaemex.mx/redalyc/src/inicio/ArtPdfRed.jsp?iCve=92572006>. Consultado el 1 de febrero de 2013.
- [12] Hansen, I.: State-of-the-art of railway operations research. In: *Timetable Planning and Information Quality* (2010), S. 35
- [13] de Werra, D.: An introduction to timetabling. In: *European Journal of Operational Research* 19 (1985), Nr. 2, S. 151–162. Disponible en: <http://www.sciencedirect.com/science/article/pii/0377221785901675>
- [14] Glover, F. & Laguna, M.: *Tabu Search*, Norwell, MA, USA: Kluwer Academic Publishers., 1997 Disponible en: <http://dl.acm.org/citation.cfm?id=549765>
- [15] Burke, E. K.; Kendall, G.; Mısıır, M.; Özcan, E.; Burke, E.; Kendall, G.; Özcan, E. & Mısıır, M.: Applications to timetabling. In: *Handbook of Graph Theory*, chapter 5.6., 2004 Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.188.1458>
- [16] Diaz Fernández, A. & Dowsland, K.: Diseño de heurísticas y fundamentos del recocido simulado. In: *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial* 7 (2003), Nr. 19, S. 93–102. Disponible en: <http://sci2s.ugr.es/docencia/metaheurísticas/Enfriamiento-simulado.pdf>
- [17] Vázquez Espi, M.: Recocido simulado: un nuevo algoritmo para la optimización de estructuras. In: (1994) Disponible en: <http://oa.upm.es/968/>
- [18] Gómez Toro, J. A.; Vanegas Castellanos, J. D. & Zuluaga Gómez, N.: Diseño e implementación de un algoritmo para dar solución al problema de asignación de salones (Timetabling) usando el método de colonia de hormigas. (2009) Disponible en: <http://repositorio.utp.edu.co/dspace/handle/11059/1320>
- [19] Glover, F.: Tabu Search - Part I. In: *ORSA Journal on Computing* 1 (Summer 1989), Nr. 3, S. 190-206 Disponible en: <http://joc.journal.informs.org/content/1/3/190.short>
- [20] Restrepo, G. & Moreno, L.: Modelo para la Asignación de Recursos Académicos en Instituciones Educativas Utilizando Técnicas Metaheurísticas. In: *Avances en Sistemas e Informática* 8 (2012), Nr. 3. Disponible en: <http://digital.unal.edu.co/index.php/avances/article/view/22350>
- [21] Dorigo, M.; Birattari, M. & Stutzle, T.: Ant colony optimization. In: *Computational Intelligence Magazine*, IEEE 1 (Nov.), Nr. 4, S. 28-39. Disponible en:

- http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4129846
- [22] Diaz, A. & Fernández, J. L. G.: Optimización heurística y redes neuronales: Paraninfo., 1996
- [23] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculation by fast computing machines. *Journal of Chemistry Physics*, 21: 1087-1091, 1953.
- [24] ABRAMSON, David; KRISHNAMOORTHY, Mohan; DANG, Henry. Simulated annealing cooling schedules for the school timetabling problem. *Asia Pacific Journal of Operational Research*, 1999, vol. 16, p. 1-22.
Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.994>
- [25] ABRAMSON, David. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, 1991, vol. 37, no 1, p. 98-113. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.302>
- [26] Abel Franco Flores. Estudio comparativo de Algoritmos Genéticos y Algoritmos de Búsqueda Tabú para la resolución del Flow Shop Problem.
- [27] SUÁREZ, Joseph Gallart; MANCHEGO, Fernando Alva; NICHOLSON, Anthony Alma Nole; Gissella Bejarano. Generación Inteligente de Horarios empleando heurísticas GRASP con Búsqueda Tabú para la Pontificia Universidad Católica del Perú. Disponible en: http://www.revistas.pucp.edu.pe/rii/sites/revistas.pucp.edu.pe/rii/files/Joseph_Gallart.pdf
- [28] CARDEMIL, Andrés. Optimización de fixtures deportivos: Estado del arte y un algoritmo tabusearch para el travelingtournamentproblem. Master Thesis, Universidad de Buenos Aires, Departamento de Computación, Buenos Aires, 2002. Disponible en: http://old.dii.uchile.cl/~gduran/docs/tesis/tesis_andres.pdf
- [29] Toro Ocampo, E. M., Tabares Espinosa, P., & Granada Echeverry, M. (2004). Método de colonia de hormigas aplicado a la solución del problema de asignación generalizada. *Revista Tecnura*, 8(15), 66-76. Disponible en: <http://tecnu-ra.udistrital.edu.co/ojs/index.php/revista/article/view/151>
- [30] Feo, T.A.; Resende, M.G.C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, Vol. 8, No. 2, pp. 67-71.
- [31] González, Fernando Pérez. Una metodología de solución basada en la metaheurística Grasp para el problema de diseño de red con incertidumbre. Disponible en: http://pisis.fime.uanl.mx/ftp/pubs/thesis/msc/2006-fernando_perez/tesis-fer-2006.pdf
- [32] Pino, R., Martínez, C., Villanueva, V., Priore, P., & Fernández, I. Application of GRASP methodology to Vehicle Routing Problem (VRP). Disponible en: <http://elrond.informatik.tu-freiberg.de/papers/WorldComp2012/ICA6018.pdf>
- [33] Raúl Esteban Naupari Quiroz, Gisella Kathryn Rosales Gerónimo. Aplicación de algoritmos genéticos para el diseño de un sistema de apoyo a la generación de horarios de clases para la Facultad de Ingeniería de Sistemas e Informática de la UNMSM. Universidad Nacional Mayor de San Marcos. Facultad e Ingeniería de Sistemas e Informática. 2010 Lima - Perú.
- [34] Elvira Mayordomo. NP-completos. Universidad de Zaragoza. Zaragoza - España. Disponible en; <http://webdiis>

- unizar.es/~elvira/mac/npcompletos.pdf
- [35] Bejarano Nicho, Gissella María. Planificación de horarios del personal de cirugía de un hospital del Estado aplicando algoritmos genéticos (Time Tabling Problem). 2011. Disponible en: http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/551/BEJARANO_NICHO_GISSELLA_MAR%C3%8DA_PLANIFICACI%C3%93N_HORARIOS_PERSONAL_CIRUG%C3%8DA.pdf?sequence=1
- [36] Francisco J. Martínez Ruiz, Eduardo García Sánchez, Jaime Muñoz Arteaga, Carlos H. Castañeda Ramírez. Timetabling Académico Usando Algoritmos Genéticos y Programación Celular. Universidad Autónoma de Zacatecas. Departamento de Ingeniería en Computación. México. Disponible en: <http://ingsw.ccbas.uaa.mx/sitio/images/pdfpublicaciones/artiCoNaCi-Co05-20.pdf>
- [37] Arranz de la Peña, J. Parra Truyol, A. Algoritmos Genéticos. Disponible en: <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf>

