

Modelo para la asignación de recursos académicos en instituciones educativas utilizando la técnica metaheurística, búsqueda tabú

Model for academic resource assignment in educational institutions using the tabu search metaheuristics

Gerley E. Restrepo Ing. & Luis Fernando Moreno Velásquez M. Sc.
Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia, Sede Medellín.
grestrep@unal.edu.co; lfmoreno@unal.edu.co

Recibido para revisión 30 de junio de 2011, aceptado 18 de octubre de 2011, versión final 16 de noviembre de 2011

Resumen — La asignación de recursos académicos en instituciones educativas es un problema complejo, especialmente por tener un gran número de restricciones, las cuales son definidas de forma particular por cada institución, y por requerir un alto esfuerzo computacional. El problema consiste en realizar la programación de actividades académicas al tiempo que se asignan recursos a las mismas, como por ejemplo el programar una clase y asignar a ésta su respectivo docente y aula. Este documento presenta aspectos generales del problema, algunos fundamentos relacionados con la optimización combinatoria y las metaheurísticas. Propone además, un modelo para la asignación de recursos académicos utilizando la metaheurística Búsqueda Tabú (Tabu Search). Dicho modelo es de aplicación general, sin embargo, para validar su desempeño se ha empleado en un ambiente de pruebas real para la Universidad Nacional de Colombia, Sede Medellín, obteniendo muy buenos resultados, superiores a los ofrecidos por el método manual que actualmente se ejecuta.

Palabras Clave — Metaheurísticas, Programación de Horarios, Asignación de Recursos Académicos, Asignación de Aulas, Asignación de Docentes, Optimización Combinatoria.

Abstract — The assignment of academic resources in educational institutions is a complex problem, especially for having a large number of restrictions, which are defined in a particular way by each institution, and for requiring a big computational effort. The problem is to perform scheduling of academic activities, while assigning resources to them, such as scheduling a class and assign to it the respective teacher and the classroom. This document presents general aspects of the problem, some basics related to combinatorial optimization and metaheuristics. It also proposes a model for the allocation of academic resources using the metaheuristic Taboo Search. This model is of general application, however, to validate

its performance it has been used in a real test for Universidad Nacional de Colombia, Sede Medellín, achieving very good results, better than those offered by the manual method currently used.

Keywords — Metaheuristics, Timetabling, Academic Scheduling Resources, Classroom Assignment, Teacher Assignment, Combinatorial Optimization.

I. INTRODUCCIÓN

El problema de la asignación de recursos académicos tratado en este artículo, es un problema de optimización combinatoria del tipo NP-Hard [1], lo que significa que los recursos computacionales necesarios para encontrar una solución se incrementan de forma exponencial con la magnitud del problema, especialmente cuando son tratados con algoritmos exactos. Este problema está presidido por un gran número de restricciones, cuya importancia será ponderada por cada institución educativa, pero que en esencia se pueden clasificar en restricciones fuertes (estrictas o duras) y débiles (deseables o suaves). La violación del primer tipo de restricción conduce a rechazar cualquier solución encontrada. Por otro lado, se debe minimizar la violación del segundo tipo de restricción con el fin de obtener soluciones de mayor calidad.

Como objetivo general de investigación, este trabajo busca desarrollar un Modelo para la Asignación de Recursos Académicos (MARA) en una institución educativa, incorporando la metaheurística Búsqueda Tabú, permitiendo la toma de decisiones y la asignación de recursos de una forma

parametrizable, rápida, eficaz, dinámica, eficiente y que beneficie a los diferentes actores involucrados tales como docentes y alumnos.

El presente documento ha sido distribuido de la siguiente forma: En la sección 2, se presenta una perspectiva general de la asignación de recursos académicos y se definen los principales problemas a los que se enfrentan las instituciones educativas en lo referente a dicha asignación. En la sección 3, se ofrecen los conceptos teóricos básicos relacionados con la optimización y se describen diferentes metaheurísticas que podrían ser empleadas en la solución del problema tratado. En la sección 4, se presenta el modelo propuesto para la asignación de recursos académicos en las instituciones educativas, relacionando los problemas de la programación de horarios, la asignación de aulas y la asignación de docentes. La sección 5 presenta la implementación del modelo con el cual se da solución al problema planteado. En la sección 6, se muestran los resultados obtenidos de la aplicación del algoritmo, así como las instancias y características de los problemas tratados. Finalmente, en la sección 7 se presentan las conclusiones finales y se propone además el trabajo futuro a realizar.

II. FUNDAMENTOS DE ASIGNACIÓN DE RECURSOS ACADÉMICOS

El problema de la asignación de recursos académicos en una entidad educativa puede verse como tres sub problemas, que consisten fundamentalmente en la programación óptima, o al menos la mejor posible, de profesores, aulas y cursos, bajo el máximo cumplimiento de ciertas restricciones. Debido a que los modelos tradicionales no ofrecen soluciones satisfactorias en un corto tiempo computacional, han aparecido en la última década y seguirán apareciendo, modelos que incorporan métodos metaheurísticos y otras técnicas de Inteligencia Artificial tales como redes neuronales o algoritmos genéticos, para enfrentar la complejidad en la solución del problema. Las soluciones existentes a este tipo de problemas incluyen métodos exactos, heurísticos y metaheurísticos, tales como Métodos secuenciales, Métodos de agrupación, Métodos de criterios múltiples, Relajación Lagrangiana, Modelos de período simple, Modelos multiperíodo, Modelos multiobjetivo, Métodos basados en razonamiento de casos, Coloramiento de grafos (Graph Coloring) [2], Algoritmos Branch and Bound (BB) [3], Algoritmos basados en Búsqueda Tabú (Tabu Search, TS) [4][5], Algoritmos Genéticos (Genetic Algorithms, GA) [6] [7], Colonia de Hormigas (Ant Colony) [8][9], Enfriamiento Simulado (Simulated Annealing) [10], Programación lógica de restricciones, Redes Neuronales, Programación Lineal, Programación Entera [11].

El problema de la programación de horarios en las instituciones educativas se refiere a la necesidad o deseo de programar los grupos de las respectivas asignaturas en un

periodo académico determinado tal como semestre o año, y durante un ciclo previamente definido, normalmente una semana. Según las políticas de cada institución, puede existir prioridad de la programación de horarios sobre la asignación de aulas. El problema de la programación de aulas consiste en realizar la programación adecuada de los espacios físicos, ya sean aulas, talleres, salas de informática, auditorios o laboratorios, entre otros. Un aspecto fundamental para la correcta asignación de un aula a un grupo, es determinar las características de la primera y los requerimientos del segundo. Idealmente, el problema de la asignación de docentes considera la asignación de éstos como un aspecto ligado a los horarios, asignando un periodo de tiempo para cada grupo y profesor; lo cual significa que no puede crearse más de un grupo para un mismo docente en un mismo horario. Se ha evidenciado que en ocasiones algunas instituciones deciden programar sus horarios y luego conseguir el recurso docente.

III. METAHEURÍSTICAS Y OPTIMIZACIÓN

El término Metaheurística [12], fue introducido por Fred Glover en 1986. La palabra metaheurística combina el prefijo griego ‘meta’, que significa ‘más allá’ y ‘heurística’, que significa ‘encontrar, descubrir, inventar’. Una metaheurística es entonces, un método heurístico que busca resolver problemas computacionales de tipo general, a partir de parámetros dados por el usuario y que generalmente se aplican cuando no se ha hallado una solución satisfactoria a través de una heurística o cuando ésta no es posible de implementar. La mayoría de las metaheurística son usadas para enfrentar los problemas de optimización combinatoria [13], pero no son exclusivas de estos.

La optimización, es el proceso de buscar la mejor solución posible o al menos, la más adecuada para un problema específico. El problema se plantea mediante una ecuación llamada función objetivo, la cual posee unas variables de decisión cuyos valores están sujetos a ciertas restricciones. Lo que se pretende entonces, es encontrar el valor de las variables de decisión que maximicen o minimicen la función objetivo, al tiempo que satisfacen las restricciones planteadas.

Si el problema es de minimización, éste se formula de la siguiente manera:

$$\min\{f(x) \mid x \in X\}$$

Donde f es la función objetivo y x es una solución dentro del espacio X de soluciones factibles del problema.

El problema tratado en este artículo, es un problema de optimización combinatoria. Este tipo de optimización tiene como propósito el encontrar estados dentro de un espacio de búsqueda, que maximicen o minimicen una determinada

función objetivo. Una característica importante de este tipo de problemas es que los estados y su espacio de búsqueda son, comúnmente, particulares para cada problema.

Algunas de las metaheurísticas más comunes empleadas en problemas de optimización son:

Algoritmos Genéticos: El propósito de los Algoritmos Genéticos, propuestos por John H. Holland en 1975 [14] fue el de construir un modelo general de proceso adaptable; desde entonces, esta técnica se ha convertido en una de las áreas de estudio más prometedoras de la Inteligencia Artificial, la Investigación de Operaciones, la Teoría de Algoritmos, la Ingeniería de Software y de la Teoría de la Complejidad Computacional. El funcionamiento básico de un Algoritmo Genético parte del crear un cromosoma o cadena de información, generalmente binaria, conocida como genotipo, la cual establece la relación entre un conjunto de soluciones de un problema, al cual se le conoce como fenotipo, y el conjunto de individuos de una población inicial. Los nuevos cromosomas se formarán utilizando operadores genéticos de cruzamiento y mutación y serán evaluados en cada nueva iteración o generación usando una medida de aptitud, originándose así, una nueva descendencia.

Algoritmos Meméticos: Los Algoritmos Meméticos (Memetic Algorithms) [15] tienen una naturaleza poblacional. Combinan conceptos y estrategias de diferentes metaheurísticas, especialmente de los Algoritmos Genéticos. El término ‘memético’ nace del inglés ‘meme’, palabra acuñada por R. Dawkins para representar el análogo del ‘gen’ utilizado en la teoría evolutiva y llevarlo a la evolución cultural. Los operadores de recombinación de los Algoritmos Meméticos incorporan conocimiento del dominio durante el proceso de generación de la nueva población, a diferencia con los Algoritmos Genéticos que no lo hacen, para evitar sesgar la búsqueda e impedir convergencias a soluciones óptimas. Este conocimiento comúnmente es utilizado para realizar la selección de los atributos, sean de los padres o de otros individuos, que posteriormente serán transmitidos a la nueva población.

Búsqueda Tabú: La Búsqueda Tabú (Tabu Search) [16][17] [18] es un método metaheurístico que emplea un algoritmo determinístico de avance iterativo pero con la posibilidad de aceptar soluciones mejores o peores a la actual, lo cual permite guiar la búsqueda fuera de óptimos locales y a través de nuevos entornos de soluciones. Posee una estructura altamente adaptable a los detalles del problema. Toma de la Inteligencia Artificial el concepto de memoria, permitiendo eludir los mínimos locales así como ejecutar su estrategia de exploración, la cual pretende evitar búsquedas reiterativas en una misma región. La memoria adaptativa es representada a través de una lista tabú, la cual contiene para cada iteración, las mejores soluciones o en su defecto, los movimientos realizados para obtener dicha solución, los cuales no serán contemplados nuevamente en futuras iteraciones, lo que facilita tener un

reducido número de soluciones elegibles; sin embargo, para evitar la pérdida de información al no incluir las soluciones completas, se emplea un ‘criterio de aspiración’ que permite introducir nuevas soluciones aunque estén prohibidas por la lista tabú.

Colonia de Hormigas: Es una de las metaheurísticas más recientemente empleadas para enfrentar problemas de optimización combinatoria y que ha dado resultados relativamente satisfactorios, especialmente en lo relacionado con la eficiencia. La hormiga registra trayectorias aleatorias en búsqueda de su alimento, al hallarlo y regresar a su colonia deposita feromona, la cual permite que otras hormigas sigan el rastro reforzando la intensidad de la feromona y evitando su evaporación, con lo que se define el camino más corto entre su nido y la fuente de alimento. Al momento de definir la metaheurística, el concepto de la evaporación de la feromona es utilizado para evitar que el algoritmo converja a un óptimo local. 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 del entorno de soluciones.

Método de Entorno Variable: A diferencia de la mayoría de las metaheurísticas que trabajan basadas en búsqueda local, como la Búsqueda Tabú, la Búsqueda de Entorno Variable (Variable Neighborhood Search – VNS) [19][20] no trabaja con una única estructura de entorno, sino con un conjunto de estructuras de vecindad. Esta idea fue presentada por N. Mladenovic y P. Hansen (1997), la cual, para evitar quedar atrapada en óptimos locales, realiza cambios sistemáticos de la estructura de vecindades al interior de un procedimiento de búsqueda local.

Recocido Simulado: Dada a conocer por Kirkpatrick [21], el Recocido o Enfriamiento Simulado (Simulated Annealing – SA) más que una metaheurística es una estrategia heurística para problemas de optimización global de funciones con un espacio de búsqueda amplio, la cual requiere de varias decisiones que tienen gran influencia en la calidad de las soluciones generadas y que está inspirada en el proceso del tratamiento térmico del acero, denominado recocido. Si se establece una relación entre el proceso termodinámico del recocido y el de optimización combinatoria se tendría que: los estados del sistema representan las soluciones factibles, la energía simboliza el coste, el cambio de estado representa la solución del entorno, la temperatura simula a los parámetros de control y por último, el estado congelado o de enfriamiento representa la solución heurística. Con características de búsqueda estocástica, el Recocido Simulado permite movimientos de empeoramiento de la solución actual al tiempo que establece criterios para controlar la probabilidad de aceptar transformaciones que no produzcan mejoría en la solución.

En la revisión del estado del arte, llama la atención el sistema presentado por Partovi y Arinze [22]. Dicho sistema está basado en el conocimiento, y trata el problema de la asignación de cursos a una facultad, el cual involucra criterios tales como preferencia de

cursos por facultad, programación de cursos ofrecidos, clases por facultad y número de clases por semana para un docente. Utiliza un procedimiento denominado backtracking (volver hacia atrás), el cual garantiza que al generar soluciones en un nivel determinado, éstas cumplen con las asignaciones realizadas en niveles anteriores. Se destaca de este sistema que es originado por la combinación de la inteligencia artificial y la ciencia de la computación, logrando comportarse como un experto para dar solución a los diferentes problemas; además, puede ser combinado con otros algoritmos para producir un nuevo sistema híbrido. Bradi Masood A. et al [23], presentan un modelo multi-objetivo que busca maximizar las preferencias de una facultad en la asignación de cursos y aulas (horarios). La solución del problema se realiza por medio de una programación del método simplex modificado. Belfares et al [24], formulan el problema de la asignación de recursos como una programación multi-objetivo que explora soluciones adecuadas a partir de métodos no exactos como la metaheurística de Búsqueda Tabú y el concepto de la Optimización de Pareto. Pongcharoen P. et al [25], han desarrollado una herramienta para la programación de cursos mediante Optimización Estocástica. Se han incluido las técnicas de Algoritmos Genéticos (AG), Enfriamiento Simulado (SA) y búsqueda aleatoria. Una conclusión bastante interesante de este trabajo es que identifica que la combinación de AG y SA genera muy buenas opciones de solución, y de manera independiente, SA es levemente mejor que AG pero este último es significativamente más rápido. Este artículo proporciona datos muy importantes tomados de otras fuentes.

Debido a que las metaheurísticas pueden ser no determinísticas, la ejecución repetida del mismo algoritmo para el mismo problema no necesariamente obtendrá el mismo resultado. Esto se ha convertido en uno de los problemas más importantes a la hora de evaluar y comparar la calidad de las metaheurísticas. Dicha comparación debería incluir parámetros que reflejen su eficiencia, eficacia y robustez.

Eficiente. Es decir, que la solución óptima o cercana a la óptima encontrada por la metaheurística, debe darse en un tiempo computacional razonable y con el mejor aprovechamiento de los recursos computacionales.

Eficaz. Es decir, la metaheurística debe proporcionar soluciones óptimas o muy cercanas al óptimo, en especial para un alto porcentaje de los casos reales donde se tendría un parámetro de comparación.

Robusta. El comportamiento de la metaheurística debe ser estable para una gran variedad de casos, es decir, debe ser poco sensible a pequeñas variaciones en el modelo. La probabilidad de obtener una solución lejos del óptimo debe ser baja.

IV. MODELADO DEL PROBLEMA

El modelo a construir, considerará como problema base, el de la programación de horarios, que consiste en definir un

horario para cada uno de los grupos o asignaturas existentes, considerando las diferentes restricciones y la disponibilidad de recursos. Para llegar a este punto, se requiere que las asignaturas hayan sido desglosadas a grupos y los docentes hayan sido relacionados a las asignaturas que dictarán; así, restaría definir el horario y aula de cada uno de los grupos. Para involucrar en el modelo el problema de asignación de aulas, el cual busca la adecuada programación de las mismas según ciertos criterios, se construyen matrices que describan tanto las características de las aulas como las exigencias en éstas por parte de las asignaturas. El problema de la asignación de docentes será involucrado al modelo a partir de matrices que describen las relaciones entre éstos y las asignaturas.

El modelo será estructurado a partir de varios componentes, los cuales buscan darle más control al mismo, mejorando además su eficiencia y eficacia.

Componente Lógico o Algorítmico: Este componente define la metaheurística a emplear por el modelo para la solución del problema e incide directamente en la eficacia del mismo.

Componente de Datos: Este componente define qué información se debe incluir o procesar en el modelo y actúa como un organizador de la misma. Los datos necesarios para alimentar el modelo, están relacionados con Pensum, Carreras, Asignaturas, Grupos, Requisitos, Prerrequisitos, Sedes, Núcleos, Edificios, Aulas, Facultades, Unidades Académicas, Docentes y Estudiantes, entre otros. La estructuración o representación de los datos utilizada por el modelo y que busca disminuir la complejidad del problema, es la siguiente:

Horario académico: La representación de un horario académico u horario de clases, se define como una matriz tridimensional de la forma $H_{nd \times na \times nhd}$, donde nd es el número de días contemplados en el horizonte de planificación, na es el número de aulas existentes y nhd es el número de horas por día a tener en cuenta para la asignación. Según lo anterior, cada asignación de dicha matriz estará definida por una terna $h_{d a h}$, representando el día d , el aula a y a la hora h en la que es programada la dupla asignatura-grupo. Esta matriz es construida en tiempo de ejecución del modelo, y siendo constantemente validada por su componente lógico permite almacenar la mejor solución encontrada para la función objetivo planteada.

Asignaturas: Los datos relacionados con las asignaturas son representados como una matriz de la forma $AS_{nas \times nca}$, donde nas es el número de asignaturas existentes y nca es el número de características a incluir para cada asignatura.

Grupos: Esta matriz, es de la forma $G_{nas \times ng \times nca}$, donde nas es el número de asignaturas existentes, ng es el número de grupos por asignatura y nca es el número de características a incluir para cada grupo.

Grupos-día-hora: Esta estructura temporal de datos, es construida en tiempo de ejecución del modelo y es un paso

fundamental para construir la matriz ‘Horario Académico’; adicionalmente, permite validar algunas restricciones. Los datos están representados de la forma $GDH_{nd \times nhd}$, donde nd es el número de días en los que la dupla asignatura-grupo es programada y nhd es el número de horas por día a programar.

Asignaturas-aulas_sugeridas: Se define una matriz de la forma $ASA_{nas \times na}$, donde nas es el número de asignaturas para las cuales se tiene identificada mínimo un aula en la que es posible programarla y na es el número de aulas sugeridas por asignatura identificada.

Aulas: Los datos relacionados con las aulas son representados como una matriz de la forma $A_{na \times nca}$, donde na es el número de aulas existentes y nca es el número de características a incluir para cada aula.

Aulas-día-hora: Su representación es de la forma $ADH_{nd \times nhd}$, donde nd es el número de días en los que el aula es programada y nhd es el número de horas por día a programar.

Docentes: Los datos relacionados con los docentes son representados como una matriz de la forma $DOC_{ndoc \times nca}$, donde $ndoc$ es el número de docentes existentes y nca es el número de características a incluir para cada docente.

Grupos-docentes: A esta estructura de datos es posible cargar información histórica de la relación docente-asignatura o permitir que el modelo construya una instancia de la misma. Si la información es definida previa a la ejecución del modelo, ofrece como beneficio un aumento considerable en la eficiencia del mismo.

Docentes-día-hora: La representación de esta estructura, es de la forma $DDH_{nd \times nhd}$, donde nd es el número de días en los que el docente es programado y nhd es el número de horas por día a programar.

Docente-asignaturas-perfil: Comúnmente, debido a su propia formación y especialización, el docente tiene un portafolio de asignaturas a impartir. Dependiendo de las políticas de cada Institución, el docente debe ser programado con una o varias de dichas asignaturas.

Componente Matemático: Este componente define la representación matemática adoptada por el modelo,

$$h_{(as.g)dah} = \begin{cases} 1 & \text{si la dupla asignatura.grupo es programada en el día } d, \text{ en el aula } a, \text{ a la hora } h. \\ 0 & \text{en caso contrario.} \end{cases}$$

Función objetivo: Debido a que el problema a tratar es un problema de optimización combinatoria y considerando que cualquier solución del problema es aceptada si y solo si satisface todas las restricciones duras, se debe entonces definir una ecuación que permita determinar qué solución es mejor que otra. Ésto se hará evaluando las restricciones suaves, por lo que la función objetivo será un modelo de costos que permita

principalmente sus variables de decisión, la función objetivo y las limitaciones o restricciones que deberá satisfacer. Un aspecto importante de este componente, es que debe permitir la reclasificación de las restricciones, de forma tal que, una restricción fuerte pueda definirse en cualquier momento como restricción débil y viceversa. Esto permitirá que el modelo se ajuste a las políticas de las instituciones.

Componente Controlador: Este componente es el encargado de tomar los datos y controlar que, tanto la lógica propuesta por el algoritmo como las restricciones definidas, se cumplan. Al igual que el componente algorítmico, el controlador también posee una lógica importante, la cual estará dedicada a transformar la información incluida al modelo en la información final o resultado del mismo.

Componente Informático: Este componente es un apoyo al controlador y básicamente es quien define el tipo y cantidad de recursos de software y hardware que empleará el modelo, lo cual incide directamente en la eficiencia del mismo.

Componente Parametrizable: En este componente se definen todos los parámetros que el modelo permitirá ajustar y que no necesariamente son una constante para las diferentes instancias del problema. La eficiencia del modelo se ve afectada por este componente debido precisamente a que no se tiene un total control de la cantidad y dimensión de cada uno de sus parámetros.

Representación de la Solución: Considerando el problema base y los componentes previamente citados, se define a continuación la representación que tendrá el MARA.

Sea H la matriz que representa un horario académico:

$$H_{nd \times na \times nhd}$$

∴

nd , es el número de días contemplados en el horizonte de planificación.

na , es el número de aulas existentes.

nhd , es el número de horas por día a tener en cuenta para la asignación.

Cada asignación de dicha matriz o variable de decisión, estará representada de forma matemática por:

medir el nivel de satisfacción de las mismas y determinar así la calidad de la solución. Para controlar que se cumpla la restricción, su costo asociado deberá ser igual a cero; de lo contrario, cuanto mayor sea su valor, más lejos estará de la satisfacción. En conclusión, se ha definido el problema como un problema de minimización, donde la función objetivo con menor valor, será considerada la más óptima.

La representación matemática de la Función Objetivo, es:

$$\min f(H) = \sum_{k=1}^r \text{peso}_k * \text{costo}_k$$

∴

k , cantidad de restricciones suaves.

peso_k , peso definido para la restricción k .

costo_k , costo asociado a la insatisfacción o incumplimiento de la restricción k .

Restricciones Fuertes: Las restricciones fuertes, también conocidas como restricciones duras, son aquellas condiciones que, estrictamente y sin excepción, deberá satisfacer el modelo, de lo contrario, la solución será rechazada. A continuación se muestran las restricciones fuertes consideradas para este trabajo.

1. Un grupo solo puede asignarse a una única aula y franja horaria.
2. Dos grupos pueden asignarse a una misma aula en un mismo horario, siempre que dichos grupos estén previamente habilitados para esto.
3. Un aula puede asignarse a dos o más grupos al tiempo (día-hora), solo si su capacidad, y si las características de los grupos, lo permiten
4. Un aula solo debe ser programada en un horario en el que esté disponible.
5. Las horas programadas semanalmente a un grupo (asignatura-grupo) deben ser las requeridas por la asignatura.
6. Un grupo (asignatura-grupo) debe ser programado en un aula con capacidad suficiente para sus estudiantes.
7. Las horas diarias de clase para un grupo deben programarse de forma consecutiva (bloque) y en la misma aula.
8. Un grupo de una asignatura ya sea teórica o práctica debe programarse en un aula propia para esto.
9. Una asignatura solo debe programarse dentro del horizonte de tiempo definido para ello.
10. Un grupo solo puede programarse si tiene asignado un docente.
11. Un docente solo puede asignarse a un único grupo en un único horario.
12. Un docente solo debe ser programado en un horario en el que esté disponible.
13. Un docente solo puede ser programado en las asignaturas de su perfil.
14. El número de horas clase programadas para un docente no

puede superar la dedicación que su categoría define para esto.

Restricciones Débiles: Las restricciones débiles, también conocidas como restricciones suaves, son aquellas condiciones que, deseablemente o en la medida de lo posible, procurará satisfacer el modelo y las cuales serán un criterio para la selección de la solución de mejor calidad. A continuación se muestran las restricciones débiles consideradas para este trabajo.

1. Los grupos para cada asignatura, deben programarse en el horario de preferencia de los docentes.
2. En caso de que un docente tenga en su perfil más de una asignatura a dictar, éstas deberán ser asignadas de acuerdo con su preferencia.
3. Los grupos pertenecientes a un mismo nivel y programa académico, no deben estar programados al mismo tiempo.
4. Los grupos pertenecientes a un mismo nivel y programa académico, deben programarse en una misma jornada del día.
5. Los grupos pertenecientes a un mismo nivel y programa académico, deben exigir el mínimo desplazamiento entre aulas.
6. Un grupo debe ser programado en un aula cuya capacidad sea igual a la cantidad de alumnos que conforman el grupo.

Pesos: Los pesos peso_k , asociados a cada una de las restricciones son manejados como un parámetro para el modelo, lo que permite que cada Institución con base en su experiencia o políticas establecidas, determine su valor y con éste, la prioridad o importancia de las restricciones. Cada uno de estos parámetros tendrá un valor entre 0.0 y 1.0, donde 1.0 representa una alta importancia de la restricción. Por defecto, estos parámetros son definidos en el modelo con los valores de 0.8, 0.8, 0.3, 0.8, 0.8 y 0.3 para las restricciones débiles 1, 2, 3, 4, 5 y 6, respectivamente. Lo anterior significa por ejemplo, en el caso de la primera restricción débil, que es muy importante que la programación de los grupos se haga en el horario de preferencia de los docentes.

V. IMPLEMENTACIÓN DEL MODELO

Selección de la Metaheurística: Debido a que en la literatura no se encontraron instancias idénticas del problema tratado y solucionadas por diferentes estrategias, no existen criterios de comparación detallados para preferir una metaheurística frente a otra. Por lo anterior, este artículo se ha apoyado de forma general en las características o bondades que sobre las metaheurísticas se encuentra en el estado del arte, particularmente en algunas fuentes como J.A. Pacheco [26], quien luego de un estudio comparativo de metaheurísticas para la solución del problema de programación de tareas (el cual también es un problema del

tipo NP-Hard), selecciona a la Búsqueda Tabú como la que ofrece mejor calidad y tiempo de respuesta; y Pongcharoen P. et al [25], quienes exponen que la técnica de Colonia de Hormigas ha ofrecido la mejor solución a este tipo de problemas, seguida por la Búsqueda Tabú y Algoritmos Genéticos.

Teniendo presente que actualmente las instituciones educativas poseen experiencia valiosa en la asignación de los recursos tratados en este trabajo, permitiendo contar entre otras cosas con datos reales y en vista de que Colonia de Hormigas crea una nueva solución sin tomar en cuenta soluciones iniciales o históricas, se ha descartado dicha metaheurística para el modelo, la cual, como trabajo futuro se podría aplicar y tener así una primera comparación de desempeño. Por su parte, los tiempos de respuesta de Algoritmos Genéticos son presentados en la literatura como mucho más lentos que los obtenidos por la Búsqueda Tabú, además de tener una mayor complejidad al requerir un conjunto de soluciones de partida. Por lo anterior, y en vista de la robustez, eficiencia y eficacia que ofrece, se utiliza la Búsqueda Tabú para dar solución al problema de optimización combinatoria, que pretende realizar la programación de un horario académico, relacionando asignaturas, aulas y docentes como los elementos más descriptivos del mismo. Su aplicación, se hará tomando como referencia los componentes y representaciones del modelo, expuestos en la sección 4.

Aplicación de la Metaheurística: La metaheurística Búsqueda Tabú aplica un procedimiento de búsqueda local, impidiendo a través de la penalización de ciertos movimientos, que se caiga en óptimos locales. El punto de partida para la aplicación, es una solución inicial, la cual puede ser obtenida a través de algún algoritmo constructor o utilizando datos de soluciones históricas. La eficiencia de la metaheurística, y en general del MARA, dependen considerablemente de la calidad de la solución inicial.

Para la implementación del modelo, y específicamente para la aplicación de éste en la Universidad Nacional de Colombia, Sede Medellín, se ha desarrollado un algoritmo que construye la solución inicial a partir de los datos correspondientes al segundo semestre del año 2010. La figura 1, ilustra el diagrama de flujo del algoritmo constructor de la solución inicial, el cual busca el cumplimiento de las restricciones duras presentadas en la sección 4, ofreciendo además elementos para obtener una solución de muy buena calidad. Se inicia consultando si existen grupos sin programar; de ser así, se define un criterio de programación (como por ejemplo, iniciar con los grupos de las asignaturas que disponen de una única aula para ser programadas); se buscan aulas habilitadas y disponibles que cumplan las restricciones; luego, se busca un día y hora disponible que cumpla las restricciones y se procede a programar el grupo, aula y docente. En caso de que no exista disponibilidad de recursos según el criterio de programación, se procede a definir un nuevo criterio y se inicia de nuevo la búsqueda. Es importante anotar que los buenos resultados de la

solución inicial, obedecen en gran medida a la estructuración de los datos.

Movimientos: En los problemas de optimización combinatoria, los movimientos son utilizados para definir una nueva vecindad y de esta forma, ir de una solución a otra. Los movimientos a realizar corresponden al intercambio de una posición por otra en la matriz horario, donde una posición representa la programación de un grupo en un aula, día y hora determinado. A continuación se muestra una lista de algunos posibles movimientos, los cuales deben de hacerse garantizando que se cumplan las restricciones duras.

1. Intercambio de dos grupos programados en una misma aula, en el mismo bloque horario, pero en días diferentes.
2. Intercambio de dos grupos programados en diferentes aulas, en el mismo bloque horario y en diferentes días.
3. Movimiento de un grupo al mismo bloque horario libre, en la misma aula, pero en un día diferente.
4. Movimiento de un grupo al mismo bloque horario libre, en diferente aula y en el mismo día.
5. Movimiento de un grupo al mismo bloque horario libre, en diferente aula y en un día diferente.
6. Movimiento de un grupo a un bloque horario libre, en la misma aula y en el mismo día.
7. Intercambio de dos grupos programados en una misma aula, en diferente bloque horario, pero en el mismo día, con igual número de días programados y para cada uno de estos.
8. Intercambio de dos grupos programados en diferentes aulas, en diferente bloque horario, en diferentes días, con igual número de días programados y para cada uno de estos.
9. Movimiento de un grupo a un diferente bloque horario libre, en la misma aula, en el mismo día, para cada uno de los días programados.
10. Movimiento de un grupo a un diferente bloque horario libre, en la misma aula, en diferente día, por cada uno de los días a programar.
11. Movimiento de un grupo a un diferente bloque horario libre, en diferente aula, en el mismo día, para cada uno de los días programados.

Según las pruebas realizadas (ver tablas 2 y 3), se concluye que para beneficio del modelo, es importante definir previamente los tipos de movimiento permitidos, de lo contrario, si se hace uso de la aleatoriedad de intercambios entre grupos, el algoritmo no es eficiente y puede incluso finalizar al agotarse el tiempo estipulado como criterio de parada, sin lograr mejoras considerables o ninguna, en la solución inicial. La figura 2 ilustra algunos de los movimientos aplicados al TS. A partir de la solución inicial H , la metaheurística Búsqueda Tabú tiene a una solución generada por un movimiento, como

su vecino H' , y a todo el conjunto de soluciones generadas por cada posible movimiento, como su vecindario $N(H)$. Según lo anterior, y por el tipo de problema tratado, se hace necesario reducir esta vecindad. El modelo planteado, propone generar un vecindario $N'(H)$ a partir de una muestra mínima de cada movimiento previamente definido, lo cual no siempre generará

una vecindad del mismo tamaño, puesto que a medida que se avanza en la optimización, se agotan los posibles movimientos.

La filosofía de la Búsqueda Tabú permite que la solución del vecindario que tenga el mínimo valor de la función objetivo $f(H)$, sea la nueva solución inicial, sin importar que sea mejor o peor que el valor de la solución actual.

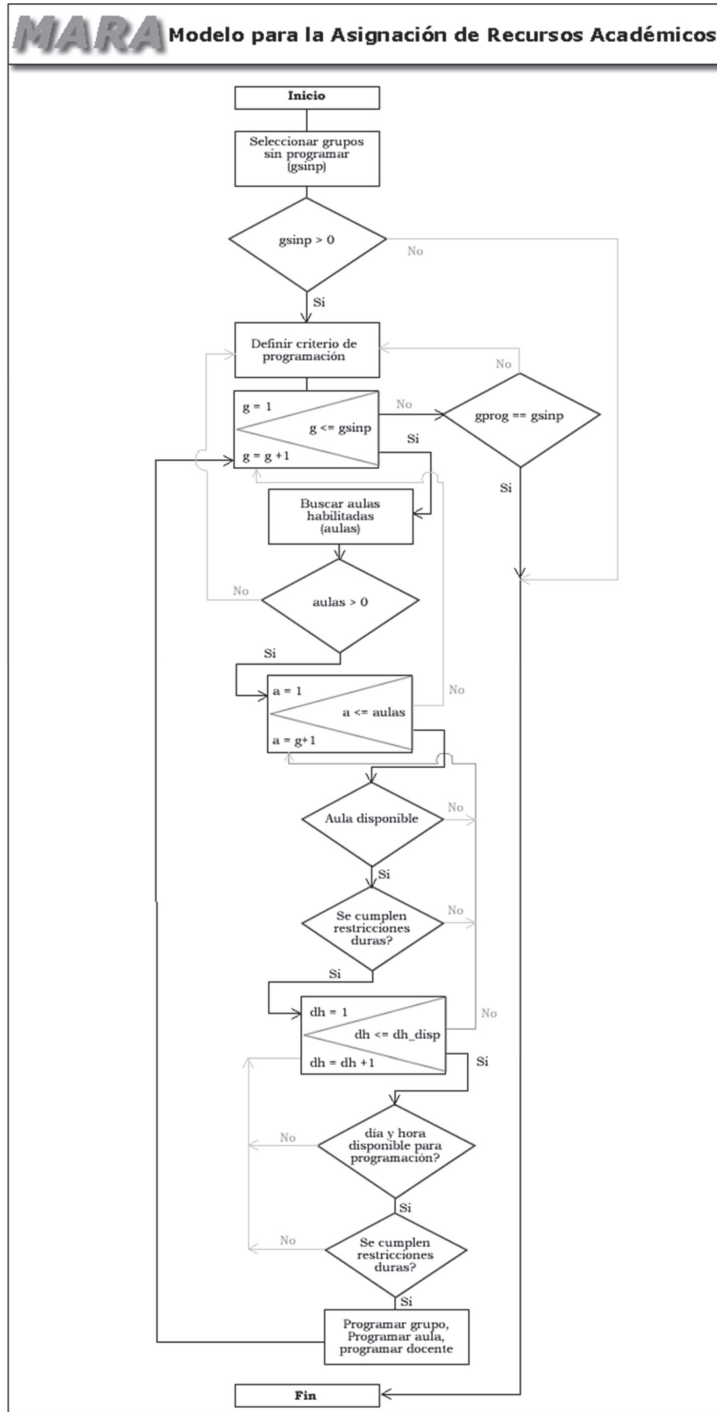


Figura 1. Constructivo solución final


```

Tabu(MejorVecino);//
//Agrega el Mejor Vecino a la matriz Tabú.
Parada();//
//Define los Criterios de Parada a utilizar;
//-----
Repetir hasta (CriterioParada());//
//Ciclo que ejecuta el proceso tantas veces como lo definan los Criterios de Parada.
  Movimientos(SolucionInicial);//
  //Genera los Movimientos, con los cuales se construye el Vecindario para una iteración.
  Vecindario(SolucionInicial,movimientos);//
  //Genera el Vecindario para toda una iteración.
  FuncionObjetivo(Vecindario,Restricciones);//
  //Calcula la Función Objetivo para el Vecindario.
  MatrizIteracion();//
  //Matriz resumen de la iteración. El registro con el mínimo valor de la Función Objetivo se adicionará a la Matriz Tabú.
  MejorVecino(MatrizIteracion,Tabu,CriterioAspiracion,SolucionInicial);//
  //Se identifica al Mejor Vecino de la iteración y se identifica su estado Tabú.
  //Si es el caso, se evalúa el Criterio de Aspiración.
  //Si es el caso, se actualiza la Solución Inicial con la generada por el Mejor Vecino.
  //Se actualiza la Lista Tabú.
  MejorSolucion(MejorVecino,MejorSolucion);//
  //Se evalúa si el mejor vecino de la iteración actual es el mejor de todas las iteraciones realizadas.
  //Si es el caso, se actualiza la Mejor Solución.
  CriterioParada();//
  //Se evalúa el Criterio de Parada.
  //Si es el caso, se finaliza el ciclo.
//-----
ImprimirMejorSolucion();//
//Al finalizar el algoritmo, se imprime la Mejor Solución.
//-----
//-----

```

Figura 3. MARA, Modelo para la Asignación de Recursos Académicos

VI. AMBIENTE DE APLICACIÓN

El MARA desarrollado en este trabajo, fue programado usando el lenguaje PHP Versión 5.3.5, corriendo sobre un servidor Apache Versión 2.2.17 y empleando como motor de bases de datos MySQL Versión 5.5.8. Este ambiente de desarrollo permitió construir un prototipo de aplicación orientada a la Web, cuyo objetivo es el de permitir aplicar el modelo desarrollado al ambiente de pruebas seleccionado. Las pruebas de ejecución del modelo, se realizaron sobre varios equipos tipo PC de gama media con diferentes características. A continuación se muestran algunos datos técnicos de dos de los más representativos:

- Computador Dell Inspiron 530S, con procesador Intel® Core™2 Duo 2.20 GHz, Memoria RAM 3 GB, Sistema Operativo Windows Vista™ Home Basic a 32Bits.
- Computador HP Compaq 8100 Elite CMT, con procesador Intel® Core™ i7 2.93 GHz, Memoria RAM 8 GB, Sistema Operativo Windows 7™ Professional a 64Bits.

El modelo construido se aplicó a la Universidad Nacional de Colombia, Sede Medellín, en donde la programación del horario académico, es realizada de forma prácticamente manual por la Dirección Académica, apoyándose en herramientas ofimáticas tales como hojas de cálculo y con datos recolectados de diversas fuentes, especialmente de las Escuelas adscritas a cada Facultad. Para efectos de pruebas, la Dirección Académica entregó datos

pertenecientes a los semestres 01 y 03 del año 2009 y semestre 02 del año 2010. Una vez organizada y depurada la información suministrada, se tuvo un ambiente de pruebas con los siguientes datos: Aulas: 260, Asignaturas: 871, Grupos: 1.493, Grupos con superposición: 90.

Es importante resaltar que en las fuentes de información consultadas no se encontró una instancia igual o con gran similitud del problema tratado en este trabajo, como tampoco datos detallados de modelos, como para realizar una comparación de resultados. Lo anterior obedece a diferentes motivos, entre los que se encuentra el constante cambio tecnológico; el constante cambio en políticas académico administrativas presentes en las instituciones educativas; y los derechos de autor o trabajos reservados al interior de los grupos de investigación, de quienes es difícil o quizá imposible obtener algo más allá de un paper, excepto que se pertenezca a dicho grupo.

La tabla 2 muestra algunas instancias de prueba utilizadas, cada una de las cuales fue ejecutada en tres ocasiones, obteniendo resultados muy similares. Las instancias 1 a 3, con valores para la Función Objetivo de 203, 215 y 201 respectivamente, ejecutan el modelo utilizando 71 aulas, 34 asignaturas y 238 grupos. Se realizaron 12 iteraciones para cada instancia. Cada iteración arroja un mejor vecino o mejor solución, la cual a su vez genera un nuevo vecindario y luego una nueva iteración. La solución óptima para la instancia 1, fue encontrada en la iteración 3, a los 00:01:33 (hh:mm:ss); para

la instancia 2, dicha solución se encontró en la iteración 4, a los 00:01:33; y para la instancia 3, la solución óptima se halló en la iteración 3, a los 00:01:14. El tiempo total para las 12

iteraciones en las instancias 1, 2 y 3 fue de 00:33:09, 00:33:51 y 00:03:35, respectivamente.

Tabla 2: Resultados para algunas instancias del problema

Resultados del Modelo								
Instancia	Aulas	Asignaturas	Grupos	min $F(H)$	Iteración	Iterac.Totales	t.Mejor Soluc.	t.Total
1	71	34	238	203	3	12	00:01:03	00:03:09
2	71	34	238	215	4	12	00:01:33	00:03:51
3	71	34	238	201	3	12	00:01:14	00:03:35
4	57	169	169	145	1	12	00:00:54	00:04:34
5	57	169	169	151	2	12	00:01:34	00:04:54
6	57	169	169	148	1	12	00:00:59	00:04:39
7	147	323	927	623	5	12	00:02:16	00:05:41
8	147	323	927	589	6	12	00:02:27	00:05:06
9	147	323	927	645	6	12	00:02:56	00:06:31
10	172	537	1158	857	3	12	00:03:15	00:07:07
11	172	537	1158	950	6	12	00:05:18	00:07:26
12	172	537	1158	819	5	12	00:04:12	00:07:03
13	189	705	1327	920	7	12	00:05:40	00:07:58
14	189	705	1327	903	3	12	00:03:45	00:08:17
15	189	705	1327	956	3	12	00:04:16	00:07:35
16	260	871	1493	835	7	12	00:04:17	00:07:50
17	260	871	1493	849	7	12	00:04:59	00:07:59
18	260	871	1493	891	8	12	00:05:02	00:08:25

La tabla 3, resalta para las tres primeras instancias anteriormente descritas, el aumento en la eficiencia y eficacia del modelo al aplicar los criterios previamente definidos para seleccionar el vecindario. Utilizando dichos movimientos, se

aprecia una disminución en el valor de la función objetivo y en el tiempo de ejecución del modelo, comparado con el criterio de aleatoriedad.

Tabla 3: Comparación de resultados según criterio en movimientos aplicados en la metaheurística

Resultados del Modelo								
Aleatoriedad de Intercambios (Vecindario)								
Instancia	Aulas	Asignaturas	Grupos	min $F(H)$	Iteración	Iterac.Totales	t.Mejor Soluc.	t.Total
1	71	34	238	350	8	12	00:04:01	00:04:55
2	71	34	238	421	8	12	00:03:50	00:05:15
3	71	34	238	397	7	12	00:04:46	00:05:39
Aplicando Movimientos Previamente Definidos (Vecindario)								
Instancia	Aulas	Asignaturas	Grupos	min $F(H)$	Iteración	Iterac.Totales	t.Mejor Soluc.	t.Total
1	71	34	238	203	3	12	00:01:03	00:03:09
2	71	34	238	215	4	12	00:01:33	00:03:51
3	71	34	238	201	3	12	00:01:14	00:03:35

VII. CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

El Modelo para la Asignación de Recursos Académicos, MARA, planteado en este artículo y cuyo componente lógico se basa en la metaheurística Búsqueda Tabú, ha sido implementado en un ambiente de pruebas y utilizando datos reales suministrados por la Dirección Académica de la Universidad Nacional de Colombia, Sede Medellín. Este modelo ha tenido un notable mejor desempeño comparado con el método manual que actualmente aplica la institución.

La estructura de datos diseñada y el constructivo para la solución inicial, han garantizado el cumplimiento de todas las restricciones fuertes, permitiendo concentrar esfuerzos en el control de las restricciones suaves las cuales son empleadas para determinar la calidad de la solución.

El esfuerzo, quizá extra, realizado en la estructuración de los datos y en la obtención de la solución inicial, es recompensado por la eficiencia y eficacia del modelo.

El criterio definido en el modelo para seleccionar el vecindario, formado por un representante de cada tipo de movimiento, ofreció buenos resultados, comparado con el criterio de aleatoriedad, especialmente en términos de eficiencia al reducir el tiempo de ejecución del modelo.

El modelo desarrollado incluyó un mayor número de restricciones fuertes y débiles, comparado con las contempladas en el proceso que actualmente ejecuta la Universidad Nacional de Colombia, Sede Medellín; adaptándose así a las políticas y tendencias de las actuales instituciones.

Trabajo Futuro

Adaptar diferentes metaheurísticas al modelo propuesto en esta tesis, especialmente Colonia de Hormigas y Algoritmos

Genéticos. De esta forma se podría aplicar el mismo ambiente de pruebas bajo diferentes componentes lógicos, y realizar posteriormente comparativos en la calidad de las soluciones y/o desempeño de dichos componentes.

Complementar el modelo en lo referente a su horizonte de planificación, de tal forma que además de permitir definir las horas y días de la semana a programar, permita agregar fechas sin clases, tales como días festivos a lo largo de todo el periodo académico. Esto permitiría incluir otro tipo de restricciones, como por ejemplo, controlar la intensidad semestral para una asignatura.

Abordar el problema planteado en esta tesis como un problema de optimización multi-objetivo en donde se plantean dos o más funciones objetivo que se optimizan al tiempo. Por ejemplo, una primera función objetivo podría buscar maximizar las preferencias de los docentes frente a los cursos asignados; una segunda función objetivo buscaría maximizar la preferencia de los mismos docentes frente a los horarios programados; mientras que una tercera función objetivo buscaría maximizar la preferencia de los alumnos frente a los horarios de clase.

Realizar la definición de las restricciones para la asignación de recursos académicos buscando alcanzar un nivel de equilibrio entre la actividad académica y la salud de docentes y alumnos, a quienes se les ve frecuentemente con trastornos en su alimentación, sueño y estabilidad emocional. Para mejorar la calidad de vida de unos y otros, se sugiere por ejemplo conocer la variación de la capacidad laboral y mental de docentes y alumnos para programar las asignaturas más complejas en horarios de alta capacidad y evitar hacerlo en periodos posteriores a un descanso considerable (por ejemplo un lunes en la mañana) o a una fatiga acumulada (por ejemplo un viernes en la tarde).

Desarrollar un software de acceso y ejecución vía Web, que incorpore el modelo creado y que permita incorporar en las restricciones las políticas académico administrativas de las instituciones educativas, constituyéndose así en una herramienta institucional.

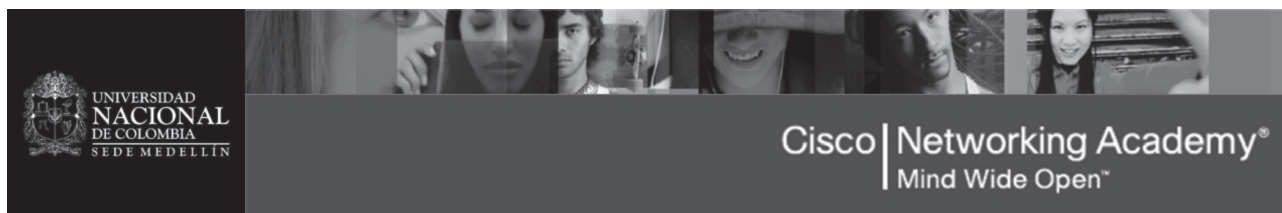
Diseñar indicadores de calidad (tales como excelencia del profesor) e incorporarlos al modelo desarrollado. De esta forma un valor alto en un indicador otorgaría prioridad al docente en una determinada programación (por ejemplo docentes con doctorado tendrían prioridad en las asignaturas de postgrado; o los docentes con baja calificación por parte de los estudiantes serían los últimos a programar). Es importante aclarar que dicha prioridad pertenecería a una restricción débil y que existiría una restricción fuerte que obligue a que a todos los docentes se les asigne un mínimo de horas.

REFERENCIAS

- [1] Garey M. and Johnson, D. S., 1979. Computers and intractability. A guide to the Theory of NP-Completeness. Bell Telephone Laboratories.
- [2] Hertz A.; and Werra, D., 1987. Using Tabu Search Techniques for Graph Coloring, Computing 39, pp. 345 - 351.
- [3] Papadimitriou Ch. H.; and Steiglitz K., 1982. Combinatorial Optimization - Algorithms and Complexity. Prentice Hall, Englewood Cliffs, NJ.
- [4] Hertz A., 1991. Tabú Search for Large Scale Timetabling Problems. European Journal of Operational Research 54, pp. 39-47.
- [5] Alvarez Valdes R.; Crespo Enric; and Tamarit Jose, 2002. Design and implementation of a course scheduling system using tabu search. European Journal of Operational Research, 137, pp. 512-523.
- [6] Wang Y., 2003. Using genetic algorithm methods to solve course scheduling problems. Expert Systems with Applications, 25, pp. 39-50.
- [7] López B. y Johnston J., 2007. Modelo de Asignación de Carga Académica Usando Algoritmos Genéticos. Instituto Tecnológico de Nuevo Laredo, México.
- [8] Socha K.; Sampels M. and Manfrin M., 2003. Ant algorithm for the university course timetabling problem with regard to the state-of-the-art. In Applications of evolutionary computing, Vol. 2611 of LNCS, pp. 334-345.
- [9] Peñuela C.; Franco J. y Toro E., 2008. Colonia de Hormigas Aplicada a la Programación Óptima de Horarios de Clase. Universidad Tecnológica de Pereira. Scientia et Technica Año XIV, No.38.
- [10] Abramson D., 1991. Constructing school timetables using simulated annealing: sequential and parallel algorithms. Management Science, Vol. 37, pp. 98-113.
- [11] Saldaña A.; San Martín C. y Pradenas L., 2007. Modelos de Programación Entera para un Problema de Programación de Horarios para Universidades. Ingeniare, Revista Chilena de Ingeniería. Vol.15 No.3, pp. 245-259.
- [12] Melián B.; Moreno P. J. y Moreno V. J., 2003. Metaheurísticas: una visión global. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 7-28.
- [13] Martí R., 2005. Algoritmos Heurísticos en Optimización Combinatoria. Departamento de Estadística e Investigación Operativa, Universidad de Valencia.
- [14] Holland J. H., 1975. Adaptation in Natural and Artificial Systems. 6th Ed. 2001 MIT Press.
- [15] Burke E. K.; Newall J. P. and Weare R. F., 1996. A memetic algorithm for University exam timetabling. In: Burke and Ross, pp. 241-250.
- [16] Glover F. y Melián B., 2003. Búsqueda Tabú. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 29-48.
- [17] Glover F.; and Laguna M., 1997. Tabu Search. University of Colorado at Boulder.
- [18] Glover F., 1995. Tabu Search Fundamentals and Uses. University of Colorado.
- [19] Hansen P., Mladenovic N. y Moreno J., 2003. Búsqueda de Entorno Variable. Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 77-92.
- [20] Mladenovic N., and Hansen P., 1997. Variable Neighborhood Search. Com. Oper. Res, Vol. 24, pp. 1097 - 1100.
- [21] Kirkpatrick S.; Gelatt C. D. and Vecchi P. M., 1983. Optimization by simulated annealing. Science, 220, pp. 671-680.
- [22] Partovi F. Y.; and Arinze B., 1995. A Knowledge Based Approach to the Faculty-course Assignment Problem. Socio-Econ. Plann. Sci. Vol.29, No. 3.
- [23] Badri M. A.; Davis D. L.; Davis D. F. and Hollingsworth J., 1998.

A Multi-Objetiv Course Scheduling Model: Combinig Faculty Preferences for Courses and Times. Computers Ops Res. Vol 25, N 4, , Elsevier Science Ltd, pp. 303-316.

- [24] Belfares L.; Klibi W., Lo N. and Guitouni A., 2007. Multi-objectives Tabu Search based algorithm for progressive resource allocation. European Journal of Operational Research 177, pp. 1779-1799.
- [25] Pongcharoen P.; Promtet W.; Yenradee P. and Hicks C., 2007. Stochastic optimization timetabling tool for university course scheduling, International Journal of Production Economics.
- [26] Pacheco J. A. y Casado S., 2003. Estudio Comparativo de Diferentes Estrategias Metaheurísticas para la Resolución del Labor Scheduling Problem. Estudios de Economía Aplicada, Vol 21. Número 003, Diciembre, pp. 537-557. Madrid España.
- [27] Molina C., 1995. Higiene de la actividad docente. La Habana. Editorial Pueblo y Educación. 90 p.
- [28] Hertz A. and Robert V., 1998. Constructing a course schedule by solving a series of assignment type problems. European Journal of Operational Research 108, pp. 585-603.
- [29] Wang Y., 2002. An application of Genetic Algorithm Methods for Teacher Assignment Problems. Expert Systems with Applications 22, pp. 295-302.



Cisco Networking Academy es un programa ampliamente conocido de e-doing que enseña a los estudiantes las habilidades tecnológicas de Internet en una economía global. El programa proporciona contenido basado en la Web, pruebas en línea, seguimiento del desempeño de los estudiantes, laboratorios con equipos reales y con simuladores, soporte y entrenamiento por parte de los instructores, así como preparación para las certificaciones estándares de la industria.



Oferta de cursos

- ✓ Mantenimiento de PC: IT Essentials
- ✓ Redes básicas: Cisco Certified Network Associate
- ✓ Redes avanzadas: Cisco Certified Network Professional
- ✓ Seguridad en routers: CCNA Security
- ✓ Voz sobre IP
- ✓ Asterisk básico

Programación 2011

Ciclo 48: Inicia 17 de enero. Finaliza 12 de marzo

Ciclo 49: Inicia 22 de marzo. Finaliza 23 de mayo

Ciclo 50: Inicia 30 de mayo. Finaliza 29 de julio

Ciclo 51: Inicia 8 de agosto. Finaliza 3 de octubre

Ciclo 52: Inicia 10 de octubre. Finaliza 12 de diciembre



Consulte los horarios de cada nivel a través de nuestros canales informativos al pie de página



Además...

- ✓ Alquiler de laboratorios virtuales para auto-estudio o cursos empresariales
- ✓ Presentación de exámenes de certificación para múltiples áreas bajo el Centro Pearson VUE
- ✓ Cursos exclusivos para su empresa
- ✓ Pregunte por nuestros descuentos

CATC - Academia Regional - Academia Local

Universidad Nacional de Colombia sede Medellín

Calle 65 78-28 Bloque M1 Oficina 101. Facultad de Minas

Teléfono: +57 4 4255268 Fax: +57 4 2341002 E-mail: catc@unal.edu.co Web: <http://cnap.unalmed.edu.co>

Facebook: [fb.me/catcunal](https://www.facebook.com/catcunal) Twitter: [@catcunal](https://twitter.com/catcunal) Buzz: [google.com/profiles/catcunal](https://www.google.com/profiles/catcunal)

Medellín, Colombia

