

Aplicación del método de composición musical al problema de asignación de unidades de enseñanza y aprendizaje

Application of the Method of Musical Composition to the teaching and learning units assignment problem

Rafaela Blanca Silva López¹ Rosa Elena Cruz Miguel¹ Eric Alfredo Rincón García¹
Roman Anselmo Mora Gutiérrez¹ Antonin Ponsich¹

Recibido 21 de junio de 2013, aceptado 17 de enero de 2014
Received: June 21, 2013 Accepted: January 17, 2014

RESUMEN

El problema de horarios en instituciones educativas tiene una complejidad computacional que dificulta la obtención de respuestas mediante el uso de métodos exactos, por lo que se han empleado técnicas heurísticas que generan soluciones factibles en tiempos de cómputo aceptables. En este trabajo se presenta el problema de acomodar unidades de enseñanza y aprendizaje en el menor número de trimestres, al tiempo que se satisfacen los criterios establecidos para que los estudiantes puedan cursarlas. La solución de este problema se obtiene mediante la implementación de una heurística poblacional, el Método de Composición Musical. Los resultados obtenidos cumplen con los requerimientos establecidos y actualmente están siendo valorados por las autoridades educativas.

Palabras clave: Asignación de horarios, heurística, método de composición musical, NP-Duro, programación lineal, optimización.

ABSTRACT

Scheduling problems in educational institutions have a computational complexity that makes it difficult to obtain optimal solutions using exact methods. Thus, heuristic techniques have been used to generate feasible solutions in acceptable computation times. In this paper, we solve the problem of assigning teaching and learning units in the least number of quarters using a population heuristic, Method of Musical Composition. The generated solutions satisfy the requirements and show higher quality than manual-built schedules, so that they are currently being evaluated by the educational authorities.

Keywords: Assignment problem, heuristic, method of musical composition, NP-Hard, linear programming, optimization.

INTRODUCCIÓN

El problema de asignación de horarios puede entenderse como la asignación, sujeta a restricciones, de recursos a ciertas tareas disponibles dentro de ventanas de tiempo específicas, de tal forma que se satisfacen, en la medida de lo posible, un conjunto de objetivos deseables [4].

Este tipo de problemas aparece en diferentes actividades diarias, como generar turnos de enfermeras [1], horarios en medios de transporte [5, 10], en asignación de equipos de trabajo [2]. En particular es común en instituciones educativas de nivel superior, y normalmente implican la asignación de profesores, aulas, laboratorios, cursos, exámenes, entre otros, a horarios establecidos [13-14]. Un caso

¹ Universidad Autónoma Metropolitana Azcapotzalco. Departamento de Sistemas. Av. San Pablo 180. Col. Reynosa Tamaulipas. C.P. 02200. México, D.F. E-mail: rbsl@correo.azc.uam.mx; recm@correo.azc.uam.mx; rigaeral@correo.azc.uam.mx; mgra@correo.azc.uam.mx; aspo@correo.azc.uam.mx

especial de este tipo de problemas es conocido como currículo académico, en el que se busca determinar el periodo escolar en el que un estudiante deberá cursar cada una de las materias requeridas para poder completar su licenciatura, de tal forma que se cumplan con restricciones, como seriaciones, y se optimice una función objetivo, como cargas académicas balanceadas durante toda la carrera [7]. Con frecuencia, para resolver el problema resultante se recurre a procesos manuales que requieren gran cantidad de tiempo y que en muchas ocasiones deben repetirse al modificar algún curso. Por lo tanto, diseñar una herramienta capaz de generar soluciones de forma automática y eficiente se ha convertido en un objetivo valioso. Sin embargo, este problema puede verse como un caso particular del problema de empaquetamiento con restricciones de precedencia, del cual hereda la complejidad computacional, que ha demostrado ser NP-Duro [6], por lo que el uso de métodos exactos puede ser inadecuado en instancias de la vida real.

Por lo anterior, el uso de técnicas heurísticas se ha convertido en una herramienta que permite encontrar soluciones de buena calidad en tiempos de cómputo aceptables. De hecho, en los trabajos mencionados anteriormente se emplearon técnicas como una variante de búsqueda armónica, búsqueda local iterada, un algoritmo mimético basado en búsqueda Tabú y apareamiento de abejas melíferas. Cabe mencionar, no obstante, que para una formulación general para esta clase de problemas no se ha identificado un método que supere significativamente a los demás.

En este trabajo se presenta una aplicación del problema de currículo académico en el que se busca ubicar unidades de enseñanza y aprendizaje (UEA) en diferentes trimestres, de tal forma que se satisfacen las condiciones, establecidas por la División de Ciencias Básicas de la Universidad Autónoma Metropolitana unidad Azcapotzalco (UAM-A), para que los estudiantes puedan inscribirse en ellas, al tiempo que se busca minimizar el número de trimestres requeridos para completarlas. Para encontrar soluciones en tiempos de cómputo aceptables se consideró necesario aplicar un método heurístico, ya que cálculos preliminares realizados con un método de resolución exacto comprobaron la incapacidad de este último para determinar una solución factible en un tiempo de cómputo

razonable. En las últimas décadas se ha desarrollado una amplia variedad de técnicas heurísticas [3], que buscan mejorar la calidad de las soluciones encontradas o disminuir los tiempos de cómputo necesarios para generarlas. En este sentido se buscó una técnica de reciente creación para adaptarla al problema de asignación de EE.UU., los resultados obtenidos permitirían evaluar su desempeño en esta clase de problemas, y de ser satisfactorios se podría emplear en instancias de mayor dificultad. Finalmente, se optó por implementar una técnica heurística poblacional conocida como Método de Composición Musical (MMC). La razones por las cuales se seleccionó esta técnica son: a) su desempeño ha sido prometedor en el marco de problemas de optimización continua; b) el MMC es un algoritmo poblacional, por lo que permite la exploración simultánea de varios semiespacios del espacio de búsqueda; c) el MMC es una heurística social, la que presenta el fenómeno de inteligencia de partículas (inteligencia colectiva), lo que le permite intensificar la búsqueda en regiones prometedoras del espacio de búsqueda; d) el MMC presenta tres dinámicas de escalas (personal, local y social), lo que está directamente relacionado con las fases de exploración y diversificación del algoritmo, ya que existe un mecanismo de triple herencia. En [8] se ha discutido las ventajas y desventajas que presentan algunas heurísticas en función del número de dinámicas de escala.

En la siguiente sección se describe el problema de asignación de cursos estudiado en este trabajo incluyendo el modelo de programación matemática propuesto para resolverlo. A continuación se explican las características y modo de operación de la técnica heurística seleccionada para resolver este problema. Luego se presenta el algoritmo desarrollado y se analizan los resultados obtenidos. Finalmente, las conclusiones son presentadas en la sección cinco.

ASIGNACIÓN DE UNIDADES DE ENSEÑANZA Y APRENDIZAJE

Los planes de estudio de la Universidad Autónoma Metropolitana están formados por UEA que deben ser cursadas por los estudiantes en un número preestablecido de trimestres. Cada UEA tiene asociado un número de créditos y en algunos casos requisitos para poder cursarla. Estos requisitos se dividen en dos grupos:

R1) Seriación. La inscripción a algunas UEA está sujeta a la acreditación previa de otras unidades, esto con el fin de que los estudiantes cuenten con los conocimientos básicos necesarios para el correcto aprendizaje de cada UEA.

R2) Mínimo número de créditos. La inscripción a algunas UEA está sujeta a que el estudiante haya obtenido cierto número de créditos por las UEA acreditadas.

Del mismo modo, se establece un número máximo de créditos que puede llevar un estudiante durante cada trimestre.

R3) Máximo 46 créditos en el primer trimestre.

R4) Máximo 60 créditos en los trimestres restantes.

Estos límites se han establecido para evitar que los estudiantes soliciten una carga académica excesiva, lo que podría repercutir negativamente en su desempeño global.

Recientemente se realizaron modificaciones en algunos planes de estudio de la UAM-A, dando como resultado la desaparición y creación de UEA, lo que afectó la seriación y por lo tanto el orden en que deben ser acreditadas las unidades. Estos cambios hicieron necesaria la revisión de los nuevos planes, tanto para establecer una nueva distribución de las UEA en diferentes trimestres como para determinar si los nuevos planes podrían ser completados en el tiempo establecido por la UAM-A. Este trabajo fue realizado manualmente y requirió un esfuerzo de varios días, para cada plan de estudio, el tratar de diseñar una solución capaz de satisfacer las condiciones establecidas **R1-R4**. De hecho, en algunos planes de estudio se observó que no existía una solución factible, ya que el número de trimestres requerido para cursar las UEA excedía lo establecido en el reglamento.

Gran parte de este esfuerzo puede evitarse mediante el uso de un modelo numérico de asignación de tareas, el que ayude a determinar una solución factible, es decir, una solución que permita al estudiante cursar todas las UEA requeridas en el tiempo establecido. Este problema de factibilidad se reformula, en el marco de este estudio, como un problema de optimización que busca el mínimo número de trimestres necesarios para poder cursar todas las UEA, respetando las restricciones **R1-R4**.

El modelo de programación matemática empleado en este trabajo se muestra a continuación.

$$\text{Min } \sum_{i \in U} y_{ij} \quad (1)$$

Sujeto a:

$$\sum_{j=1}^n y_{kj} - \sum_{j=1}^n y_{ij} \leq 0 \quad \forall i \in U, \forall k \in S_i \quad (2)$$

$$\sum_{j=1}^n y_{ij} = 1 \quad \forall i \in U \quad (3)$$

$$\sum_{i=1}^m c_i y_{i(j-1)} - \min c_i y_{ij} \geq 0 \quad \forall i \in U, \forall j \geq 2 \quad (4)$$

$$\sum_{i=1}^m c_i y_{i1} \leq 46 \quad (5)$$

$$\sum_{i=1}^m c_i y_{ij} \leq 60 \quad \forall j \geq 2 \quad (6)$$

$$y_{ij} \in \{0, 1\} \quad (7)$$

$$n, m \in \mathbb{N} \quad (8)$$

Donde

y_{ij} , es una variable binaria que vale 1 si la i -ésima UEA se cursa en el trimestre j , o cero en otro caso. $U = \{1, 2, \dots, m\}$, es el conjunto de UEA que debe cursar un estudiante durante la licenciatura.

S_i , es el conjunto de UEA que deben acreditarse antes de cursar la i -ésima UEA.

c_i , es el número de créditos obtenidos al aprobar la i -ésima UEA.

$\min c_i$, es el mínimo número de créditos necesarios para cursar la i -ésima UEA.

n , es un número de trimestres en el que podrían acomodarse las UEA sin violar las restricciones.

m , es el número de UEA.

De esta forma, la función objetivo, dada por la ecuación (1), busca minimizar el número de trimestres requeridos para que un estudiante curse todas las UEA requeridas en la licenciatura. Las restricciones (2) y (4) garantizan el cumplimiento de seriación y mínimo número de créditos para cursar una UEA (**R1**, **R2**). La restricción (3) asegura que cada UEA será asignada a exactamente un trimestre. Finalmente, las restricciones (5) y (6) evitan que se exceda el máximo número de créditos permitidos

por trimestre (**R3**, **R4**). Por lo tanto, el problema de currículo académico considerado en este artículo requiere que todas las restricciones sean satisfechas para considerar que la solución es factible.

Como se mencionó anteriormente, la complejidad computacional de este tipo de problemas hace que el uso de métodos exactos se restrinja a instancias pequeñas, mientras que instancias medianas o grandes son resueltas, normalmente, mediante técnicas heurísticas. De hecho, el modelo anterior fue implementado en GUSEK, un programa de código abierto para resolver problemas lineales y problemas lineales enteros mixtos [9], pero el tiempo de ejecución para resolver el problema analizado en este artículo superó los 4 días sin que se reportara una solución factible, por lo que se descartó como una opción conveniente. Por otro lado, el personal administrativo de la UAM-A requirió de varias horas de trabajo para obtener una solución adecuada. Por lo anterior, se consideró conveniente la adaptación e implementación de una técnica heurística. En la siguiente sección se describe la técnica empleada en este artículo, el Método de Composición Musical, cuya aplicación en problemas de asignación de tareas no ha sido reportada en la literatura revisada. Los elementos considerados para la elección de esta heurística se mencionaron en la introducción.

DESCRIPCIÓN DE LA TÉCNICA HEURÍSTICA UTILIZADA

Método de composición musical

El Método de Composición Musical (MCM) es una técnica heurística poblacional desarrollada recientemente, inspirada en una analogía entre el proceso de optimización y el proceso creativo de composición musical en un entorno sociocultural [11-12]. La estructura básica del algoritmo se explica en el siguiente párrafo.

Se genera una sociedad artificial de N_c compositores y se define una red de interacción entre ellos. Para cada compositor se crea aleatoriamente un conjunto de N_s soluciones, no necesariamente factibles, que se registran en la memoria asociada al compositor P_i . Posteriormente, los compositores interactúan entre sí analizando la información recibida de los compositores con los cuales tengan un vínculo de acuerdo con la red, y seleccionando los datos que tomarán de ellos. A este conjunto de datos se le

llama ideas socialmente adquiridas, ISA_i . La unión de P_i e ISA_i se denomina matriz de conocimiento del compositor i . Cada compositor genera nuevas soluciones empleando la información almacenada en su matriz de conocimiento y agregando posibles destellos de creatividad. Finalmente cada compositor determina, mediante un criterio glotón, si su nueva creación debe reemplazar alguna de las soluciones contenidas en P_i . Después de cada iteración se puede modificar la red de interacción entre los compositores para promover la distribución del conocimiento. Lo anterior se repite hasta satisfacer con el criterio de paro.

El pseudocódigo de este algoritmo se presenta en la Tabla 1. Para más detalles sobre el diseño de la heurística se refiere al lector a [11-12].

Tabla 1. Pseudocódigo del Método de Composición Musical.

1.	Crear una sociedad artificial con una red de interacción entre ellos
2.	Para cada uno de los compositores hacer
3.	Generar aleatoriamente un conjunto de soluciones
4.	Fin Para
5.	Mientras no se cumpla el criterio de paro hacer
6.	Intercambiar información entre los compositores
7.	Para cada uno de los compositores hacer
8.	Actualizar la matriz de conocimiento
9.	Generar y evaluar una nueva solución
10.	Actualizar el conocimiento
11.	Fin Para
12.	Actualizar los vínculos de la red
13.	Fin Mientras

Implementación para el problema tratado

Para este problema una melodía representa una solución del problema de asignación de tareas. Por ende, una melodía es un vector con 66 entradas, una por cada UEA considerada, y en la entrada i ($i = 1, \dots, 66$) se coloca un número natural que representa el trimestre en el cual se debe cursar la i -ésima UEA. Por otro lado, se genera una sociedad con N_c compositores, donde cada compositor posee N_s melodías, por lo que el algoritmo crea $N_c * N_s$ soluciones. En cada iteración se compondrán N_c melodías nuevas, una por cada compositor, y este agente decidirá si la incluye o no en su repertorio.

A continuación se describe cada una de las etapas del algoritmo propuesto, implementadas en el marco de este problema.

- Al inicio el algoritmo genera para cada compositor una red social, *i.e.* un conjunto de arcos que representa las relaciones entre compositores. En este caso se determinó el uso de una red lineal, en la que el compositor i , solo intercambia información con los compositores $i - 1$ e $i + 1$, con excepción del compositor 1 y el N_c -ésimo compositor, quienes solo tienen un vecino. Esta red social permanece fija durante todo el algoritmo, por lo que se omite el proceso de actualización de vínculos.
- Para cada compositor se crea aleatoriamente un conjunto de soluciones. Un número aleatorio generado uniformemente en el rango $[0,1]$ permite asignar un valor aleatorio a cada variable binaria.
- Cada uno de los compositores compara su peor melodía (la que tiene el peor valor del objetivo) con las de los compositores pertenecientes a su red social. Adquiere aquellas que la mejoran.
- Basándose en la etapa previa de intercambio de información, se actualiza la matriz de conocimiento, más específicamente la submatriz ISA_i asociada a las ideas socialmente adquiridas.
- Para llevar a cabo el proceso de creación de nuevas soluciones cada compositor elige 3 melodías. La primera es la mejor solución de su matriz de conocimientos, S_M , la segunda es elegida al azar entre las soluciones restantes, $\{P_i \cup ISA_i\} - S_M$, y la última es una melodía propia del compositor, S_P . Posteriormente se emplea un procedimiento similar a una cruce de dos puntos para tres padres, como se aplica en el marco de Algoritmos Evolutivos. En este paso, el orden en que los padres participan en la creación de la nueva solución es aleatorio, para favorecer la diversificación en la búsqueda. El costo de la solución creada es evaluado en términos de la función objetivo y el número de restricciones que viola.
- Sobre la base de la etapa previa de generación de una nueva solución, se actualiza la matriz de conocimiento, más específicamente la submatriz P_i asociada a las creaciones personales del compositor, de la siguiente forma: si la solución S_P es mejorada se reemplaza, en caso contrario se desecha la nueva solución.

EXPERIMENTOS NUMÉRICOS Y RESULTADOS

Como se mencionó anteriormente, el problema estudiado consiste en ubicar 66 UEA en el menor número de trimestres posible, al tiempo que se satisfacen las restricciones **R1-R4**, para ello se diseñó un algoritmo basado en MCM.

Para determinar el número de compositores y el número de melodías por compositor se realizaron ajustes progresivos, iniciando con 5 compositores y 3 melodías. Primero se aumentó el número de compositores, sin modificar el número de melodías, hasta que no se obtuvieron mejoras en la convergencia del algoritmo. Posteriormente se aumentó el número de melodías hasta llegar a 9, donde las mejoras obtenidas no justificaban el aumento en los tiempos de cómputo. En las Tablas 2 y 3 se presentan los porcentajes de soluciones factibles y el tiempo de cómputo promedio empleado por el algoritmo, respectivamente. Se hicieron 100 corridas para cada combinación de compositores y melodías considerados.

Finalmente, el algoritmo se diseñó para emplear 15 compositores, cada uno de los cuales posee un conocimiento de 3 soluciones que debe modificar y

Tabla 2. Porcentaje de soluciones factibles de 11 trimestres encontradas por MMC.

		Melodías		
		3	6	9
Compositores	5	84%	81%	66%
	10	98%	89%	76%
	15	99%	94%	87%
	20	94%	98%	96%
	25	95%	99%	99%

Tabla 3. Tiempo promedio en segundos empleado por MMC.

		Melodías		
		3	6	9
Compositores	5	18	39	82,3
	10	25,7	77,2	173,7
	15	31	117,1	187,3
	20	43,4	115,1	188,3
	25	52,8	127	139,9

mejorar mediante interacciones con los compositores dentro de su red social. El algoritmo se detiene al cumplir con 20.000 generaciones, que completa en un tiempo promedio de 31 segundos, y devuelve la mejor solución encontrada.

En la Tabla 4 se presentan las 66 UEA consideradas en este caso. También se incluye el número de créditos asignados a cada una, y las UEA con las que está seriada. En la Tabla 5 se presentan las 7 UEA que están sujetas a un número mínimo de créditos para poder ser cursadas.

Después de 100 corridas del programa diseñado para este problema, se obtuvieron 99 soluciones factibles con 11 trimestres, y una solución factible con 12 trimestres. Por lo tanto, el algoritmo es capaz de generar soluciones que satisfacen los criterios **R1-R4**, requiriendo a lo más de 12 trimestres. Los resultados fueron revisados por las autoridades del Departamento de Sistemas de la UAM-A. En la Tabla 6 se presentan 2 soluciones generadas por el algoritmo, en cada una de ellas se indican las UEA y el total de créditos que deben cursarse cada trimestre. Es importante destacar que el plan de estudios establece 12 trimestres para cursar la carrera.

Al observar las soluciones se pensó en la posibilidad de generar escenarios con 13 trimestres pero con una carga académica menor, de tal forma que el estudiante no tuviera que cursar más de 50 créditos por trimestre. Con esta idea en mente se implementaron las modificaciones apropiadas en el modelo. En esta ocasión el algoritmo se detiene en el momento en que encuentra una solución factible con 13 trimestres, o bien al alcanzar las 20.000 iteraciones. Se realizaron 100 corridas, con una duración promedio de 4,36 segundos, de las cuales se obtuvieron 98 soluciones factibles con 13 trimestres, y 2 soluciones con 13 trimestres que no cumplieran con el criterio de seriación. En la Tabla 7 se presentan 2 de las soluciones generadas con esta modificación. Como se deseaba, la carga académica en cada trimestre es menor, lo que puede favorecer el desempeño académico de los estudiantes sin afectar demasiado el tiempo requerido para terminar la licenciatura.

De esta forma, el algoritmo permite proponer, en tiempos de cómputo satisfactorios, diferentes

Tabla 4. UEA, créditos y seriación.

UEA	Créditos	Seriación	UEA	Créditos	Seriación
1	4		35	9	27
2	9		36	3	29
3	3		37	9	34
4	4		38	9	8; 20
5	3		39	9	22; 32; 33
6	7		40	6	
7	9	1	41	6	
8	6	2	42	12	31; 36
9	3	3	43	8	31; 36
10	9	6	44	9	32
11	6	6	45	7	36
12	9	7	46	9	27; 37
13	3	7	47	9	35; 46
14	6		48	12	46
15	3	12; 13	49	9	22; 30
16	9	11	50	6	
17	7	10; 11	51	6	
18	6	14	52	9	27; 44
19	9	16	53	9	42; 47
20	9	12	54	6	50
21	7	17	55	7	42; 43; 46
22	8	17	56	8	43
23	12	17	57	6	42; 43; 46
24	6		58	8	42; 43
25	6	16; 12	59	3	54
26	9	19	60	8	42; 53
27	9	23; 22	61	6	
28	3	24	62	6	
29	8	17	63	7	55; 56; 57
30	9	17	64	18	59
31	8	22; 26	65	6	54
32	9	17	66	9	24; 43
33	9	21			
34	9	19			

soluciones válidas para que los responsables de tomar las decisiones finales cuenten con una amplia variedad de posibilidades antes de determinar la estructura más adecuada para los estudiantes.

Tabla 5. UEA con requisitos mínimos de créditos.

Trimestre	Créditos requeridos
14	50
40	150
41	150
50	280
59	360
61	150
62	150

Tabla 6. Dos soluciones para 11 trimestres con un máximo de 60 créditos por trimestre.

Propuesta 1		
Trimestre	UEA	Créditos
1	1; 2; 4; 6; 24; 51	36
2	3; 7; 10; 11; 66	36
3	8; 12; 14; 16; 17; 28	40
4	19; 20; 22; 23	38
5	18; 26; 27; 29; 34	41
6	13; 30; 31; 32; 36; 37	41
7	21; 35; 42; 43; 44; 46	54
8	9; 25; 33; 47; 48; 50; 56; 62	59
9	45; 52; 53; 54; 61	37
10	5; 38; 39; 55; 57; 59; 60; 65	51
11	15; 40; 41; 49; 58; 63; 64	57
Propuesta 2		
Trimestre	UEA	Créditos
1	1; 2; 4; 6; 24	30
2	5; 7; 8; 10; 11	33
3	3; 16; 17; 28; 51	28
4	12; 14; 19; 22; 30	41
5	13; 21; 23; 26; 29; 32; 34	57
6	15; 27; 31; 33; 36; 37; 66	50
7	25; 35; 42; 46; 62	42
8	9; 18; 20; 43; 47; 50; 61	47
9	44; 45; 48; 53; 54; 57; 58	57
10	40; 52; 55; 56; 59; 60; 65	47
11	38; 39; 41; 49; 63; 64	58

CONCLUSIONES

En este trabajo se presentó un problema de asignación de tareas, consistente en ubicar un conjunto de UEA en el mínimo número de trimestres, respetando restricciones de seriación, mínimo número de créditos antes de cursar una UEA y máximo número de créditos por trimestre. Por tratarse de un problema NP-Duro, y tras considerar que se resolvería una instancia mediana, se decidió implementar un algoritmo basado en la heurística Método de Composición Musical para generar soluciones del problema planteado.

Tabla 7. Dos soluciones para 13 trimestres con un máximo de 50 créditos.

Propuesta 1		
Trimestre	UEA	Créditos
1	1; 2; 3; 6; 24	29
2	7; 8; 9; 10; 11; 51; 66	48
3	4; 12; 14; 16; 17; 28	38
4	19; 23; 29; 32	38
5	18; 22; 26; 34; 36; 62	41
6	21; 27; 31; 37; 44; 61	48
7	35; 41; 42; 43; 52	44
8	13; 20; 30; 46; 50	36
9	5; 25; 33; 40; 47; 54; 57	45
10	53; 55; 56; 59	27
11	38; 58; 64; 65	41
12	15; 39; 60	20
13	45; 48; 49; 63	35
Propuesta 2		
Trimestre	UEA	Créditos
1	1; 2; 6; 24	26
2	3; 4; 8; 10; 11; 51	34
3	7; 9; 16; 17	28
4	13; 19; 30	21
5	12; 21; 22; 26; 29; 32	50
6	23; 34; 36; 61; 66	39
7	14; 20; 27; 31; 37; 44	50
8	35; 43; 45; 46; 52	42
9	15; 38; 40; 42; 47; 50	45
10	28; 33; 53; 56; 57	35
11	5; 48; 54; 62	27
12	25; 39; 55; 58; 59; 60; 65	47
13	18; 41; 49; 63; 64	46

La instancia presentada en este trabajo está formada por 66 UEA, de las cuales 52 tienen al menos una restricción de seriación, y 7 requieren que se haya cubierto un número mínimo de créditos antes de poder ser cursada. Es importante destacar que al inicio del proceso de investigación se sabía que estas UEA podían asignarse en 12 trimestres, pero se deseaba diseñar un algoritmo más general, capaz de indicar el número mínimo de trimestres requeridos, así como la asignación correspondiente a dicho número de periodos.

En el modelo inicial se estableció que el número máximo de créditos por trimestre sería de 40 para el primer trimestre y de 60 a partir del segundo. Con estas restricciones el algoritmo encontró asignaciones válidas de las UEA que requerían

menos de los 12 trimestres estipulados por el plan de estudios. Sin embargo, las soluciones obtenidas motivaron una modificación de este modelo, y se propuso aumentar a 13 el número de trimestres y disminuir a un máximo de 50 créditos a partir del segundo trimestre.

Con estas modificaciones se obtuvieron soluciones con cargas académicas menores cuya implementación podrá ser evaluada por las autoridades de las UAM-A. En este punto resulta importante destacar que adaptar el algoritmo para cumplir con esta nueva característica requirió de algunos minutos para modificar un par de líneas del código original, mientras que la creación de nuevos planes con el método tradicional podría haber tomado varios días. Asimismo, como se mencionó anteriormente, el desempeño del algoritmo propuesto superó el de una técnica exacta, incapaz de converger en un tiempo de cómputo razonable.

De esta forma, se comprobó la utilidad del modelo y del algoritmo propuesto, para la obtención de asignaciones de las UEA, y para establecer el mínimo número de trimestres necesarios para cursarlas.

REFERENCIAS

- [1] M.A. Awadallah, A.T. Khader, M.A. Al-Betar and A.L. Bolaji. "Global best harmony search with a new pitch adjustment designed for Nurse Rostering". *Journal of King Saud University - Computer and Information Sciences*. Vol. 25, Issue 2, pp. 145-162. 2013.
- [2] D. Barrera, N. Velasco and C.A. Amaya. "A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation". *Computers & Industrial Engineering*. Vol. 63, Issue 4, pp. 802-812. 2012.
- [3] J. Brownlee. "Clever Algorithms: Nature-Inspired Programming Recipes". Lulu Enterprises. Melbourne, Australia. 2011.
- [4] E.K. Burke and S. Petrovic. "Recent research directions in automated timetabling". *European Journal of Operational Research*. Vol. 140, pp. 266-280. 2002.
- [5] V. Cacchiani and P. Toth. "Nominal and robust train timetabling problems". *European Journal of Operational Research*. Vol. 219, Issue 3, pp. 727-737. 2012.
- [6] M. Chiarandini, L. Gaspero, S. Gualandi, and A. Schaerf. "The balanced academic curriculum problem revisited". *Journal of Heuristics*. Vol. 8, Issue 1, pp. 119-148. 2012.
- [7] M. Dell'Amico, J.C. Díaz Díaz and M. Iori. "Bin packing problem with precedence constraints". *Operations Research*. Vol. 60, Issue 6, pp. 1491-1504. 2012.
- [8] C. Gershenson. "Computing networks: a general framework to contrast neural and swarm cognitions". *Paladyn J. Behav Robot*. Vol. 1, Issue 2, pp. 147-153. 2010.
- [9] GUSEK, GLPK Under Scite Extended Kit. Fecha de Consulta: 22 de noviembre de 2013. URL: <http://gusek.sourceforge.net/gusek.html>
- [10] O.J. Ibarra-Rojas and Y.A. Rios-Solis. "Synchronization of bus timetabling". *Transportation Research Part B: Methodological*. Vol. 46, Issue 5, pp. 599-614. 2012.
- [11] R.A. Mora-Gutiérrez, J. Ramírez-Rodríguez, E.A. Rincón-García, A. Ponsich and O. Herrera. "An optimization algorithm inspired by social creativity systems". *Computing*. Vol. 94, Issue 11, pp. 887-914. 2012.
- [12] R.A. Mora-Gutiérrez, J. Ramírez-Rodríguez and E.A. Rincón-García. "An optimization algorithm inspired by musical composition". *Artificial Intelligence Review*. Vol. 3, pp. 1-15. 2012.
- [13] N.R. Sabar, M. Ayob, G. Kendall and R. Qu. "A honey-bee mating optimization algorithm for educational timetabling problems". *European Journal of Operational Research*. Vol. 216, Issue 3, pp. 533-543. 2012.
- [14] A. Salwani and T. Hamza. "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems". *Information Sciences*. Vol. 191, pp. 146-168. 2012.