DISEÑO DE UN ALGORITMO PARA REALIZAR LA PROGRAMACIÓN DE HORARIOS DE LA CARRERA DE INGENIERÍA INDUSTRIAL DE LA PONTIFICIA UNIVERSIDAD JAVERIANA

Propuesta de Investigación

AUTORES:

MANUEL DAVID LOZANO AMÉZQUITA

DIRECTORES:

DAVID BARRERA FERRO

RICARDO FERNANDO OTERO CAICEDO

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA INDUSTRIAL

BOGOTÁ D.C.

Contenido

Índice De Tablas

Índice De Figuras

Glosario de Términos

Capacidad Instalada: atención a la demanda actual y futura por un bien o servicio que una organización puede suplir dada una cantidad de factores productivos disponibles, entendidos estos como la combinación de mano de obra y recursos que interactúan en un periodo específico de tiempo (Orejuela, Manyoma, & González, 2011).

Metaheurística: "Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria en los que las heurísticas no son efectivas. Las Metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos." (Osman & Kelly, 2012)

Jefe de sección: Es la persona encargada de planear, dirigir, coordinar, controlar y evaluar las actividades propias del área a su cargo, teniendo en cuenta los recursos existentes y los procedimientos establecidos para el logro de los objetivos académicos investigativos y administrativos de la Universidad.

Relación de inscripción de materias: arreglo bidimensional que permite determinar la cantidad de estudiantes que inscriben determinada materia en relación con otra el mismo semestre.

Resumen Ejecutivo

Actualmente la mayoría de las universidades enfrentan cada semestre el problema de la programación de horarios y la asignación de salones de clase para cada una de las asignaturas de los diferentes programas académicos que se ofrecen. Esta programación de horarios, desde la perspectiva de la investigación de operaciones, es una caracterización del problema general de timetabling o programación horaria (Hernández & Miranda, 2008). Los problemas de esta área consisten en programar asignaturas en un horizonte de planificación (generalmente una semana) considerando las variables: profesores, días, periodos de clase y un conjunto de condiciones y restricciones, impuestas por la universidad y/o la facultad.

En el caso de la Pontificia Universidad Javeriana, la carrera de ingeniería industrial realiza este proceso de forma manual. Se programan los horarios de cada asignatura teniendo en cuenta la asignación hecha en semestres anteriores, realizando muy pocas variaciones en los horarios y en la asignación de los profesores. Teniendo en cuenta esto, se propone realizar la programación por medio de un algoritmo, que tiene en cuenta las restricciones propias del programa, las asignaturas que usualmente se cursan en el mismo periodo académico, la disponibilidad y horarios de preferencia de los profesores.

Se realiza la simulación del semestre 2015-III y se compara con el real. El resultado del algoritmo genera una solución que disminuye considerablemente la cantidad de cursos asignados en periodos no preferenciales, además de los cruces de horario en materias con incidencia de inscripción, obteniendo una solución mejor a la realmente programada en dicho semestre.

1. Justificación

Las instituciones de educación superior deben decidir, cada periodo académico, sobre la programación de horarios y la asignación de salones invirtiendo mucho tiempo en actividades de planeación (Torres Ovalle, 2013). En este contexto, el problema de asignación de horarios UCTP (*University Course Timetabling Problem*) busca programar, en un horizonte de planificación, las asignaturas ofrecidas considerando restricciones asociadas a los profesores, los días o períodos disponibles, entre otros (Lewis, 2008). En el programa de Ingeniería Industrial, de la Pontificia Universidad Javeriana, los horarios de las asignaturas se programan manualmente. El Director de carrera se basa en los horarios históricos para solicitar, a los diferentes departamentos, la apertura de determinado horario para cada curso.

Semestralmente, deben programarse un total de 235 clases correspondientes a 45 asignaturas ofrecidas por el departamento de Ingeniería Industrial. Posteriormente, los jefes de sección llegan a un acuerdo con cada profesor para determinar el horario en el cual dictará la asignatura. Ahora bien, en los últimos semestres la configuración de la planta profesoral del Departamento de Ingeniería Industrial ha cambiado y ahora se pretende que al menos una clase de cada asignatura del Núcleo de Formación Fundamental sea dictada por un profesor de planta. Sin embargo, los horarios de las asignaturas no han cambiado en función de esta nueva configuración provocando que no se tome en cuenta las preferencias en los horarios de los profesores de planta.

En la literatura científica, se ha resuelto el problema de programación de horarios desde diferentes perspectivas, dependiendo del tipo de institución (colegios o universidades), del tipo de eventos a programar (Clases o Evaluaciones), de las variables utilizadas y las restricciones

establecidas para cada modelo. Por ejemplo, Rojas, Saldaña, & Oliva San Martín (2007) proponen dos modelos de programación lineal entera con el objetivo de minimizar la asignación en periodos no deseados, buscando un balance en la carga de trabajo de los estudiantes. De igual manera, Aladag, Hocaoglu, & Basaran (2009) buscan en su modelo de programación obtener un horario compacto eliminando los tiempos ociosos de los estudiantes. Por otro lado, Daskalaki & Birbas (2005) consideran una función objetivo en su modelo de programación entera en el que representa las preferencias de los profesores en horarios y en determinados salones asignados para la asignatura.

Debido a la complejidad de los modelos, distintos investigadores han utilizado metaheurísticas para obtener soluciones al problema en un tiempo razonable (MirHassani, 2006). Peñuela, Franco, & Toro (2008) utilizan colonia de hormigas para programar horarios de clase definiendo la hora y el salón de clase para cada asignatura. Constantino, Marcondes, & Landa-Silva (2010) utilizan tres enfoques heurísticos cuyo objetivo es maximizar la concentración de estudiantes en el mismo grupo y dentro de la misma área geográfica. A pesar de que muchas investigaciones corresponden a casos de estudio, no existe una técnica general para dar solución al problema que pueda ser utilizada para todas las universidades (Babaei, Karimpour, & Hadidi, 2014). Esto se debe a que las políticas de cada universidad en la asignación de horarios, asignaturas y profesores son diferentes, variando las restricciones del modelo.

Ahora bien, dado que una asignación de horarios sistematizada permite respetar la disponibilidad de horarios de los profesores, disminuir las horas libres entre clases del mismo semestre, entre otros (Schaerf, 1999), este trabajo responde a la pregunta *¿Cómo diseñar una técnica de solución para el problema de programación de horarios de la carrera de ingeniería industrial de la Pontificia Universidad Javeriana?*

2. Objetivos

2.1 Objetivo General

Diseñar una técnica de solución para el problema de programación de horarios de la Carrera de Ingeniería Industrial de la Pontificia Universidad Javeriana.

2.2 Objetivos Específicos

a. Identificar las características, propias del programa de ingeniería industrial de la Pontificia Universidad Javeriana, que deban ser consideradas en su modelamiento.

b. Diseñar un modelo de optimización para el problema de programación de horarios de la Carrera de ingeniería industrial de la Pontificia Universidad Javeriana.

c. Aplicar una técnica de solución para el modelo propuesto.

d. Medir el impacto del algoritmo diseñado comparándolo con el proceso de programación de horarios que actualmente se lleva a cabo en el programa de Ingeniería industrial.

3. Aspectos relacionados con el Diseño

El proceso de diseño estará centrado en la técnica de solución. Al respecto se tendrán en cuenta las siguientes consideraciones:

Requerimientos de diseño (Norton, 2009) . La técnica diseñada cumple los siguientes lineamientos:

a. Usar como información de entrada: i) el número de clases y cursos ofrecidos por el programa, ii) el número de sesiones a la semana de cada asignatura, iii) las preferencias en horario de cada una de ellas y iv) la relación que existe en la inscripción de determinadas materias.
b. Asignar cada materia a una franja horaria en la cual se satisfaga la disponibilidad de horario para los profesores de planta y los profesores de cátedra.
c. Minimizar la violación de las restricciones blandas del modelo, obteniendo un listado de clases asignadas a diferentes horarios.
d. A pesar de ser un problema de diseño y no de control, para este proyecto, el criterio de parada de la técnica de solución tendrá en cuenta el número de iteraciones para dar cumplimiento al alcance y objetivos del proyecto en el tiempo establecido.

Requerimientos de desempeño

a. El algoritmo desarrollado mejora la asignación de horarios de clase *teniendo en cuenta la preferencia de tiempo de los profesores y la relación de inscripción existente entre materias*, el cual se compara con la programación del semestre 2015-III.

Declaración de estándares

Para el desarrollo de este proyecto se usó la norma (TC69, I. S. O. , 2011) que estandariza la aplicación de la metodología DMAIC. Esta metodología hace referencia a un ciclo de mejora continua que se compone de cinco pasos: Definir, Medir, Analizar, Mejorar y Controlar.

4. Antecedentes

Una gran variedad de soluciones a los problemas de programación de horarios educacionales se han propuesto en la literatura. Cada uno de ellos difiere principalmente en la institución involucrada (universidad o colegio) y sus respectivas restricciones (De Causmaecker, Demeester, & Berghe, 2009). Dentro de este contexto se identifican tres diferentes tipos de problemas (Schaerf, 1999): asignación de horarios escolares (*School Course Timetabling*), programación de horarios universitarios (*University Course Timetabling*) y asignación de horarios de exámenes (*Examination timetabling*). Usualmente, en la asignación de horarios escolares los alumnos que pertenecen a un determinado curso toman en bloque las mismas asignaturas, por lo tanto, se desean horarios compactos de clases. En el caso de la asignación de exámenes consiste en asignar el horario a los exámenes, determinando la cantidad de salas y tiempo para realizar cada examen, mientras que, en la programación de horarios universitarios, debe existir cierta flexibilidad en los horarios y en la selección de los cursos que toma cada estudiante, como es el objetivo de este estudio.

Dada la complejidad del problema, las investigaciones continúan centradas en técnicas de solución tratando de mejorar los resultados expuestos por otros equipos en cuanto a calidad y tiempo requerido para encontrar la respuesta, como fue expuesto en diferentes competencias mundiales (ITC, 2007). Sin embargo, aún no se ha encontrado una solución general, así que el

problema sigue sin resolverse por completo (San Martín & Guzmán, 2013). Dos tipos de métodos se han descrito para resolver este problema de asignación: los métodos tradicionales y los métodos no tradicionales.

Los métodos tradicionales según Mejía Caballero & Paternina Arboleda (2009) son aquellos que, debido a su estrategia de búsqueda, recorren todo el espacio de soluciones. Es decir, encuentran todas las soluciones posibles a un determinado problema, sin embargo, el éxito de estas depende directamente de las variables que intervienen en el problema. En este grupo se encuentran herramientas como la programación entera, la programación lineal, entre otras (Torres Ovalle, 2013). Ahora bien, el problema ha sido clasificado como "NP-complejo" en consecuencia no es posible encontrar soluciones óptimas a grandes conjuntos de datos en un tiempo razonable (Hernández & Miranda, 2008) (Chaudhuri, 2010). Es por esto, que se busca por medio de métodos no tradicionales encontrar una solución posible al problema.

De acuerdo a Torres Ovalle (2013), las técnicas no tradicionales no encuentran todas las soluciones a un problema dado que acotan el espacio de búsqueda. Estas técnicas son conocidas como Metaheurísticas y son métodos aproximados que están diseñados para resolver problemas complejos de optimización combinatoria (Gallego, Escobar, & Toro, 2008). Entre los métodos más utilizados en programación de horarios están los algoritmos evolutivos (Ross, Hart, & Corne, 1998) (Ross, Corne, & Fang, Fast practical evolutionary timetabling, 1994), búsqueda Tabú (Di Gaspero & Schaerf, 2001) (Abdullah & Turabieh, 2012), colonia de hormigas (Dowsland & Thompson, 2005), algoritmos voraces (Casey & Thompson, 2003) (Suaréz, Manchego, & Nicho, 2010), redes neuronales (Pichardo, 2011), entre otras. En la Tabla 1 se presenta un cuadro resumen con los artículos consultados.

	Método de Optimización							
	No tradicionales (Metaheurísticas)				Tradicionales	Agentes de la Función Objetivo		
Autores	Búsqueda Tabú	Recocido Simulado	Algoritmo Genético	Colonia de Hormigas	Programación Lineal Entera	Estudiantes	Profesores	Recursos Universitarios
(Pacheco, 2000)			✓					
(López, 2000)	✓							
(Di Gaspero & Schaerf, 2001)	✓							
(Dowsland & Adenso-Díaz, 2003)		✓						
(Dowsland & Thompson, 2005)				✓				
(Daskalaki & Birbas, 2005)					✓	✓		
(MirHassani, 2006)					✓	✓		
(Rojas, Saldaña, & Oliva San Martín, 2007)					✓	✓		
(Hernández & Miranda, 2008)					✓			✓
(Franco, Toro, & Gallego, 2008)	✓							
(Mejía Caballero & Paternina Arboleda, 2009)			✓					✓
(Cortez, Rosales, Naupari, & Vega, 2010)			✓					
(Miranda & Rey, 2012)					✓			✓
(Cifuentes, 2012)					✓	✓		
(Abdullah & Turabieh, 2012)	✓						✓	
(Torres Ovalle, 2013)					✓			✓

Tabla 1 Cuadro Artículos Consultados

En Colombia distintas universidades han utilizado estas técnicas para formular soluciones al problema de asignación. Franco, Toro, & Gallego (2008) Utiliza una búsqueda tabú para la asignación de salones de la Universidad del Norte en Barranquilla, utilizando tres fases de búsqueda. Otros autores como Mejía Caballero & Paternina Arboleda (2009) utilizaron algoritmos evolutivos para resolver el problema de asignación de horarios en la Facultad de Ingeniería de la Universidad de La Guajira. Finalmente, Torres Ovalle (2013) buscó solucionar la asignación de horarios y salones de la Universidad de la Sabana, en el cual, con el objetivo de minimizar la asignación de asignaturas en franjas horarias no deseables, en la Tabla 2 se muestra la contribución de cada uno de estos trabajos, comparado con el proyecto propuesto. En este contexto, este proyecto realiza una asignación sistematizada de horarios de las diferentes asignaturas que ofrece el programa de Ingeniería Industrial de la Pontificia Universidad Javeriana, teniendo en cuenta la preferencia de los profesores para dictar cada asignatura.

Fecha	Universidad	Método	Contribución
2008	Universidad del Norte	No Tradicional. Búsqueda Tabú	Realiza la programación de las asignaturas teniendo en cuenta el número de estudiantes inscritos y la capacidad de los salones. Penaliza las asignaciones en el último horario del día, el cruce de asignaciones para materias que tienen mismo grupo de estudiantes.
2009	Universidad de la Guajira	No Tradicional. Algoritmo Evolutivo	Realiza la prrogramación de las asignaturas teniendo en cuenta la capacidad de los salones, los estudiantes inscritos en cada asignatura.
2012	Universidad Central	Tradicional	Realiza la programación de las asignaturas del porgrama de Ingeniería Industrial de jornada nocturna. Asigna un horario a cada estudiante teniendo en cuenta la ruta crítica del plan de estudios, las prioridades del programa y los prerequisitos cumplidos.
2013	Universidad de la Sábana	Tradicional	Realiza la programación de asignaturas para las facultades de Ingeniería y de Ciencias Económicas y administrativas, teniendo en cuenta una matriz de ponderación para cada franja horaria, penalizando las asignaciones en franjas con un mayor peso.
2013	Universidad Pontificia Bolivariana	Tradicional	Realiza el modelo matemático para la programación de asignaturas penalizando franjas horarias establecidas.
2016	Pontificia Universidad Javeriana	No Tradicional. Búsqueda Tabú	Realizar la programación de las asignaturas de la facultad de Ingeniería Industrial, penalizando la franja horaria de almuerzo, la asignación de materias en el mismo periodo con un alto grado de relación de inscripción y la asignación de materias en franjas en las cuales el docente no tiene preferencia.

*Tabla 2 Contribución Artículos relacionados en Colombia****¡Error! No se encuentra el origen de la referencia.***

5. Metodología

De acuerdo con la declaración de estándares, la metodología de este proyecto está basada en la norma (TC69, I. S. O. , 2011), referente a la aplicación de la metodología DMAIC (Definir, Medir, Analizar Mejorar y Controlar). En la fase de **Definición**, se establece la relevancia y pertinencia del proyecto. Como resultado de un primer análisis, se formula la pregunta de investigación. Dentro de las fases de **Medición y Análisis**, se identifican las características del proceso de programación de horarios de la carrera de Ingeniería industrial **(objetivo específico a)**.

Posteriormente, se realiza la fase de **Mejora** en el que se diseña el modelo matemático y la técnica de solución (Búsqueda Aleatoria) para el problema de programación de los horarios de las distintas asignaturas **(objetivo específico b y c)**. Finalmente, dado el alcance del estudio, el impacto del algoritmo diseñado se evalúa comparando los resultados dados con el proceso de

programación actual, tomando la información del semestre 2015-III y los resultados que ofrece el algoritmo **(objetivo específico d).** La Figura 1 es una representación gráfica de la metodología propuesta, agrupando las actividades en tres fases.

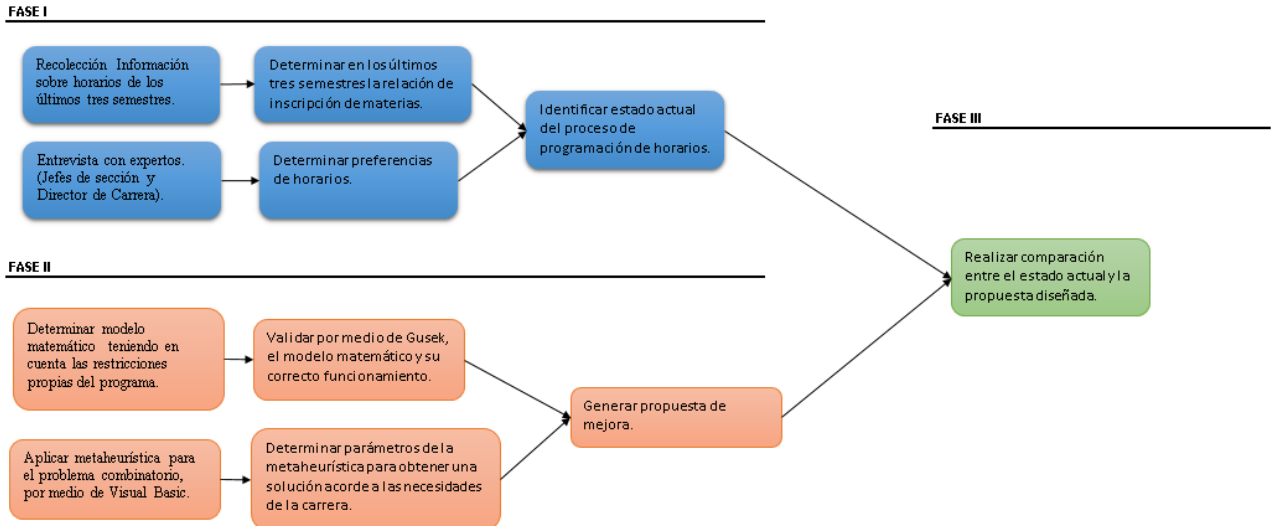

Figura 1. Fases metodológicas del proyecto

Fase I: Objetivo específico a

El programa de Ingeniería industrial tiene una duración de diez semestres que son guiados por una malla curricular o plan de estudios, la cual determina el orden en que los alumnos deben tomar los distintos cursos por cada semestre. Actualmente, la programación horaria es generada por el director del programa, la cual consta de 45 asignaturas para un total de 210 cursos aproximadamente por semestre. La programación para un semestre cualquiera, se genera principalmente sobre la base de la programación horaria utilizada en el semestre anterior. Esta última es actualizada solamente al existir un nuevo requerimiento, al incorporar un curso nuevo en el plan de estudios o al haber cambios en las preferencias horarias de los profesores. En el Anexo 1 se muestra el diagrama de flujo del proceso actual.

Ahora bien, para determinar las preferencias actuales de los diferentes profesores se consultó con los jefes de sección sobre cada una de las materias que tiene a su cargo y así,

establecer de acuerdo a su experiencia, conocer cuál es la preferencia de horarios de los profesores. En el Anexo 2 se encuentra la plantilla que se realizó con cada jefe de sección.

Adicionalmente, al ser un programa académico flexible, no se tiene certeza de cuáles son las materias que los estudiantes inscriben simultáneamente en un mismo semestre. Por esta razón, se decide analizar la información que tiene la carrera sobre la inscripción de cada estudiante de los últimos tres semestres (2015-I, 2015-III y 2016-I) obteniendo la matriz de incidencia que permite conocer el porcentaje de estudiantes que inscriben una asignatura al mismo tiempo que otra (Anexo 3).

Finalmente, las restricciones necesarias para realizar una programación de horarios son categorizadas en dos grupos: Duras y Blandas. Las restricciones duras, deben ser cumplidas obligatoriamente; las restricciones blandas, no son obligatorias y representan condiciones deseables. En el contexto de la Universidad Javeriana se identificaron las siguientes restricciones:

Restricciones Duras

- La programación se debe realizar completa, es decir, la cantidad de periodos asignados para cada clase de cada asignatura debe ser igual a la cantidad programada.
- Cada clase, de cada asignatura, debe tener asignado máximo un tipo de bloque (Bloque I que corresponde a una clase de dos horas consecutivas, Bloque II corresponde a una clase de cuatro horas y dos asignaciones por semana o Bloque III corresponde a una clase de tres horas consecutivas).
- Una clase no puede pertenecer a más de una asignatura.
- En un mismo día, cada clase de cada asignatura puede ser asignada a lo más a una franja horaria.

- Cuando la asignatura corresponde al tipo de Bloque II (debe asignar dos franjas horarias de inicio a la semana) ésta asignación se debe realizar con un día de intermedio y garantizando la misma franja horaria.

Restricciones Blandas

- Se busca no realizar asignaciones de cursos en un determinado periodo, éste es escogido por el usuario.
- Cada curso tiene horarios de preferencia en los cuales se desea asignar.
- Las asignaturas que tienen un alto grado de incidencia de inscripción por semestre deben asignarse en franjas horarias diferentes, es decir, las materias que los estudiantes inscriben conjuntamente por semestre no deben tener el mismo periodo.

Fase II: Objetivos específicos a y b

Formulación de un modelo matemático.

Conjuntos

C $conjunto\ de\ cursos\ a\ programar.$
D $conjunto\ de\ días\ en\ los\ que\ se\ puede\ asignar\ cada\ curso.$
T $conjunto\ de\ periodos\ de\ tiempo\ en\ que\ se\ deben\ programar\ los\ cursos.$
B $conjunto\ de\ bloques\ en\ los\ que\ puede\ pertencer\ una\ materia.$
M $el\ conjunto\ de\ materias\ que\ ofrece\ la\ carrera.$

Parámetros

$Cursos_{cm} = 1\ si\ el\ curso\ c\ pertenece\ a\ la\ materia\ m$

$Matriz_{ij} = 1\ si\ el\ curso\ i\ tiene\ incidencia\ de\ inscripción\ con\ el\ curso\ j$

$Duracion_{bm} = 1\ si\ la\ materia\ m\ tiene\ tipo\ de\ bloque\ b$

$Pref_{mtd} = 1\ si\ la\ materia\ m\ tiene\ preferencia\ en\ la\ franja\ horaria\ t\ y\ el\ día\ d.$

$P_b =$ $total\ de\ horas\ que\ tiene\ cada\ tipo\ de\ bloque\ B.$

Variables Auxiliares

$V_{cdt} =$ $Representa\ la\ cantidad\ de\ cursos\ programados\ en\ Horario\ No\ Preferencial\ del\ curso\ c \in C\ en\ el\ día\ d \in D\ y\ franja\ horaria\ t \in T.$

$U_{cdt} = Representa\ la\ cantidad\ de\ cursos\ programados\ en\ la\ misma\ Franja\ horaria\ de\ un\ curso\ c \in C\ en\ el\ día\ d \in D\ y\ franja\ horaria\ t \in T\ con\ incidencia\ de\ inscripción.$

Variable de Decisión

$$X_{bcdt}\begin{cases}1\ Si\ asigna\ el\ curso\ c\ \in C\ con\ tipo\ de\ bloque\ b \in B\ en\ el\ día\ d\ \in D\ y\ franja\ horaria\ t \in T\\ 0\ no\ realiza\ asignación\ enel\ curso\ c\ \in C\ con\ tipo\ de\ bloque\ b \in B\ en\ el\ día\ d\ \in Dy\ franja\ horaria\ t \in T\end{cases}$$

Función objetivo:

$$min\ z: \quad \alpha * \frac{\sum_{b\in B}\sum_{c\in C}\sum_{d\in C} X_{bcd7}}{\sum_{c\in C}\sum_{m\in M} Cursos_{cm}} + \beta * \frac{\sum_{t\in T}\sum_{c\in C}\sum_{d\in C} V_{cdt}}{\sum_{c\in C}\sum_{l\in C} Matriz_{lc}} + \theta * \frac{\sum_{t\in T}\sum_{c\in C}\sum_{d\in C} U_{cdt}}{\sum_{c\in C}\sum_{m\in M} Cursos_{cm}} \qquad (1)$$

Sujeto a:

$$\sum_{d\in D}\sum_{t\in T} X_{bcdt} = \sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} \qquad \forall\ c \in C\ , \forall\ b \in B \qquad (2)$$

$$\sum_{m\in M} Pref_{mtd} * Cursos_{cm} \geq \sum_{b\in B} X_{bcdt} - U_{cdt} \qquad \forall\ d \in D\ , \forall\ c \in C\ , \forall\ t \in T \qquad (3)$$

$$\sum_{b\in B} X_{bc1t} \leq \sum_{b\in B} X_{bc3t} \qquad \forall\ c \in C\ , \forall\ t \in T\ , tal\ que: \sum_{b\in B}\sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} = 2 \qquad (4)$$

$$\sum_{b\in B} X_{bc2t} \leq \sum_{b\in B} X_{bc4t} \qquad \forall\ c \in C\ , \forall\ t \in T\ , tal\ que: \sum_{b\in B}\sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} = 2 \qquad (5)$$

$$\sum_{b\in B} X_{bc3t} - \sum_{b\in B} X_{bc1t} \leq \sum_{b\in B} X_{bc5t} \quad \forall\ c \in C\ , \forall\ t \in T\ | \sum_{b\in B}\sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} = 2 \qquad (6)$$

$$\sum_{t\in T}\sum_{b\in B} X_{bc4t} \leq \left(1 - \sum_{t\in T}\sum_{b\in B} X_{bc5t}\right) \qquad \forall\ c \in C\ | \quad \sum_{b\in B}\sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} = 2 \qquad (7)$$

$$\sum_{t\in T}\sum_{b\in B} X_{bcdt} \leq 1 \qquad \forall\ c \in C\ , \forall\ d \in D\ | \quad \sum_{b\in B}\sum_{m\in M} P_b * Duración_{bm} * Cursos_{cm} = 2 \qquad (8)$$

$$\sum_{b\in B} X_{bcdt} + \sum_{b\in B} Matriz_{lc} * X_{bldt} - V_{cdt} \leq 1 \qquad \forall\, c \in C\,, \forall\, l \in C\,, \forall\, d \in D\,, \forall\, t \in T \tag{9}$$

La ecuación (1) expresa la función objetivo, la cual busca minimizar la suma de todas las posibles penalizaciones, en el primer argumento el periodo "7" corresponde a hora de almuerzo (1:00 PM). La ecuación (2) garantiza la cantidad de asignaciones de cada curso en la semana. La ecuación (3) busca asignar cada curso en un horario preferencial, de lo contrario aumenta la variable auxiliar U_{cdt}. Los conjuntos de ecuaciones (4), (5) y (6) garantizan que cuando la clase tiene dos sesiones por semana, realice la programación con un día de intermedio y en la misma franja horaria, por lo tanto, si asigna una sesión el lunes programe la otra sesión el miércoles en el mismo periodo. La ecuación (7) impone la restricción que materias con dos sesiones por semana no se programen jueves y viernes al mismo tiempo. La restricción (8) garantiza que cada clase con dos sesiones por semana se programen en diferentes días y por último, la restricción (9) busca programar las clases con un alto grado de incidencia en diferentes franjas horarias, de lo contrario aumenta la variable auxiliar V_{cdt}.

La formulación fue implementada en GLPK (GNU Linear Programming Kit) por medio de la interfaz GUSEK (GLPK Under Scite Extended Kit). Para resolver el caso bajo estudio, se utilizó un computador personal con Procesador Intel(R) Core(TM) i7 con memoria RAM de 4GB. Se corrieron un total de 25 instancias, cada tres instancias se cambió el tamaño de cursos a programar. Para instancias de gran tamaño se establecieron tiempos límites de 3600 y 10.000 segundos respectivamente, con el fin de analizar la variación en la Función objetivo y el GAP con respecto al modelo relajado. Finalmente, se definieron los pesos función objetivo $\left(\alpha, \beta, \theta = \frac{1}{3}\right)$ para así, obtener una función objetivo donde todos los argumentos tienen la misma prioridad.

Diseño de una técnica de solución. Al ser un problema que modifica sólo una variable (la franja horaria de cada asignatura) es posible utilizar una técnica de solución basada en trayectoria. Se selecciona la búsqueda aleatoria utilizando operadores de la metaheurística búsqueda tabú, ya que pueden proporcionar soluciones de mejor calidad en problemas de programación de tareas (Bonrostro & Yusta, 2003) (Pongcharoen, Promtet, Yenradee, & Hicks, 2008). Por lo tanto, se utiliza una lista de candidatos aleatoria, una lista tabú, un criterio de aspiración y una lista de frecuencia para dirigir la búsqueda a caminos no realizados.

La programación de la metaheurística se realiza en Visual Basic obteniendo la información de entrada en un archivo de Excel para luego procesarla y usarla como parámetro de entrada en la Búsqueda Aleatoria.

Construcción de una solución inicial

La configuración inicial partirá de la información de entrada dada por el usuario para luego asignar el mayor número de cursos posibles en horarios preferentes siempre y cuando las demás clases con incidencia no estén en la misma franja horaria. En la **¡Error! No se encuentra el origen de la referencia.** se muestra el diagrama para la construcción de la solución inicial.

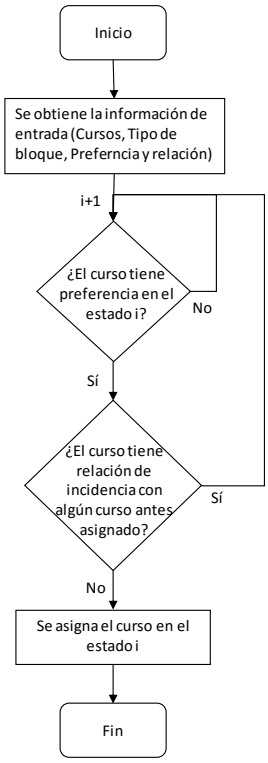

Figura 2. Diagrama Configuración de la Solución Inicial

En la segunda etapa de la búsqueda genera la lista de candidatos con un tamaño determinado por el usuario. Cada candidato se reemplaza en la iteración anterior y se evalúa la nueva función objetivo, determinando si el cambio producido disminuye la función objetivo con respecto a la obtenida en la iteración anterior.

Se escoge entonces el candidato que más disminuya la función objetivo actual. El movimiento realizado se guarda en una Lista Tabú, que almacena los movimientos realizados al tiempo que les asigna un estado de prohibición durante un número determinado de iteraciones en el cual no se asignará este movimiento. Este número periodo Tabú es:

$$Periodo\ Tab\acute{u} = \mathrm{Ln}(Periodos * Días * Cursos) \quad (10)$$

De la misma manera se guarda en otra lista las veces que se ha realizado un movimiento para busca motivar cambios que no se hayan realizado. Para penalizar estas asignaciones que ya

se han hecho, se aumenta en el siguiente factor la función objetivo que genera el cambio de este candidato:

$$FO = FO * (1 + \frac{Cantidad\ de\ veces\ asignado}{Cantidad\ de\ iteraciones\ hasta\ el\ momento}) \quad (11)$$

Debido a que existe la posibilidad de que la mejor solución tenga un estado Tabú al tiempo que el valor de la función objetivo mejora la solución actual, la metaheurística permite quitar el estado de prohibición cuando se cumple el criterio de aspiración. El criterio de aspiración para esta programación sucede cuando los últimos tres movimientos consecutivos se han cambiado cursos con tipo de bloque II, es decir, cursos que se deben programar dos veces a la semana, lo cual significa que se han realizado tres cambios en cursos de dos sesiones, siendo un cambio importante dentro de la búsqueda que generaría el escape de un óptimo local.

Teniendo claro los criterios que se debe tener para realizar la metaheurística, a continuación, se muestra el diagrama de la Búsqueda Aleatoria para la programación de horarios de la facultad de Ingeniería Industrial:

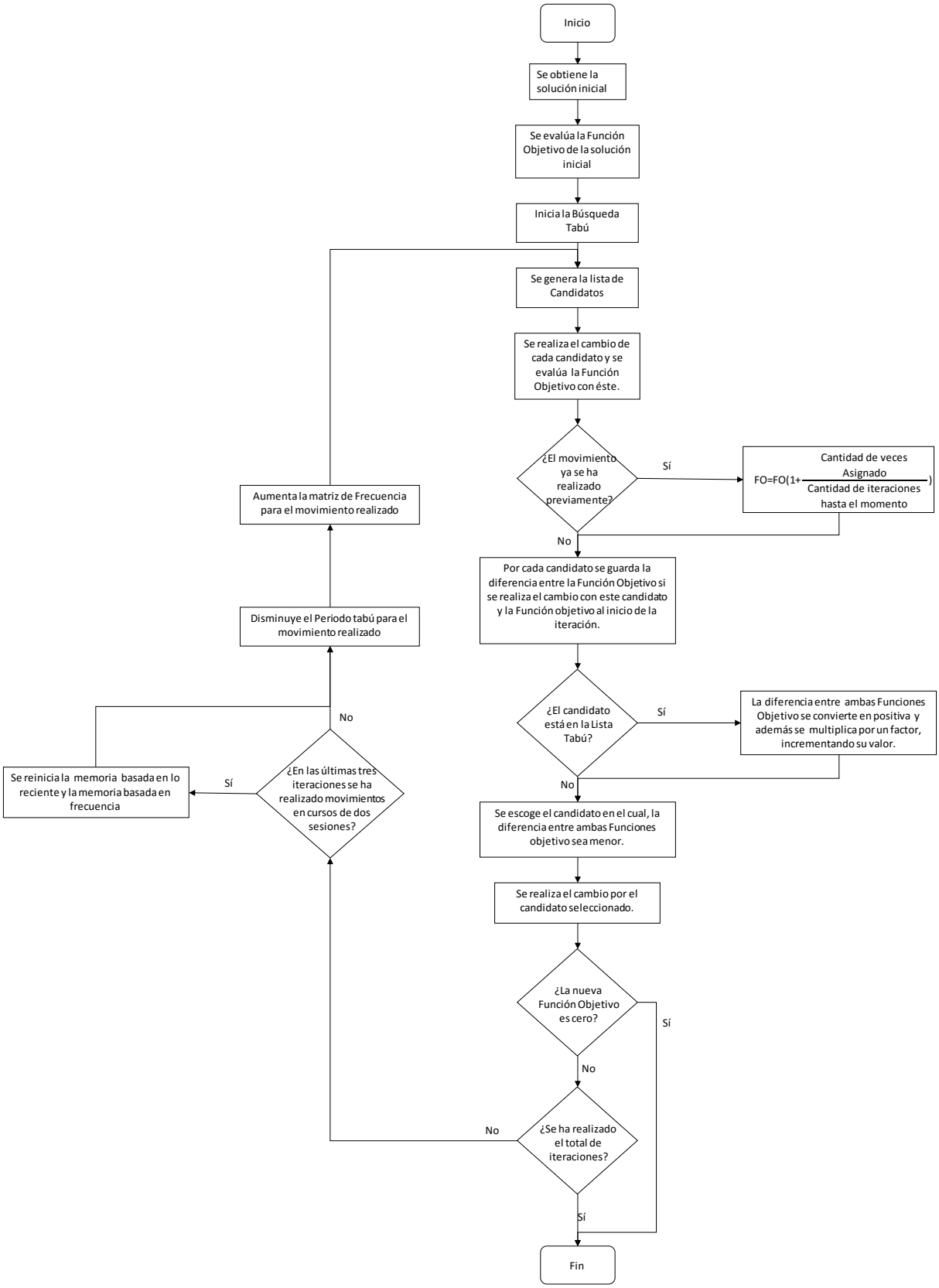

Figura 3. Diagrama configuración Búsqueda Aleatoria

Para determinar la combinación de factores de la metaheurística que logra una función objetivo menor se realizó un Diseño de Experimentos factorial 4^2, teniendo como variable respuesta la función objetivo y los siguientes factores con sus respectivos niveles: tamaño lista de Candidatos (5, 10, 20, 40) y Número de Iteraciones (100, 400, 700, 1000).

Debido a que no se cumple el supuesto de que los grupos deben proceder de poblaciones con varianzas iguales y además que la muestra no cumple el supuesto de normalidad, se realizaron pruebas no paramétricas. La prueba T2 de TamHane, la cual está basada en la prueba T y realiza comparaciones múltiples sin necesidad de estos supuestos, permite concluir que estos factores son significativos y que la mejor combinación de parámetros es 1000 iteraciones y 40 candidatos, como también se muestra en el gráfico de perfil (Figura 4).

Ahora bien, analizando las combinaciones de estos parámetros y el costo computacional asociado a cada uno de ellos, se decide escoger la combinación 1000 iteraciones y 20 candidatos, dado que según la Figura 4 esta combinación de parámetros no empeora significativamente la función objetivo y sí representa una mejora en el tiempo computacional.

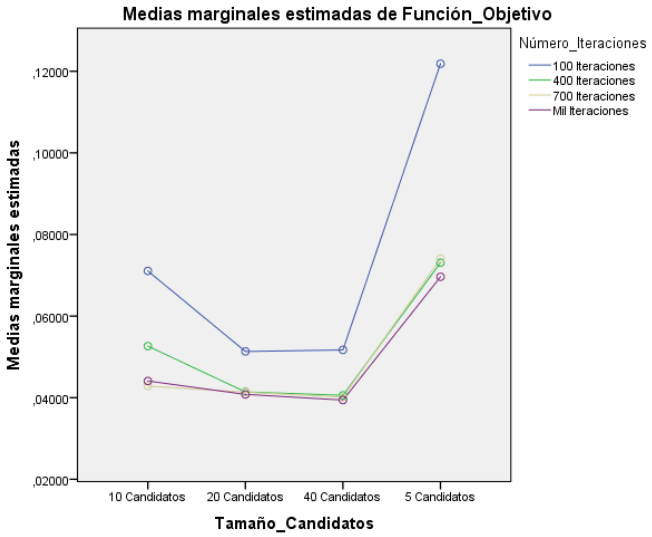

Figura 4. Gráfico de Perfil- Parámetros Búsqueda Aleatoria.

6. Resultados

Se corrieron un total de 25 instancias para ambos modelos. Los resultados del modelo de programación lineal están divididos en tres grupos según el tamaño de la instancia y la solución obtenida.

El primer grupo de instancias corresponden a una cantidad de cursos pequeña (1 a 40 cursos) comparada con el problema real, además el modelo obtiene la solución óptima para cada prueba. En la Tabla 3 se muestran los resultados para estas instancias.

	Parámetros de Entrada					**Modelo Matemático**	
Instancia	Periodos	# Incidencias	# de Cursos	# de Cursos 1 sesión	# de Cursos 2 Sesiones	FO	Tiempo (s)
1	3	0	1	1	0	0	0.22
2	3	0	1	0	1	0	0.217
3	3	6	3	0	3	0.11000	0.328
4	3	6	3	0	3	0.11000	0.417
5	4	20	5	3	2	0.03300	0.23
6	4	20	5	0	5	0.09900	0.225
7	4	20	5	0	5	0.09900	0.33
8	5	90	10	4	6	0	0.569
9	5	90	10	5	5	0.02590	43
10	5	90	10	0	10	0.04810	611
11	8	380	20	13	7	0	3.77
12	8	380	20	0	20	0	2.97
13	8	304	20	0	20	0	8
14	12	1300	40	25	15	0	30.7
15	12	1278	40	25	15	0	14.418
16	12	510	40	25	15	0	23

Tabla 3. Resultados Modelo Programación Entera. Grupo 1.

Allí se observa que en todos los casos el modelo obtiene la solución óptima en un tiempo computacionalmente bajo. De igual manera, a medida que aumenta la cantidad de cursos a programar y/o el número de cursos con dos sesiones a la semana, el modelo necesita de mayor tiempo para lograr una solución.

El segundo grupo de instancias consta de pruebas donde la solución del modelo no llega a ser óptima. En la Tabla 4 se muestran estos resultados, se definieron dos tiempos máximos de ejecución: 3600 y 10000 segundos. Cabe resaltar que la función objetivo en la mayoría de las instancias es explicada por las penalizaciones en asignar cursos con incidencia. La tabla 3 reporta el GAP respecto al modelo relajado.

	Parámetros de Entrada					**Modelo Matemático**		
Instancia	Periodos	# Incidencias	# de Cursos	# de Cursos 1 sesión	# de Cursos 2 sesiones	FO	Tiempo(s)	GAP
17	12	5600	80	50	30	0.00875	3600	97%
17	12	5600	80	50	30	0.00196	10000	88%
18	12	5600	80	40	40	0.12661	3600	82%
18	12	5600	80	40	40	0.04001	10000	42%
19	12	5600	80	30	50	0.00661	3600	94%
19	12	5600	80	30	50	0.00065	10000	36%

Tabla 4. Resultados Modelo Programación Entera. Grupo 2.

Por último, para el modelo de programación entera se tiene el grupo de las instancias más grandes, las cuales tienen cursos de 150 y 230 respectivamente, el programa no encuentra soluciones enteras factibles. Cabe anotar que la situación actual del programa de Ingeniería Industrial asigna aproximadamente un total de 210 cursos por semestre, distribuidos en 45 a 50 asignaturas, por lo cual, realizar la respectiva programación de horarios del programa de ingeniería industrial no sería posible utilizando este método.

Ahora bien, utilizando la metaheurística como método de solución es posible encontrar soluciones cuando el tamaño de la instancia aumenta. La Tabla 5 muestra los resultados obtenidos. Se observa que a medida que aumenta el número de cursos a programar también el tiempo de procesamiento aumenta considerablemente ya que la metaheurística realiza pequeños cambios por iteración, necesitando una mayor cantidad de iteraciones para encontrar una solución adecuada.

	Parámetros de Entrada					Modelo Matemático			Metaheurística			
									Solución Inicial		Búsqueda Aleatoria	
Instancia	Periodos	# Incidencias	# de Cursos	# de Cursos 1 Bloque	# deCursos 2 Bloques	FO	Tiempo(s)	GAP	FO	Tiempo (s)	FO	Tiempo (s)
1	3	0	1	1	0	0	0.22	NA	0	0.035	0	0.0039
2	3	0	1	0	1	0	0.217	NA	0	0.035	0	0.007
3	3	6	3	0	3	0.11000	0.328	NA	0.11	0.08	0.11	0.015
4	3	6	3	0	3	0.11000	0.417	NA	0.11	0.05	0.11	0.085
5	4	20	5	3	2	0.03300	0.23	NA	0.066	0.117	0.033	0.1484
6	4	20	5	0	5	0.09900	0.225	NA	0.1998	0.085	0.1	0.15
7	4	20	5	0	5	0.09900	0.33	NA	0.1	0.07	0.1	0.17
8	5	90	10	4	6	0	0.569	NA	0.1796	0.17	0	0.035
9	5	90	10	5	5	0.02590	43	NA	0.1554	0.167	0.02593	1.1
10	5	90	10	0	10	0.04810	611	NA	0.1739	0.1523	0.048	1.429
11	8	380	20	13	7	0	3.77	NA	0.051	0.449	0	0.328
12	8	380	20	0	20	0	2.97	NA	0.051	0.449	0	0.19
13	8	304	20	0	20	0	8	NA	0.1	0.445	0	0.4521
14	12	1300	40	25	15	0	30.7	NA	0.058	1.38	0	12.13
15	12	1278	40	25	15	0	14.418	NA	0.058	1.38	0	13.15
16	12	510	40	25	15	0	23	NA	0.06	1.3	0	1.85
17	12	5600	80	50	30	0.00875	3600	97%	0.1509	2.67	0.00285	239.31
17	12	5600	80	50	30	0.00196	10000	88%	0.1509	2.67	0.00285	239.31
18	12	5600	80	40	40	0.12661	3600	82%	0.2339	2.42	0.0411	193.9
18	12	5600	80	40	40	0.04001	10000	42%	0.2339	2.42	0.0411	193.9
19	12	5600	80	30	50	0.00661	3600	94%	0.049	3.027	0.00095	268.35
19	12	5600	80	30	50	0.00065	10000	36%	0.049	3.027	0.00095	268.35
20	12	6448	150	100	50	Memoria	NA	NA	0.091	17.93	0.00103	750.58
21	12	1840	150	75	75	Memoria	NA	NA	0.1201	17.55	0.00059	1148.2
22	12	13312	150	40	110	Memoria	NA	NA	0.086	18.32	0.00013	835.83
23	12	2846	230	190	40	Memoria	NA	NA	0.074	41.6	0.00023	1718
24	12	8072	230	115	115	Memoria	NA	NA	0.068	39.55	0.00273	1635
25	12	2590	209	104	105	Memoria	NA	NA	0.114	40.26	0	244.79

Tabla 5. Resultados Modelo Matemático vs Búsqueda Aleatoria

De la misma manera se puede observar para cada instancia que la metaheurística llega al mismo valor que el modelo matemático en un menor tiempo, tanto para instancias pequeñas como para las instancias más grandes. Con los resultados de las instancias pequeñas donde el modelo matemático encontró la solución óptima diferente de cero se valida el funcionamiento de la Búsqueda Aleatoria, concluyendo que la programación de la metaheurística es suficiente para encontrar la misma solución que la resuelta por el modelo matemático, como se observa para las instancias 3 a 7.

Al comparar con el modelo matemático de las instancias más grandes la metaheurística encuentra soluciones cercanas o mejores en un tiempo mucho menor. Por ejemplo, para la instancia 17, que tiene un total de 80 cursos y 5600 cursos con incidencia, el modelo matemático con un tiempo de 3600 obtiene una función objetivo mayor. La función objetivo obtenida por medio de la metaheurística consigue una mejor solución y además disminuye el tiempo de ejecución en un 93%. Cuando aumenta el tiempo de ejecución del modelo matemático a 10.000 segundos el modelo encuentra una mejor solución, sin embargo, la metaheurística con 1000 iteraciones y un tiempo de ejecución considerablemente menor de 240 segundos consigue una función objetivo con una diferencia porcentual promedio de 37% comparándolo con el modelo matemático.

Se realiza igualmente una gráfica la cual explica el comportamiento de la función objetivo a medida que pasan las iteraciones. En la Figura 5 se muestra el comportamiento para una instancia pequeña de 5 cursos, pero que, según los parámetros de entrada, la solución óptima no da un valor en la función objetivo de cero ya que consiste en programar dos cursos con incidencia que tienen el mismo periodo. Se observa en esta figura que después de un número de iteraciones, la

metaheurística encuentra el óptimo (0,033) y así mismo, después sigue buscando nuevas soluciones encontrando como límite inferior el óptimo.

En cambio en la Figura 6 y **¡Error! No se encuentra el origen de la referencia.**, el comportamiento de la función objetivo es muy parecida, ya que, al tener una mayor cantidad de cursos por programar sin aumentar el número de periodos ni los días, realiza una disminución significativa en la función objetivo dentro de las primeras iteraciones, luego de esto, se queda en el nivel más bajo encontrado hasta esa iteración.

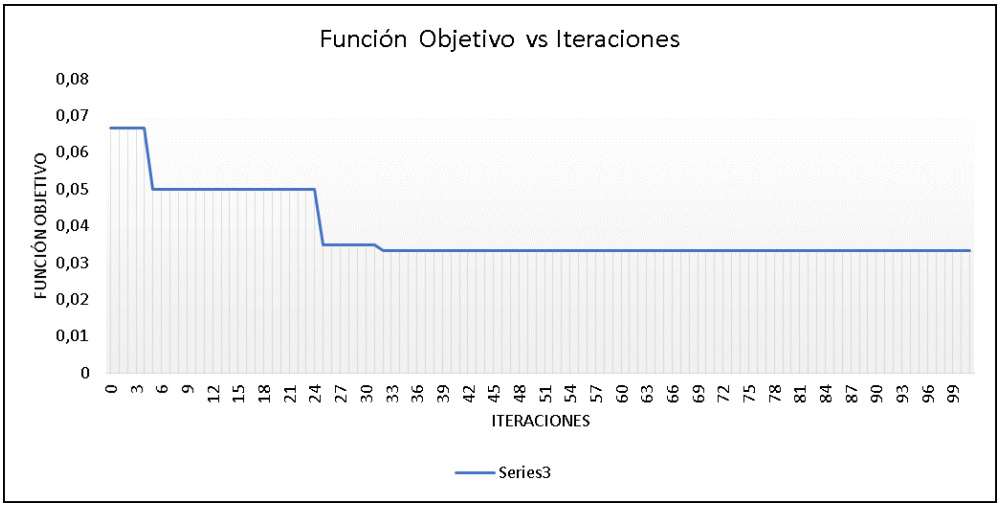

Figura 5. Gráfica Fo vs Iteraciones instancia 5- Cinco Cursos

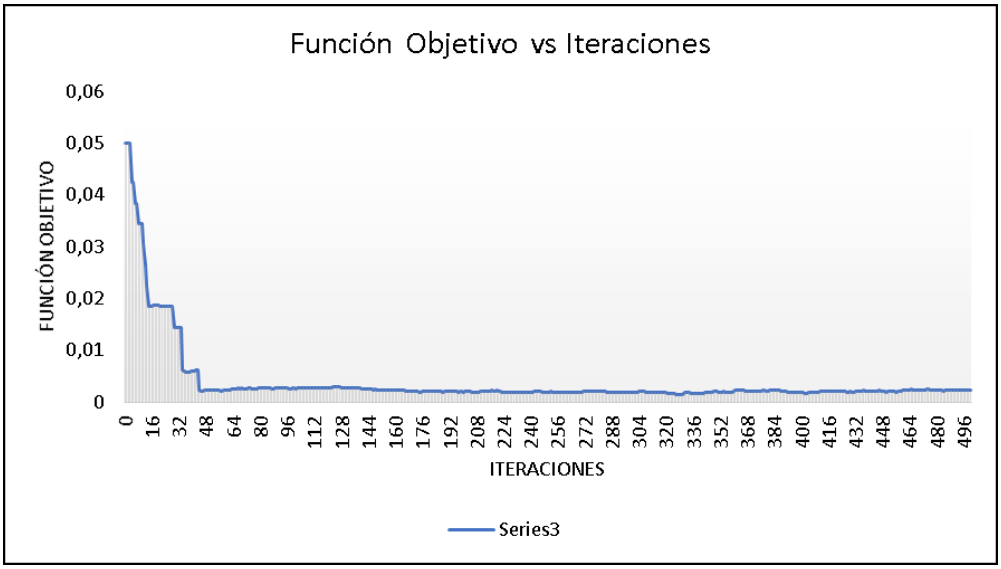

Figura 6. Gráfica Fo vs Iteraciones instancia 17- Ochenta cursos

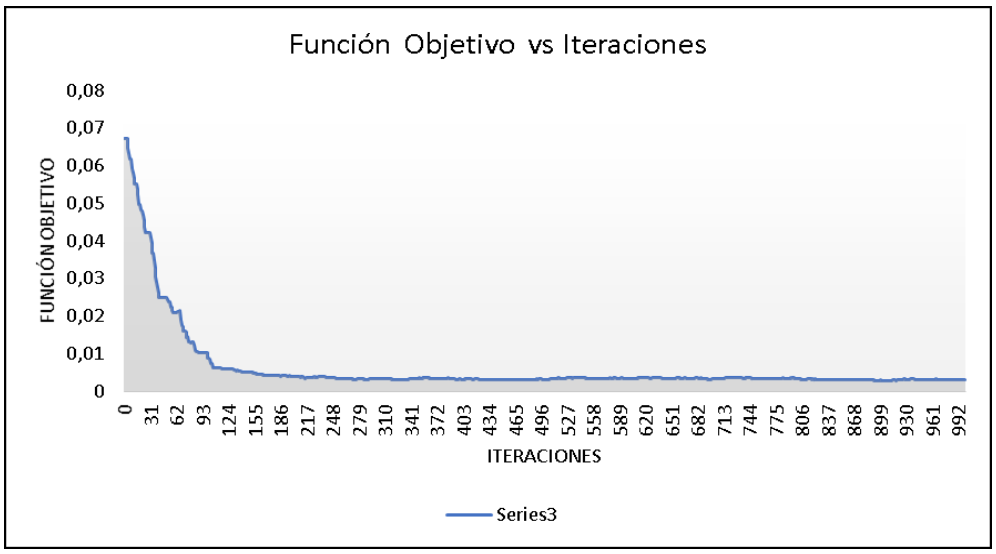

Figura 7. Gráfica Fo vs Iteraciones instancia 24- Cursos 230

Finalmente, teniendo en cuenta los parámetros escogidos, se procede a realizar la prueba con el semestre 2015-III teniendo en cuenta los mismos datos de entrada. Después de evaluar cada una de las programaciones se tiene:

	Modelo Actual	Búsqueda Tabú
Número de Cursos	209	209
Número de Incidencias	2590	2590
FO	0,09427	0
A	17	0
B	212	0
C	25	0
Tiempo	NA	245
Iteración	NA	153

Tabla 3. Comparación Modelo Actual vs Búsqueda Aleatoria

En el cual se observa que la metaheurística alcanza al valor óptimo, lo cual significa que realizó la asignación de los 209 cursos sin ninguna penalidad. Por otro lado, el modelo del semestre 2015-III tiene una función objetivo mayor, ya que asigna 17 cursos en el periodo No deseado, igualmente asigna cursos con incidencia en el mismo periodo de tiempo y finalmente

realiza 25 asignaciones en horarios no preferenciales. En la Figura 8 Se observa el comportamiento de la función objetivo a medida que aumenta el número de iteraciones, disminuyendo hasta llegar a cero.

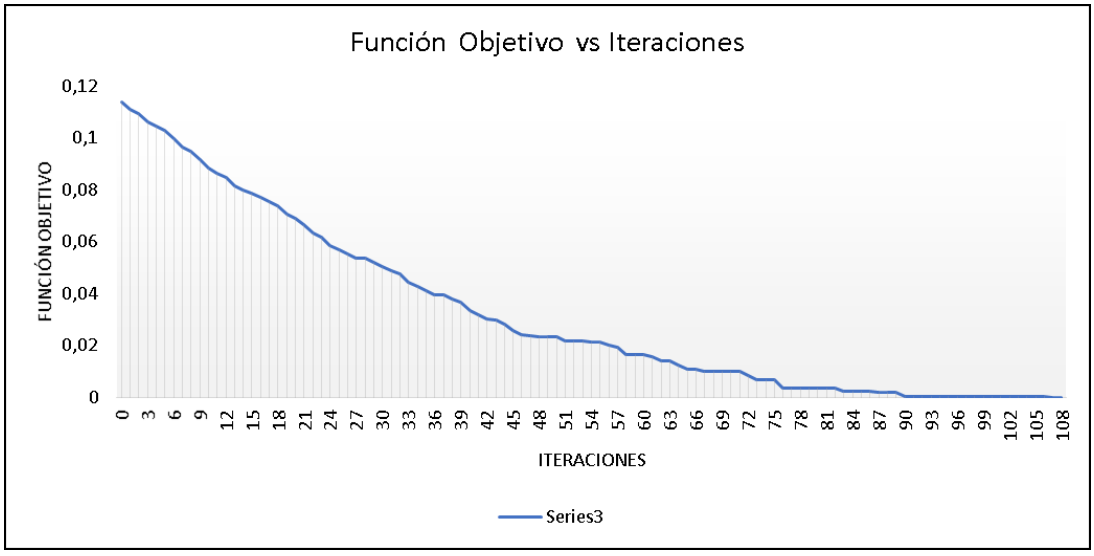

Figura 8. Gráfica Fo vs Iteraciones. Semestre 2015-III

Igualmente se realiza la

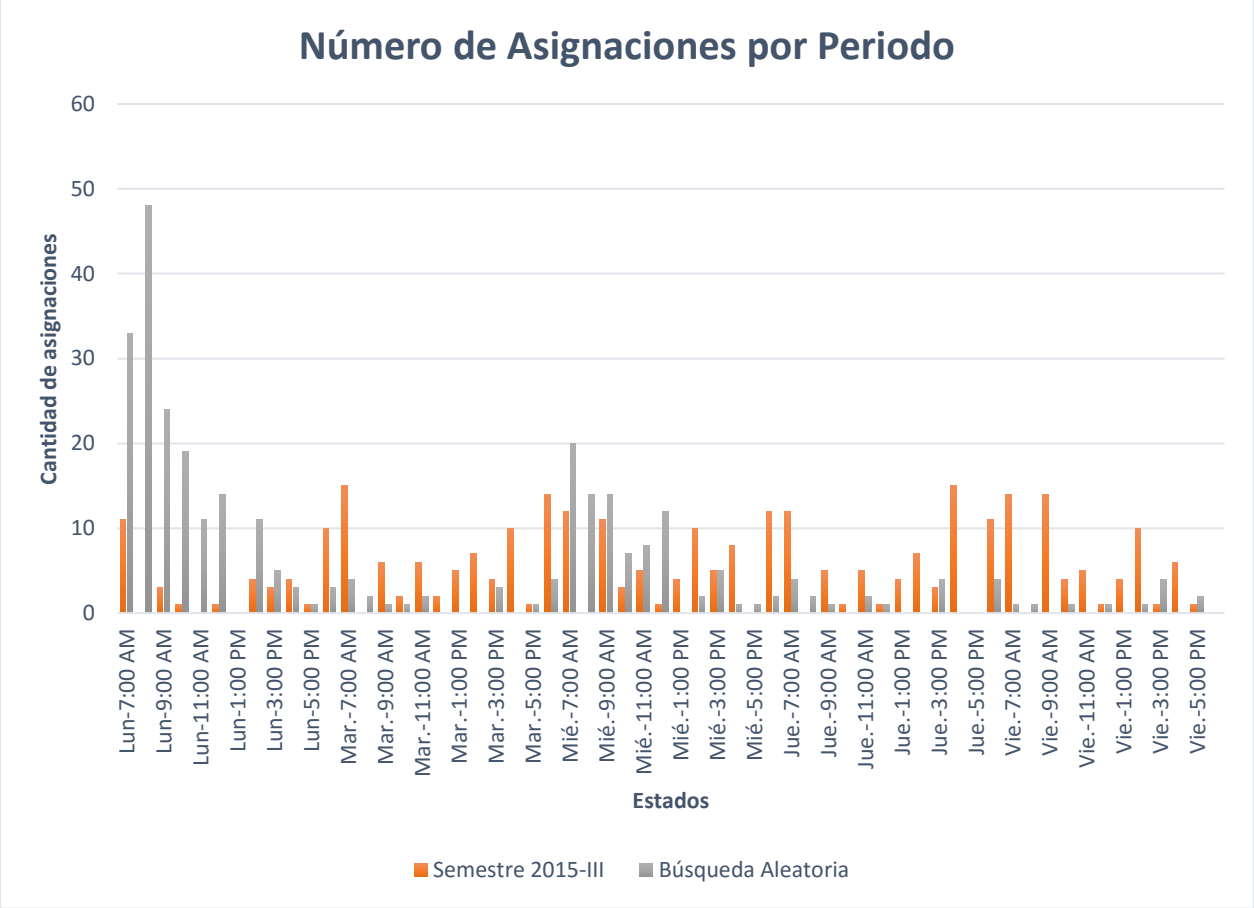

Figura 99 en donde se observa la cantidad de asignaciones realizadas por cada estado para ambas situaciones (el modelo del semestre 2015-III y la búsqueda aleatoria). La metaheurística no asigna en la franja horaria que corresponden al periodo No deseado (1:00PM) y adicionalmente la mayoría de asignaciones se realizan los primeros días de la semana, esto es dado por la solución inicial de la metaheurística que programa en la primera franja de preferencia disponible. Por otro lado, la programación del semestre 2015-III tiene una distribución mayor en los últimos periodos de la semana.

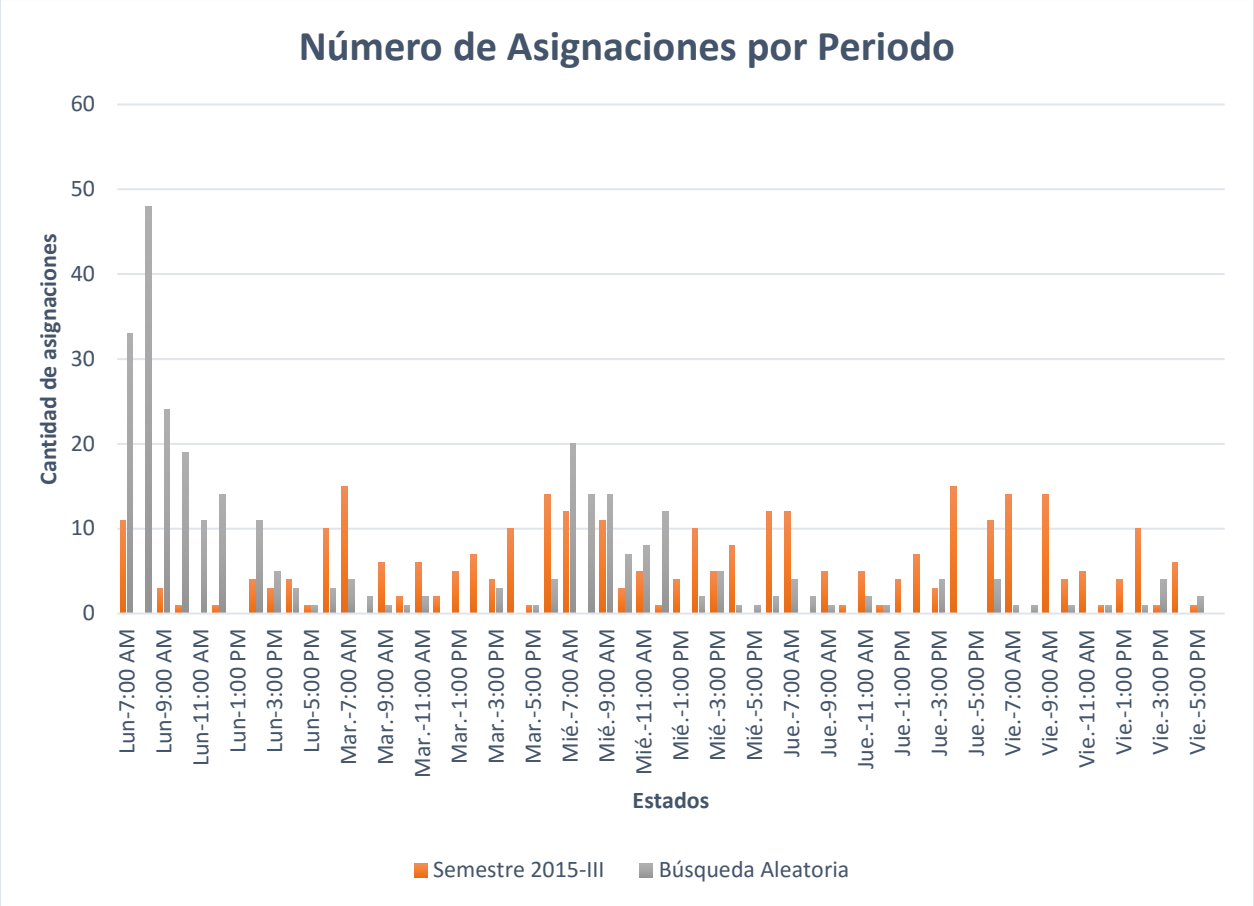

Figura 9. Cantidad de asignaciones por estado.

Como se observa en la figura anterior, la metaheurística realiza asignaciones en horarios que normalmente en la universidad Javeriana no se realizan, por ejemplo, asignar una materia para que empiece en la franja horaria 8:00 AM o 10:00 AM. Por lo tanto, este nuevo criterio restringe aún más la preferencia de horarios para la metaheurística. Conociendo esto, se genera un nuevo horario preferencial en donde no se tenga asignado estos periodos intermedios.

Los resultados obtenidos para este modelo ajustado son:

	Modelo Actual	Búsqueda Tabú
Número de Cursos	209	209
Número de Incidencias	2590	2590
FO	0,09427	0
A	17	0
B	212	0
C	25	0
Tiempo	NA	463,3
Iteración	NA	312

Tabla 4. Comparación Modelo ajustado. Actual vs Búsqueda Aleatoria

En la siguiente figura se observa que para la metaheurística ya no hay asignaciones en los periodos que se dijo anteriormente, distribuyendo las asignaciones en los primeros días de una manera uniforme. Sin embargo, estas restricciones incrementan el tiempo computacional de la metaheurística.

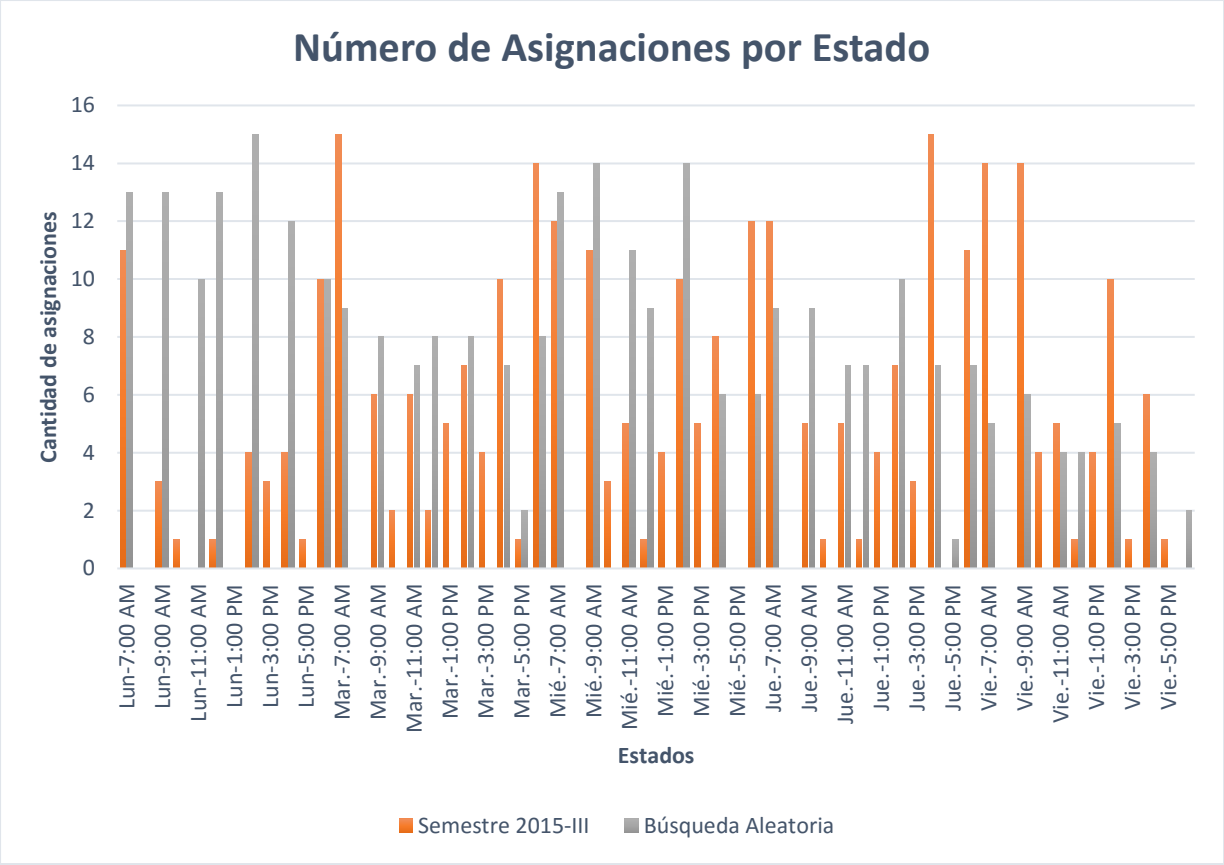

Figura 10. Cantidad de asignaciones por estado, modelo ajustado.

Finalmente, se obtiene con la búsqueda aleatoria una programación de los horarios que satisface las restricciones duras y blandas impuestas por el programa de Ingeniería Industrial, además de mejorar el tiempo de ejecución del proceso de programación de horarios

7. Conclusiones

Este proyecto aborda el problema de programación de horarios y asignación de clases universitarias, catalogado en la literatura como un problema NP-Hard. Se realiza la investigación para la Pontificia Universidad Javeriana, específicamente para el programa de Ingeniería Industrial, el cual para el semestre 2015-III, tiene un total de 46 asignaturas distribuidas en 209 cursos.

Se diseña un modelo que minimiza la cantidad de clases asignadas a horarios de inicio no preferenciales y la cantidad de asignaciones de cursos obteniendo soluciones óptimas para las instancias que contienen menos de 80 cursos con un tiempo computacional por debajo de los 1.000 segundos. Sin embargo, para la programación de instancias con 80 cursos el modelo matemático no encuentra soluciones óptimas después de 10.000 segundos y, por consiguiente, para instancias mayores como es en la situación real, no logra siquiera encontrar soluciones.

Ahora bien, se aplica una técnica de solución combinatoria (Búsqueda Aleatoria) que encuentra soluciones cercanas o iguales a las presentadas por el modelo matemático, pero en un tiempo computacional significativamente menor. Se concluye que para instancias pequeñas el tiempo con la metaheurística mejora en un 30%, consiguiendo soluciones iguales al modelo matemático. Cuando se prueba la metaheurística en problemas más grandes, el tiempo computacional puede disminuir hasta en un 93%, obteniendo soluciones cercanas a las de la programación lineal.

Para realizar las instancias más grandes se parametriza el algoritmo, tomando en cuenta el Tamaño de la lista de Candidatos y el Número de iteraciones, utilizando una prueba de diferencias de medias T2 de TamHane con lo que se obtiene una parametrización de 20 candidatos y un número de 1000 iteraciones. A diferencia de la programación lineal para

instancias superiores de 80 cursos, la metaheurística encuentra soluciones en un tiempo promedio de 1.200 segundos.

Finalmente, al realizar la simulación del semestre 2015-III con la técnica de solución, ésta asigna el total de cursos sin generar una penalidad en ninguno de los criterios escogidos. Mientras que, al calcular la función objetivo de la programación real los cursos del Semestre 2015-III, se encuentra que se realizaron 17 asignaciones en la franja horaria (1:00PM), 212 asignaciones de cursos con alta relación de inscripción en el mismo periodo y 25 asignaciones de cursos en horario no preferencial. Concluyendo que la metaheurística mejora la programación de los cursos realizando la prueba para el tercer semestre de 2015 y tomando en cuenta los criterios de preferencia e incidencia.

8. Recomendaciones

Se realizan las siguientes recomendaciones para un trabajo futuro:

- Realizar un análisis de sensibilidad exhaustivo sobre las variables de ponderación de la función objetivo (α, β, θ) para determinar el valor en el que mejor se adapten a los requerimientos del programa.
- Realizar un criterio de parada que tome en cuenta un periodo máximo de estabilización en el que la función objetivo no varía significativamente.
- Realizar un modelo matemático que busque minimizar el número de asignaciones los días lunes, evitando programar clases los días festivos.
- Realizar una parametrización que tome en cuenta más variables del algoritmo como: Periodo Tabú, Criterio de aspiración, periodo máximo de estabilización, etc.

Referencias

Abdullah, S., & Turabieh, H. (2012). On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences*, 146-168.

Acosta, R., & Marulanda, M. (2013). An Integer Programming Model for The Academic Timetabling.

Aladag, C., Hocaoglu, G., & Basaran, M. (2009). The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. *Expert Systems with Applications*.

Al-Yacoob, S., & Sherali, H. (2005). Mathematical programming models and algorithms for a class–faculty assignment problem.

Babaei, H., Karimpour, J., & Hadidi, A. (2014). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 43-59.

Bakir , A., & Aksop, C. (2008). A 0-1 Integer Programming Approach To A University Timetabling Problem.

Batista, B. M., & Glover, F. (2003). Búsqueda tabú. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, 29-48.

Bonrostro, P., & Yusta, C. (2003). Estudio comparativo de diferentes estrategias metaheurísticas para la resolución del labor scheduling problem. *Estudios de economía aplicada*.

Casey, S., & Thompson, J. (2003). GRASPing the examination scheduling problem. *In Practice and Theory of Automated Timetabling IV*, 232-244.

Chaudhuri, A. (2010). Fuzzy genetic heuristic for university course timetable problem. *International journal of Advance Soft Computing Application*.

Cifuentes, C. (2012). Programación de Estudiantes un Caso de Estudio (Universidad Central) . *Congreso Latino-Iberoamericando de Inevstigación Operativa*.

Constantino, A., Marcondes, W., & Landa-Silva, D. (2010). Iterated heuristic algorithms for the classroom assignment problem. *In Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling-PATAT*.

Cortez, A., Rosales, G., Naupari, R., & Vega, H. (2010). Sistema de apoyo a la generación de horarios basado en algoritmos genéticos. *Revista de Investigación de Sistemas e Informática*, 37-55.

Daskalaki, S., & Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 106-120.

De Causmaecker, P., Demeester, P., & Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, 307-318.

Di Gaspero, L., & Schaerf, A. (2001). Tabu search techniques for examination timetabling. *In Practice and Theory of Automated Timetabling III* , 104-117.

Dowsland, K., & Adenso-Díaz, B. (2003). Heuristics Design and Fundamentals of the Simulated Annealing.

Dowsland, K., & Thompson, J. (2005). Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 426-438.

Franco, J., Toro, E., & Gallego, R. (2008). Problema de asignación óptima de salones resuelto con búsqueda Tabú. *Universidad del Norte*, 149-175.

Galbiati Riesco, J. (2007). Diseño de Experimentos Factoriales con aplicaciónes a Procesos Industriales.

Gallego, R., Escobar, A., & Toro, E. (2008). Técnicas metaheurísticas de optimización. *Universidad Tecnológica de Pereira*, 92.

González, E. (2009). Un Algoritmo De Búsqueda Tabu Para Generar El Calendario De Exámenes De La Nueva Facultad De Ciencias De La Educación.

Gutiérrez, P. H., & De la Vara , S. R. (2004). *Análisis y diseño de experimentos.* México D.F: McGraw-Hill.

Hernández, R., & Miranda, J. (2008). Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad de Diego Portales Mediante un enfoque de Programación Entera. *Ingeniería de Sistemas*.

Herrera, F. (2006). Introducción a los algoritmos metaheurísticos. *Ciencias de la Computación e IA*.

ITC. (2007). International Timetable Competition. Metaheuristics Network. Obtenido de http://www.cs.qub.ac.uk/itc2007/index_files/overview.htm

Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR spectrum*, 167-190.

López, P. (2000). El método de búsqueda tabú para la programación de horarios. *Universidad de Sonora*.

Mejía Caballero, J. M., & Paternina Arboleda, C. D. (2009). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Educación en Ingeniería*.

Miranda, J., & Rey, P. (2012). n modelo de programación entera basado en patrones para la asignación de salas de clases para una facultad de medicina. *In Congreso Latino-Iberoamericano de Investigación Operativa.*

MirHassani, S. (2006). Improving paper spread in examination timetables using integer programming. *Applied Mathematics and Computation*.

Norton, R. (2009). *Diseño de Maquinaria.* McGraw-Hill.

Orejuela, J., Manyoma, P. C., & González, G. (2011). Methodology to determine the installed capacity of an academic program. *Estudios Gerenciales*, 143-158.

Osman, I., & Kelly, J. (2012). *Meta-heuristics: theory and applications.* Springer Science & Business Media.

Pacheco, C. (2000). Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos. *Universidad Tecnológica de la Mixteca*.

Pacheco, C. L. (2000). Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos.

Peñuela, C. A., Franco, J. F., & Toro, E. (2008). Colonia de hormigas aplicada a la programación óptima de horarios de clase. *Scientia et Technica*.

Pichardo, M. (2011). Revisión de Algoritmos Genéticos Aplicados al Problema de la Programación de Cursos Universitarios. *Universidad Autónoma del Estado de Morelos*.

Pongcharoen, P., Promtet, Yenradee, & Hicks. (2008). Stochastic Optimisation Timetabling Tool for university course scheduling. *International Journal of Production Economics*.

Rojas, L., Saldaña, A., & Oliva San Martín, C. (2007). Modelos de programación entera para un problema de programación de horarios para universidades. *Revista chilena de ingeniería*, 245-259.

Ross, P., Corne, D., & Fang, H.-l. (1994). *Fast practical evolutionary timetabling.* Berlin: Springer Berlin Heidelberg.

Ross, P., Hart, E., & Corne, D. (1998). Some observations about GA-based exam timetabling. *In Practice and Theory of Automated Timetabling II*, 115-129.

Saldaña, A., Oliva, C., & Pradenas, L. (2007). Modelos De Programación Entera Para Un Problema De Programación De Horarios Para Universidades.

San Martín, C., & Guzmán, M. (2013). Algoritmo de tipo búsqueda tabú para un problema de programación de horarios universitarios vespertinos. *INGE CUC*.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, 87-127.

Suaréz, J. G., Manchego, F. A., & Nicho, A. (2010). SUÁREZ, Joseph Gallart; MANCHEGO, Fernando Alva; NICHO, Anthony Alama Nole3 Gissella Bejarano. Generación Inteligente de Horarios empleando heurísticas GRASP con Búsqueda Tabú para la Pontifica Universidad Católica del Perú. *Revista Ingeniería Informática*, 15-24.

TC69, I. S. O. . (2011). *SC7–Applications of statistical and related techniques for the implementation of Six Sigma: ISO 13053-1: 2011-Quantitative methods in process improvement–Six Sigma–Part 1: DMAIC methodology.*

Torres Ovalle, C. (2013). *Programación de horarios y asignación de aulas de clases universitarias.* Chia: Universidad de la Sabana.

Vermuyten, H., Lemmens, S., Marques, I., & Belien, J. (2015). Developingcompactcoursetimetableswithoptimizedstudentflows.

White, G., & Xie, B. (2001). Examination timetables and tabu search with longer-term memory. *In Practice and Theory of Automated Timetabling III* , 85-103.