



UNIVERSIDAD NACIONAL DE COLOMBIA

**Disminución de la mortalidad
académica en instituciones de
educación básica y media
mediante el empleo de
técnicas inteligentes en la
asignación de horarios**

Victo Fabio Suarez Chilma

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Industrial
Manizales, Colombia
2012

**Disminución de la mortalidad académica en
instituciones de educación básica y media
mediante el empleo de técnicas inteligentes
en la asignación de horarios**

**Academic decreased mortality in institutions
of primary and secondary education by using
intelligent techniques in timetabling**

Victor Fabio Suarez Chilma

Tesis o trabajo de investigación presentada(o) como requisito
parcial para optar al título de:
Magíster en Ingeniería - Ingeniería Industrial

Director (a):
Doctor Omar Danilo Castrillón Gómez

Línea de Investigación:
Dirección y producción de operaciones
Grupo de Investigación:
Innovación y desarrollo tecnológico

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Industrial
Manizales, Colombia
2012

A Dios, por estar conmigo en cada paso, por fortalecer mi corazón y mi mente en los momentos adversos y por haber puesto en mi camino a todas aquellas personas que han sido mi soporte y compañía durante esta etapa.

A mi familia y a las personas que llevo en el corazón, por el amor con el que me han tratado, por la paciencia ante la ausencia, por las muchas palabras de aliento y por el apoyo incondicional en la consecución de esta meta.

Agradecimientos

Al llegar a la culminación de esta etapa, quiero reconocer mi agradecimiento a quienes de una u otra forma contribuyeron en el desarrollo de este proyecto.

Primero que todo a mi familia y a las personas que llevo en el corazón, por estar siempre a mi lado, por ser siempre incondicionales.

A Álvaro Guerrero, por sus meritorios aportes, por su dedicación y ante todo, por su valiosa amistad.

Al profesor Omar Danilo, por sus orientaciones y permanente interés en el desarrollo del proyecto.

A mis profesores y compañeros, por todas las nuevas ideas, las enseñanzas y el apoyo que me brindaron.

Al Colegio San Jorge y sus directivas, por su colaboración para el desarrollo de la prueba piloto.

A la Universidad Nacional de Colombia, por la financiación de este proyecto a través del Hermes (Cod. 13744).

Resumen

Esta investigación plantea la solución del problema de programación de horarios escolares bajo las características de la escuela pública colombiana incluyendo como factor primordial dentro del proceso de optimización, la ubicación horaria de las asignaturas en función de los ritmos cognitivos que presentan los estudiantes durante la jornada escolar. Para ello, se presenta el desarrollo de una aplicación en la cual se implementaron tres diferentes técnicas inteligentes a través de las cuales se busca la minimización de una función de evaluación que representa la calidad de la solución de acuerdo al enfoque establecido. Los resultados muestran una mayor eficiencia del algoritmo NSGA-II en comparación con los otros dos métodos aplicados al mismo problema y evaluados con la misma métrica. A nivel experimental, la metodología evidencia que los grupos programados mediante la orientación descrita presentan una reducción en los niveles de mortalidad académica.

Palabras clave: Programación de horarios escolares, Algoritmo genético, NSGA-II, restricciones, ritmos cognitivos, jornada escolar, escuela pública.

Abstract

This research raises the solution to school timetabling problem under the characteristics of Colombian public school including as a primary factor in the optimization process, the time location of the subjects according to cognitive rhythms presented by students during the school day. For this purpose, a development application is presented in which three different techniques were implemented through which the minimization for a Fitness function is looked for that represents the quality of the solution according to the established approach. The results show a higher efficiency of the NSGA-II algorithm compared to the other two methods applied to the same problem and evaluated with the same metric. At the experimental level, the methodology shows that the programmed groups through the guidance described show a reduction in the academic mortality levels.

Keywords: School Timetabling, Genetic Algorithm, NSGA-II, Restrictions, Cognitive Rhythms, School Day, Public School.

Contenido

	Pág.
Resumen.....	IX
Lista de figuras.....	XIII
Lista de tablas.....	XIV
1. Planteamiento del problema	3
1.1 Justificación	7
1.2 Objetivo general	8
1.3 Objetivos específicos	9
1.4 Hipótesis	9
1.5 Resultados esperados	10
1.6 Criterios de éxito	10
1.7 Metodología de la investigación	11
2. Marco teórico	13
2.1 Tipos de programación	13
2.2 Asignación de horarios educativos	14
2.2.1 Asignación de horarios escolares	14
2.2.2 Asignación de horarios universitarios	15
2.2.3 Asignación de horarios de exámenes	16
2.3 Métodos de solución	16
2.3.1 Concepto general de optimización	17
2.3.2 Técnicas para la solución del problema de asignación de horarios.....	19
2.3.3 Antecedentes del problema en el contexto local	24
2.4 Ritmos cognitivos	25
2.4.1 La cronobiología y cronopsicología	25
2.4.2 Factores asociados con la ritmicidad cognitiva	26
2.4.3 Consideraciones dentro del modelo educativo colombiano	27
2.4.4 Propuesta acorde al modelo educativo analizado	28
3. Materiales y métodos	29
3.1 Generalidades del algoritmo	29
3.2 Estructura general del algoritmo	30
3.3 Ingreso y validación de la información	30
3.3.1 Codificación de los datos y construcción de la tabla	31
3.3.2 Evaluación de las restricciones de tabla	34
3.3.3 Construcción de la solución inicial	35
3.4 Construcción de la solución de acuerdo al método seleccionado	36

XII Disminución de la mortalidad académica en instituciones de educación básica y media mediante el empleo de técnicas inteligentes en la asignación de horarios

3.4.1	Tipos de restricciones.....	36
3.4.2	Funciones de evaluación.....	38
3.4.3	Método de solución 1: Algoritmo genético NSGA-II.....	39
3.4.4	Método de solución 2: Algoritmo genético simple.....	56
3.4.5	Método de solución 3: Búsqueda Aleatoria.....	61
3.5	Presentación de resultados	66
4.	Resultados	69
4.1	Estabilidad del sistema	69
4.2	Comparación por método	70
4.3	Comparación por tiempos de ejecución	74
4.4	Prueba piloto	75
5.	Discusiones.....	79
6.	Conclusiones.....	83
A.	Anexo: Tablas de resultado para pruebas de estabilidad del sistema y comparación de tiempos de ejecución	87
B.	Anexo: Productos Académicos.....	99
C.	Anexo: Certificación prueba piloto.....	137
	Bibliografía	139

Lista de figuras

	Pág.
Figura 3-1. Diagrama general del algoritmo.....	30
Figura 3-2. Esquema de hipermatrices.....	35
Figura 3-3. Mecanismo de evolución del NSGA-II.....	39
Figura 3-4. Diagrama de flujo de operación del NSGA-II.....	40
Figura 3-5. Proceso para aplicar operador de cruce.....	43
Figura 3-6. Proceso para aplicar operador de mutación.....	44
Figura 3-7: Distribución gráfica de los frentes.....	54
Figura 3-8: Representación gráfica factor de diversidad.....	55
Figura 3-9. Diagrama de flujo de operación del algoritmo genético simple.....	57
Figura 3-10. Diagrama de flujo de operación del algoritmo búsqueda aleatoria.....	62

Lista de tablas

	Pág.
Tabla 3-1. Tabla ejemplo de información base.....	31
Tabla 3-2. Tabla de equivalencias para códigos de área.....	33
Tabla 3-3: Valores de penalización restricciones blandas.....	39
Tabla 3-4. Resultados Función F_{REST3}	48
Tabla 3-5. Ejemplo comunidad para selección de frentes.....	51
Tabla 3-6: Resultado de aplicación función FND_Ordenado para hallar primer frente.....	52
Tabla 3-7: Ejemplo de aplicación función FND_Ordenado.....	53
Tabla 3-8. Ejemplo de selección mediante algoritmo genético simple.....	59
Tabla 3-9. Ejemplo de selección mediante algoritmo genético simple.....	60
Tabla 3-10. Tabla base para organizar bloques en algoritmo de búsqueda aleatoria.....	63
Tabla 3-11. Organización de bloques en algoritmo de búsqueda aleatoria.....	63
Tabla 3-12. Ejemplo resultados de materias.....	67
Tabla 3-13. Ejemplo resultado profesores.....	67
Tabla 4-1. Problemas de prueba aplicados.....	69
Tabla 4-2. Comparación por repetición para cada método.....	70
Tabla 4-3. Resultados totales de la función Fitness por problema y método.....	70
Tabla 4-4. Descriptivo de resultados totales de la función Fitness por problema y por método.....	71
Tabla 4-5. Prueba de normalidad resultados totales de la función Fitness por problema y método.....	72
Tabla 4-6. Prueba de homogenidad de varianzas para resultados totales de la función Fitness por problema y método.....	72
Tabla 4-7. Prueba ANOVA para comparación del resultado total de la función Fitness entre métodos.....	72
Tabla 4-8. Prueba Post hoc para comparación del resultado total de la función Fitness entre métodos.....	73

Tabla 4-9. Descriptivo de los tiempos de ejecución de cada uno de los métodos para cada uno de los problemas planteados.	74
Tabla 4-10. Información académica por niveles primer periodo.....	76
Tabla 4-11. Información académica por niveles segundo periodo.....	76
Tabla A-1. Resultado Fitness por grupo, repetición y problema para algoritmo de búsqueda aleatoria.....	87
Tabla A-2. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo de búsqueda aleatoria.....	88
Tabla A-3. Resultado Fitness por grupo, repetición y problema para algoritmo genético simple.	89
Tabla A-4. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo genético simple.	90
Tabla A-5. Resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II.....	91
Tabla A-6. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II.....	92
Tabla A-7. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II.	94
Tabla A-8. Tiempos de ejecución por método y por repetición.....	98
Tabla A-9. Tiempos de ejecución por método y por repetición.....	98
Tabla A-10. Tiempos de ejecución por método y por repetición.....	98
Tabla A-11. Tiempos de ejecución por método y por repetición.....	98

Introducción

Esta investigación muestra la aplicación de tres diferentes técnicas inteligentes para la solución del problema de asignación de horarios escolares, a partir de un enfoque en los ritmos cognitivos que presentan los estudiantes durante la jornada escolar. Se tiene como objetivo realizar una programación pertinente de acuerdo a la dificultad intelectual y el grado de nivel atencional requerido por cada asignatura, de manera que su implementación permita minimizar la mortalidad académica, principalmente en aquellas áreas que requieren un mayor esfuerzo intelectual y consecuentemente reducir la tasa de repitencia en las instituciones.

El trabajo expone como a través de una asignación de horarios en la que se consideran los ritmos de eficiencia cognitiva, es posible que aspectos como la memoria y la atención se potencialicen, propiciando una mayor disposición para el aprendizaje.

El documento parte de una descripción conceptual en la que se explica el planteamiento general del problema de asignación de horarios, las diferentes corrientes que a nivel investigativo existen del mismo y los diferentes métodos aplicados para su solución. Posteriormente se establecen los elementos teóricos mediante los cuales se vinculan los ritmos biológicos al desempeño de los estudiantes durante la jornada escolar. Por último, se contextualiza el problema en la escuela pública Colombiana y se exponen los referentes científicos bajo los cuales se plantea el nuevo enfoque de solución de esta investigación.

A nivel metodológico, se presenta el desarrollo de una aplicación en la cual se implementaron tres diferentes técnicas inteligentes basadas en métodos aleatorios y algoritmos genéticos, con las que se logra realizar la asignación en función de un conjunto restricciones temporales, relacionadas con el momento propicio de la jornada para la ejecución de determinadas actividades intelectuales. En este apartado se encuentra el modelo matemático desarrollado para la solución del problema y se describe de manera detallada el modo de operación de cada una de las técnicas.

A nivel computacional, las técnicas fueron sometidas a un conjunto de problemas con los cuales se evaluó el nivel de eficiencia de las soluciones obtenidas, la confiabilidad del sistema y los

tiempos de ejecución. El algoritmo NSGA-II es el que presenta un mayor nivel de optimización en las soluciones obtenidas, siendo hasta 30% y 45% superior en eficiencia en comparación con el algoritmo genético simple y el algoritmo de búsqueda aleatoria, respectivamente.

Adicionalmente, se muestran las observaciones y resultados obtenidos de una prueba piloto llevada a cabo sobre los estudiantes de un plantel educativo de la ciudad de Manizales, en la que se compararon los resultados académicos de un conjunto de grupos cuya jornada se programó bajo el modelo propuesto, con los resultados de otro conjunto cuya programación se estableció mediante asignación aleatoria. El contraste muestra que los grupos en los que la programación horaria se basa en los ritmos de eficiencia presentan una disminución de hasta un 28% en la mortalidad académica de sus estudiantes.

1. Planteamiento del problema

La jornada escolar es el tiempo que destinan las instituciones de educación básica y media para la prestación directa del servicio educativo, la cual tiene una distribución horaria que debe ajustarse a los requerimientos legales vigentes, el proyecto educativo institucional y el plan de estudios establecido para un determinado año lectivo. Por lo general, la programación de dicha jornada se asocia exclusivamente a la distribución de aulas y docentes, cumpliendo sólo las restricciones horarias de la institución, evitando el solapamiento en la asignación de cada docente y adicionando algunas condiciones particulares de acuerdo a situaciones específicas de cada establecimiento. No obstante, esta estructura temporal no sólo supedita la administración de los recursos de infraestructura y de planta docente, sino que tiene implícitos factores asociados al desempeño del estudiante, los cuales pocas veces son estimados a la hora de asignar los horarios de clase. El principal de ellos es el relacionado con los ritmos de aprendizaje, bajo el cual se establece que los estudiantes no tienen la capacidad cognitiva para asimilar todo tipo de conocimiento o actividad de aprendizaje en cualquier momento de la jornada. Esto se debe a que la actividad mental del ser humano se encuentra supeditada a ciclos en los que posee un mayor o menor nivel de precisión en la ejecución de tareas, los cuales varían en periodicidad, encontrando ritmos de alta, media y baja frecuencia. Para el caso de la jornada escolar se han realizado investigaciones en las que se estudian los ritmos medios o también conocidos como ritmos circadianos (periodos entre 20 y 28 horas), los cuales facilitan el análisis de desempeño de acuerdo a factores como: características de la tarea (complejidad, componente motor), diferencias individuales (edad, sexo, cronotipo, estilo cognitivo, nivel de motivación) y factores socioculturales (urbano, rural) (Hederich Martínez, Camargo Uribe, & Reyes Cuervo, 2004).

La asignación de horarios escolares tiene un enfoque en las clases y no en los estudiantes, por lo que la jornada diaria (mañana o tarde) se programa de forma continua, dividida en periodos de tiempos preestablecidos, para el desarrollo de actividades académicas en un número dado de asignaturas. Estas asignaturas se asocian a un nivel o grado respectivo, razón por la cual son cursadas por un mismo grupo de estudiantes, los cuales las toman a modo de curso obligatorio y no por demanda como sucede con los cursos universitarios. En este sentido, la planta física también

cumple con la orientación a las clases, ya que los salones se asignan de forma permanente a un mismo grupo y se propicia de manera sistemática la rotación de los docentes por cada una de las aulas, según la distribución que se haga del horario en la asignación. No obstante, se deben tener en cuenta la existencia de aulas multipropósito o de uso especial, las cuales también pueden ser incluidas dentro de la programación para ser dispuestas de manera permanente en un horario dado para alguno de los grupos. En cuanto a los docentes, la asignación debe garantizar que un docente no esté en dos grupos diferentes al mismo tiempo, o que su asignación durante toda la programación semanal no supere un número preestablecido de horas de trabajo dirigido.

Desde la perspectiva anterior, el problema planteado no escapa de los alcances hechos por trabajos previos. Sin embargo, la presente investigación establece la construcción de horarios de clase acogiendo no sólo las restricciones generales sobre infraestructura y capacidad, sino incluyendo también un acercamiento a la práctica pedagógica, al considerar los momentos de la jornada en los que es más propicio el aprendizaje para ciertas áreas del conocimiento. De acuerdo con esto, es necesario centrarse en la pregunta ¿cuándo enseñar? ó ¿cuándo es mejor el desarrollo de ciertas actividades académicas?. En este sentido, se encuentran diferentes enfoques, abordando de manera específica tres tipos de orientaciones: la primera hace referencia a la implementación de una jornada continua o una jornada dividida (Feito Alonso, 2007), la segunda, a la duración de la jornada (Banco Mundial, 2009), y la tercera, a la hora la hora del día en las que es mejor el desarrollo de determinadas acciones de aprendizaje (Hederich Martinez, Camargo Uribe, & Reyes Cuervo, 2004).

Para este caso, el enfoque tomado se encuentra sustentado en la orientación sobre la hora del día más apropiada para la enseñanza. Para ello, adaptando las consideraciones hechas por Hederich et al. (2004), y ajustándolas a los parámetros generales de las instituciones públicas de Colombia, donde la jornada escolar tiene una duración de seis horas diarias y es continua, se determinaron tres instancias de tiempo en las cuales la actividad atencional y cognitiva es más propicia para cierto tipo de asignaturas. En la sección de materiales y métodos se presenta la forma de distribución acogida, la cual intenta concentrar las asignaturas de mayor dificultad en el centro de la jornada, ya que es la ventana de tiempo en la que los estudiantes presentan una mayor capacidad de atención y disposición cognitiva.

El planteamiento inicial del problema supone la existencia de las aulas suficientes para los grupos que se desean programar, es decir, la planta física debe contar mínimo con aulas disponibles y con la capacidad suficiente para albergar a los grupos de alumnos,

cuya cantidad está determinada por las políticas y carácter de la institución. Las aulas para la orientación de actividades que requieren espacios o materiales de trabajo diferentes, son contadas como adicionales y en ningún caso pueden ser asignadas a un grupo de forma permanente durante toda la jornada. De igual forma, aunque los docentes deben cumplir con horarios de trabajos iguales o similares a la jornada de los estudiantes, su carga académica no debe superar el 84% del total de horas de la semana. A la vez, se supone que la cantidad de docentes es suficiente para cumplir con la carga académica necesaria en cada área del conocimiento para la cantidad total de grupos a programar. La asignación de docentes debe contemplar la posibilidad de asociar más de una asignatura a un solo docente. De su parte, ninguna asignatura deberá tener una intensidad superior a las dos horas diarias. En esencia, las consideraciones anteriores definen de forma general las restricciones duras del problema.

Respecto a las restricciones vinculadas a los ritmos cognitivos, se consideran como las restricciones blandas del problema e incrementan considerablemente la complejidad a la hora de buscar una solución válida. No obstante, es importante hacer claridad que una asignación eficiente de acuerdo a los parámetros que establece este enfoque, se encuentra limitada por lo corta e intensiva de la jornada, es decir, no siempre será posible ajustar todas las horas de clase a los ritmos cognitivos esperados, dado que las ventanas temporales adecuadas para ciertas actividades se deben repartir entre diferentes áreas del conocimiento que así lo requieren. De esta manera, se buscará alternativamente que esas horas ideales para actividades intelectuales complejas, se distribuyan de una manera equitativa.

En este sentido, el desarrollo de modelos computacionales ofrece ventajas que permiten abarcar con detalle las múltiples necesidades planteadas, para lo cual el uso de metaheurísticas surge como una respuesta a la complejidad que encierra este problema.

El término metaheurística se obtiene de anteponer a heurística el sufijo meta que significa 'más allá' o 'a un nivel superior'. Las concepciones actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las metaheurísticas son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El término metaheurística apareció por primera vez en el artículo seminal sobre búsqueda tabú de Fred Glover en 1986 (Brito Santana, y otros, 2004). Antes de que el término fuese aceptado completamente por la comunidad científica, estas técnicas eran denominadas heurísticas modernas (Reeves, 1993). Esta clase de algoritmos incluye técnicas como algoritmos genéticos (Chao-Hsien Pan & Huang, 2009) (Manikas & Chang, 2009), lógica difusa (Yun, 2002), búsquedas tabú (Buscher &

Shen, 2009), colonias de hormigas(Xing, Chen, Wang, Zhao, & Xiong, 2010), sistemas artificiales inmune(Ge, Sun, Liang, & Qian, 2008), partículas inteligentes(Eberhart & Kennedy, 1995) y células de aprendizaje autómatas(Jafarpour, Meybodi, & Shiray, 2007). De igual forma existen modelos en los que se han desarrollado sistemas híbridos de estas metaheurísticas(Fang Ming & Qiong, 2009).

De acuerdo a Luna Valero (2008), de las diferentes descripciones de metaheurísticas que se encuentran en la literatura se pueden destacar ciertas propiedades fundamentales que caracterizan a este tipo de métodos:

- Las metaheurísticas son estrategias generales que guían el proceso de búsqueda.
- El objetivo es explorar eficientemente del espacio de búsqueda para encontrar soluciones óptimas.
- Son algoritmos no exactos y generalmente son no deterministas.
- Pueden incorporar mecanismos para evitar óptimos locales del espacio de búsqueda.
- Hacen uso del conocimiento del problema que se trata de resolver en forma de heurísticos específicos que son controlados por la estrategia de más alto nivel.

En síntesis, se pueden definir a las metaheurísticas como estrategias de alto nivel que usan diferentes métodos para explorar el espacio de búsqueda, como si se tratara de una plantilla general no determinista, que es llenada con datos específicos de los problemas (representación de las soluciones, operadores para manipularlas, etc.), los cuales pueden tener espacios de búsqueda de gran tamaño.

En la literatura se pueden encontrar gran variedad de técnicas que han sido diseñadas para la programación de horarios (De Werra, 1985), basadas en modelos como: Profesor clase, pre asignaciones, heurísticas, reglas de decisión, programación de restricciones (Valouxis & Housos, 2003), programación de horarios multiobjetivo (Kieran Burke & Petrovic, 2002), Algoritmos evolutivos (Granada E., Toro Ocampo, & Franco Baquero, 2006) (Pillay & Banzhaf, 2010)(Mejía Caballero & Paternina Arboleda, 2010), metaheurísticas (De Causmaecker, Demeester, & Vanden Berghe, 2009), modelos matemáticos basados en reducción polinómica (Studenovský, 2009), búsqueda tabú (Lü & Hao, 2010), sistemas expertos (Wu, 2011), entre otras técnicas (Kahar & Kendall, 2010)(Turabieh & Abdullah, 2011)(Soza, Landa Becerra, Riff, & Coello Coello, 2011)(Hao & Benlic, 2011).

Es de notar que esta investigación se plantea el diseño de una nueva metodología, basada en metaheurísticas, con el fin de lograr la mejor programación posible de acuerdo a las restricciones

establecidas en función de los ritmos cognitivos, buscando efectos positivos sobre el logro académico de los estudiantes.

1.1 Justificación

En la actualidad, las instituciones de educación básica y media apuntan sus objetivos hacia una formación de alta calidad. Esto implica, no sólo una adecuada administración de sus recursos de infraestructura y medios educativos, sino también, una apropiada planificación de sus procesos académicos de acuerdo a las necesidades que cada una de las áreas del conocimiento exige. En este sentido, un elemento fundamental dentro del proceso académico toma vital importancia: la jornada escolar. Por lo general, la programación horaria de las instituciones se lleva a cabo de manera manual y sólo se limita a distribuir los docentes para cada una de las áreas, evitando cruces entre asignaturas o la doble asignación de docentes. No obstante, este tipo de programación deja de lado el componente cognitivo que de acuerdo a los ritmos biológicos de los individuos, permite un mejor desempeño en el desarrollo de determinadas actividades de aprendizaje a ciertas horas del día.

En este sentido, la implementación de un método de programación para la jornada escolar atendiendo lo descrito anteriormente, no ha sido considerado dentro del ámbito administrativo y operativo de las centros educativos. Sin embargo, esta limitación se puede señalar como un aspecto prioritario, principalmente en las instituciones de carácter oficial, dados los requerimientos sociales para brindar mayor cobertura con las mismas instalaciones, el mismo número de docentes y sin disminuir la calidad en los resultados obtenidos por los estudiantes, en las evaluaciones internas y externas de su proceso de aprendizaje.

Conviene distinguir tres elementos que se conjugan para permitir que los estudiantes tengan un aprendizaje más significativo. En primer lugar, contar con los medios educativos apropiados, es decir, tener adecuada dotación de salones, laboratorios, salones multimedia y auditorios, que permitan desarrollar las labores académicas con un alto grado de satisfacción; lo segundo, es contar con una planta docente idónea y suficiente para cubrir cada una de las áreas del conocimiento. Por último, y en sí el problema abordado por esta investigación, establecer una adecuada distribución de los recursos de infraestructura y del personal docente, en función de los ritmos de aprendizaje que presentan los estudiantes durante la jornada escolar.

De esta manera, la presente investigación busca a través de la implementación de técnicas inteligentes, ofrecer una alternativa de solución al problema de asignación de horarios de clase en instituciones de educación básica y media, a partir de un enfoque

en los ritmos cognitivos, estableciendo como objetivo primordial, realizar una programación pertinente de acuerdo a la dificultad intelectual y el grado de nivel atencional requerido por cada asignatura, de manera que su implementación permita minimizar la mortalidad académica, principalmente en las áreas que requieren un mayor esfuerzo intelectual y consecuentemente reducir la tasa de repitencia en las instituciones.

Los resultados esperados de este trabajo tienen importancia no sólo a nivel conceptual, sino también en el contexto interno y externo de las instituciones. A nivel interno, el modelo de programación horaria propuesto constituye un elemento adicional a los diferentes modelos pedagógicos actuales, complementando la eficiencia en los procesos de apropiación del conocimiento. Establecer mecanismos que favorezcan los ritmos cognitivos, permite que las áreas en las que existen mayores dificultades de aprendizaje, puedan concentrarse de una manera equitativa en las ventanas de tiempo en las que es posible obtener una mejor respuesta intelectual por parte de los estudiantes. A nivel externo se pueden observar dos consecuencias. La primera, son los resultados de las instituciones en las pruebas externas, especialmente en las áreas clave del saber (MEN, 2012), lo cual además de demostrar un mejor rendimiento y calidad de la educación, se constituye en un valor agregado con el que los estudiantes puedan competir para el ingreso a las instituciones de educación superior. La segunda, es el costo por estudiante en las instituciones públicas, pues la mortalidad académica y consecuentemente la repitencia de años lectivos de un estudiante, representan un valor adicional dentro del presupuesto o en su defecto la negación de un cupo para un nuevo alumno. Además, la reducción de gastos por estudiante o la ampliación de cobertura es un beneficio significativo no sólo para la institución, sino también para la sociedad. De esta manera, sí para el 2009 la cantidad de alumnos matriculados a nivel nacional era de 11'219.097 y el total de inversión realizado por el Gobierno para educación básica y media fue de 24.83 billones de pesos (MEN, 2012), se podría estimar un costo neto anual aproximado por estudiante de \$2'213.190. Por tanto, dado que la tasa de repitencia para este mismo año fue de alrededor del 3%, podríamos decir, que si se rebajara en un punto porcentual dicha tasa, el gobierno tendría un presupuesto aproximado de \$248.300.000.000, para otras inversiones.

1.2 Objetivo general

Diseñar una metodología para el uso de técnicas inteligentes en la asignación de horarios de clase en instituciones de educación básica y media, a partir de un enfoque en los ritmos cognitivos que presentan los estudiantes durante la jornada escolar, con el

propósito de que su implementación permita minimizar la mortalidad académica en las áreas que requieren un mayor esfuerzo intelectual y consecuentemente tratar de reducir la tasa de repitencia en las instituciones.

1.3 Objetivos específicos

1. Realizar una revisión bibliográfica sobre el estado del arte en la solución del problema de asignación de horarios en los diferentes escenarios de aplicación de este tipo de secuenciación, con el fin de determinar las principales técnicas inteligentes que existen para la solución de este problema.
2. Realizar una revisión bibliográfica para explicar la forma en que los hábitos horarios y más específicamente, la jornada escolar, están directamente relacionados con el desempeño de los estudiantes, y la forma cómo a través de una asignación de horarios en la que se consideren los ritmos de eficiencia cognitiva, es posible que aspectos como la memoria y la atención se potencialicen, propiciando una mayor disposición para el aprendizaje.
3. Construir un modelo de codificación de las múltiples variables implícitas en el problema, desarrollar un método de solución para el problema e implementar un software en la cual se realice la asignación de horarios de clase de acuerdo a las restricciones establecidas por los ritmos cognitivos de los estudiantes.
4. Realizar una prueba piloto de la metodología sobre los alumnos de una institución educativa de la ciudad de Manizales, haciendo un diagnóstico de sus hábitos horarios y cualidades cognitivas, para posteriormente aplicar una reforma en su jornada escolar de acuerdo al algoritmo implementado.
5. Comparar los resultados que obtienen los estudiantes con una jornada escolar de acuerdo al enfoque en ritmos cognitivos, con los resultados obtenidos por un grupo del mismo nivel bajo una asignación basada en criterios generales, para establecer la efectividad de la metodología, principalmente en la minimización de la mortalidad académica en las áreas que requieren mayor esfuerzo intelectual y en la consecuentemente reducción de la tasa de repitencia.

1.4 Hipótesis

El uso de técnicas inteligentes en la implementación de la solución del problema de asignación de horarios en instituciones de educación básica y media, permitirá realizar una programación pertinente de acuerdo a la dificultad intelectual y el grado de nivel atencional requerido por cada asignatura, de manera que se pueda valorar la forma cómo la jornada escolar está directamente relacionada con el desempeño de los estudiantes, a fin de

establecer si es posible minimizar la mortalidad académica, principalmente en las áreas que requieren un mayor esfuerzo intelectual y consecuentemente reducir la tasa de repitencia en las instituciones.

1.5 Resultados esperados

Como resultado de esta investigación se espera:

1. Identificar las principales técnicas utilizadas para la solución del problema de asignación de horarios en los diferentes escenarios de aplicación de este tipo de secuenciación, con el fin de seleccionar aquella que se adecue mejor a las características y complejidad del problema.
2. Explicar la forma en que los hábitos horarios, y más específicamente, la jornada escolar, están directamente relacionados con el desempeño de los estudiantes, y la forma cómo a través de una asignación de horarios en la que se consideren los ritmos de eficiencia cognitiva, es posible que aspectos como la memoria y la atención se potencialicen, propiciando una mayor disposición para el aprendizaje.
3. Presentar un modelo de codificación de las múltiples variables consideradas y la propuesta metodológica para la solución del problema, a través de una aplicación que realice la asignación de horarios de clase de acuerdo a las restricciones establecidas por los ritmos cognitivos de los estudiantes.
4. Presentar la evaluación de los resultados académicos obtenidos por los estudiantes de la institución educativa de la ciudad de Manizales en la que se realizó la prueba experimental, a través del establecimiento de una jornada escolar de acuerdo al modelo que expuesto por la presente investigación.
5. Establecer sí los resultados académicos de los estudiantes con una jornada escolar de acuerdo al enfoque en ritmos cognitivos son o no superiores, a los resultados obtenidos por un grupo del mismo nivel bajo una asignación aleatoria realizada con un software comercial, dentro de la misma institución. De esta manera, determinar la efectividad de la metodología, principalmente en la minimización de la mortalidad académica en las áreas que requieren mayor esfuerzo intelectual y en la consecuentemente reducción de la tasa de repitencia.

1.6 Criterios de éxito

La presente investigación hace uso de técnicas inteligentes en la implementación de una solución al problema de asignación de horarios en instituciones de educación básica y media, con la que se busca obtener una programación pertinente de acuerdo a la dificultad intelectual y el grado de nivel atencional requerido

por cada asignatura. Como consecuencia de dicha programación, se espera que los estudiantes tengan un mejor desempeño y se reduzcan así, los índices de mortalidad académica y consecuentemente la tasa de repitencia en las instituciones. No obstante, el proceso educativo está sustentado en un conjunto integrado de elementos físicos, psicológicos, ambientales, culturales, económicos, sociales y emocionales, los cuales tienen una alta influencia en la efectividad del modelo planteado. De esta manera, los alcances a nivel práctico están condicionados a la población donde se efectuó la prueba piloto, que para este caso es una institución educativa del municipio de Manizales, que atiende a 640 estudiantes, entre los 10 y los 19 años, provenientes de familias en estratos 1, 2 y 3 principalmente.

1.7 Metodología de la investigación

En el desarrollo de este proyecto se combinarán diferentes tipos de investigación. La fase inicial, que es una investigación exploratoria, comprende una revisión del estado del arte en relación a las diferentes técnicas utilizadas para la solución del problema de asignación de horarios de clase y a las investigaciones sobre ritmos cognitivos, haciendo énfasis en aquellas vinculadas al contexto colombiano. En este mismo sentido, es necesario estudiar cada una de las diferentes técnicas inteligentes con el fin de establecer su aplicabilidad y efectividad para este problema en particular.

La segunda fase, que es una investigación descriptiva, comprende el establecimiento de las variables que hacen parte del problema y de las consideraciones sobre las cuales se van a definir el conjunto de restricciones, con su respectivo grado de cumplimiento (duras o blandas). En este punto, la investigación entra en una etapa correlacional y se debe establecer el modelo de codificación del problema, el cual debe relacionar de manera adecuada cada una de las variables implícitas. Seguidamente, se debe dar paso a la fase de implementación, en la cual es indispensable el desarrollo de una aplicación sobre la cual se ejecuten las técnicas inteligentes seleccionadas de acuerdo al modelo de codificación desarrollado. Este aplicativo debe estar en capacidad de dar solución al problema planteado de una manera óptima y cumpliendo a cabalidad con las restricciones duras impuestas y minimizando el descarte o violación de las restricciones blandas. Como última etapa de la investigación, se realizará una prueba piloto de los resultados que entrega la aplicación de la metodología, sobre un grupo de estudiantes de una institución de educativa de la ciudad de Manizales.

En síntesis, este proyecto de investigación implicará el desarrollo de las siguientes actividades:

1. Realizar una revisión bibliográfica sobre el estado del arte en la solución del problema de asignación de horarios en los diferentes escenarios de aplicación de este tipo de secuenciación, con el fin de determinar las principales técnicas inteligentes que existen para la solución de este problema.
2. Explicar la forma en que los hábitos horarios, y más específicamente, la jornada escolar, están directamente relacionados con el desempeño de los estudiantes, y la forma cómo a través de una asignación de horarios en la que se consideren los ritmos de eficiencia cognitiva, es posible que aspectos como la memoria y la atención se potencialicen, propiciando una mayor disposición para el aprendizaje.
3. Construir un modelo de codificación de las múltiples variables implícitas en el problema e implementar un software en el cual se realice la asignación de horarios de clase de acuerdo a las restricciones establecidas por los ritmos cognitivos de los estudiantes.
4. Realizar una prueba de la metodología sobre los alumnos de una institución educativa de la ciudad de Manizales, en la que se establezca la jornada escolar de acuerdo al modelo que desarrolla la presente investigación y poder así evaluar los resultados académicos obtenidos por los estudiantes.
5. Comparar los resultados que obtienen los estudiantes con una jornada escolar de acuerdo al enfoque en ritmos cognitivos, con los resultados obtenidos por un grupo del mismo nivel bajo una asignación aleatoria realizada con un software comercial, dentro de la misma institución, para establecer la efectividad de la metodología, principalmente en la minimización de la mortalidad académica, principalmente en las áreas que requieren mayor esfuerzo intelectual y en la consecuentemente reducción de la tasa de repitencia.

2. Marco teórico

La cotidianidad nos enfrenta con diferentes problemas relacionados con la secuenciación de elementos o eventos de distinta índole, sobre los cuáles en su mayoría desconocemos su forma de organización o método de programación. Cada uno de ellos supone ubicar un instrumento (individuos u objetos) en un sitio determinado, en una hora específica, bajo unas condiciones prescritas. Ejemplos de esto son las rutas de transporte público, transporte de carga, vuelos aéreos, horarios de trabajo, horarios para el uso de espacios como escenarios deportivos o salas de cirugía y horarios de clases, entre otros (Caballero Rosas & Gómez Cardona, 2004).

Dado que no es un problema general, sino que está supeditado a las restricciones que impone cada contexto, su solución se enfatiza de manera particular a un grupo o proyecto, buscando la optimización de recursos y una adecuada utilización de los tiempos. De esta manera, las diferentes investigaciones no establecen una metodología o una técnica única para solucionar el problema, sino que ofrecen diversas posibilidades en la cobertura de parámetros o en la eficiencia para la obtención de soluciones válidas.

2.1 Tipos de programación

Según Wren (1996), el problema de programación de horarios consiste en la disposición de objetos dentro de un patrón en el tiempo o el espacio, de tal manera que se logre el cumplimiento de unos objetivos determinados y que las restricciones sobre la forma en que se deben disponer dichos objetos se satisfagan. Es así como se identifican tres elementos esenciales dentro de cualquier proceso de programación: objetos, patrones y restricciones. Los objetos se refieren al recurso a organizar o disponer, los cuales pueden ser personas, máquinas, vehículos, clases, trabajos en una fábrica, etc.; el patrón es el orden que deben seguir los eventos y se crea como parte del proceso de programación; y las restricciones se definen como las relaciones tangibles o intangibles entre objetos y representan las reglas que se deben seguir para la construcción de un patrón.

Con base en lo anterior, se diferencian tres tipos de problemas relacionados con la programación, los cuales se describen de la siguiente manera:

- **Asignación de horarios (timetabling):** Es la asignación de unos recursos a un objeto dado, en un tiempo determinado, sujeta a restricciones que permitan el cumplimiento de unos objetivos específicos.
- **Asignación de turnos (rostering):** Es la colocación de unos recursos en determinados periodos siguiendo un patrón, de acuerdo a unas restricciones dadas.
- **Secuenciación (sequencing):** Es la construcción de un orden en el cual unas actividades deben ser llevadas a cabo o en el que unos objetos deben ser colocados para dar una solución deseada, de acuerdo a unas restricciones establecidas.

Para este caso particular, la investigación se concentra en el problema de asignación de horarios, el cual, aunque es aplicable a problemas de diversa índole, para el presente trabajo se ha enfocado en la programación de horarios escolares.

2.2 Asignación de horarios educativos

En el contexto educativo, el problema de asignación de horarios se enmarca en la disposición de asignaturas, estudiantes, profesores y salones, dentro de periodos de tiempo establecidos (jornadas, semanas, trimestres, semestres), los cuales están sujetos a unas condiciones instauradas de acuerdo a las características de la organización (colegios, universidades) y al tipo de evento que se desea programar (jornada, exámenes, uso de instalaciones).

Se pueden distinguir tres diferentes enfoques del problema: asignación de horarios escolares (school timetabling), asignación de horarios universitarios (course timetabling) y asignación de horarios de exámenes (examination timetabling), los cuales se describen a continuación (Franco Baquero, Toro Ocampo, & Gallego Rendón, 2008).

2.2.1 Asignación de horarios escolares

Esta programación es propia de escuelas y colegios. El problema consiste en distribuir los tiempos de una jornada diaria, que por lo general es continua y planeada para un horizonte de tiempo de una semana. En este caso se deben ubicar las asignaturas a los períodos de tiempo disponibles, de tal manera que ningún profesor tenga más de una sesión en el mismo período y que todas las sesiones de la asignatura estén presentes en el horario. Algunas variaciones de este tipo de problema son:

- La presencia de bloques de materias, es decir, que existan sesiones simultáneas de una misma asignatura.

- La posibilidad que un profesor pueda dictar más de una asignatura.
- La posibilidad de hacer uso de salas especiales (audiovisuales, sistemas) distintas a los salones asignados de forma permanente a un grupo.
- La posibilidad de que el número de asignaturas pueda ser superior para algunas jornadas dentro de algunos grupos.

En una proporción alta de instituciones, la solución a este problema es construida de manera manual, lo cual además de tomar días o semanas de trabajo, no permite establecer cierto tipo de restricciones que optimicen el uso del recurso humano y de infraestructura, al igual que da lugar a errores por traslapes en asignaturas y en la asignación de los docentes (Flores Pichardo, 2011). Sin embargo, a nivel investigativo se han aplicado diversas técnicas analíticas y heurísticas a la solución del problema de programación de horarios escolares tales como: grafos coloreados (Čangalović & Schreuder, 1991), recocido simulado (Abramson D. , 1991), algoritmos genéticos (Colorni, Dorigo, y Maniezzo, 1998; Abramson y Abela, 1992; Beligiannis, Moschopoulos, y Likothanassis, 2009), programación con restricciones (Valouxis & Housos, 2003), algoritmos de búsqueda local (Avella, D'Auria, Salerno, & Vasil'ev, 2007), programación lineal (Birbas, Daskalaki, & Housos, 2009), búsqueda tabú (Minh, Thanh, Trang, & Hue, 2010) y enjambres de partículas (Tassopoulos & Beligiannis, 2012).

2.2.2 Asignación de horarios universitarios

Esta programación es propia de instituciones universitarias o de formación superior. El problema consiste en organizar un horario para las sesiones de un conjunto de asignaturas, considerando un número determinado de salas y bloques de tiempo. En este enfoque, el estudiante escoge las asignaturas de acuerdo a sus necesidades y a su nivel de desempeño, sin la restricción de que todo un grupo deba tomar las mismas asignaturas. Algunas variaciones de este tipo de problema son:

- La asignación de periodos de tiempo variables a cada una de las asignaturas.
- La disponibilidad de salones sólo en un horario determinado.
- La disponibilidad de un docente sólo en un horario determinado.
- La regulación de no cruce de horarios entre asignaturas pertenecientes a un mismo periodo académico.
- La asignación de horarios de clase a asignaturas de un mismo periodo académico atendiendo a situaciones de localización (existencia de más de un campus universitario).
- La asignación de diversos profesores por subperíodos de tiempo a una misma asignatura.

- La asignación de salones en consideración a la cantidad de estudiantes inscritos para cada asignatura.

En la literatura se pueden encontrar investigaciones relacionadas con este tipo de programación como las llevadas a cabo por (Burke, Elliman, & Weare, 1994), (Lewis, Paechter, & Rossi-Doria, 2007), (Dammak, Elloumi, Kamoun, & Ferland, 2008), (Hernández, Miranda P, & Rey, 2008), (De Causmaecker, Demeester, & Vanden Berghe, 2009), (Mejía Caballero & Paternina Arboleda, 2010), (Al-Betar & Khader, 2012), (Abdullah, Turabieh, McCollum, & McMullan, 2012).

2.2.3 Asignación de horarios de exámenes

Esta programación está vinculada principalmente a instituciones universitarias. El problema consiste en programar el horario para los exámenes, determinando la cantidad de salas y tiempo para realizar la evaluación de conocimientos de los alumnos que cursan una asignatura. Algunas variaciones de este tipo de problema son:

- La asignación del salón en consideración a la cantidad de estudiantes que presentarán el examen.
- La asignación de periodos de tiempo variables a los exámenes de cada una de las asignaturas.

En la literatura se pueden encontrar investigaciones relacionadas con este tipo de programación como las llevadas a cabo por (Cheong, Tan, & Veeravalli, 2009), (Pillay & Banzhaf, 2010), (Burke, Kendall, Mısır, & Özcan, 2010), (Kahar & Kendall, 2010), (Mansour, Isahakian, & Ghalayini, 2011), (Ayob, et al., 2011), (Abdul-Rahman, Burke, Bargiela, McCollum, & Özcan, 2011), (Sabar, Ayob, Qu, & Kendall, 2011), (Turabieh & Abdullah, 2011), (Demeester, Bilgin, De Causmaecker, & Vanden Berghe, 2012), (McCollum, McMullan, Parkes, Burke, & Qu, 2012), (Gogos, Alefragis, & Housos, 2012), (Burke, Pham, Qu, & Yellen, 2012)

2.3 Métodos de solución

A nivel computacional, la programación de horarios se encuentra situada en la categoría de problemas de optimización. En este sentido, la presente sección se definen los conceptos asociados a este tipo de problemas y las principales técnicas empleadas en su solución, dando un enfoque en los horarios escolares que son el eje sobre el cual se desarrolla la presente investigación.

2.3.1 Concepto general de optimización

De manera general, se puede decir que optimizar significa buscar la mejor manera de realizar una actividad que tiene diferentes soluciones válidas. En el contexto científico, la optimización constituye un proceso que trata de encontrar la mejor solución posible para un determinado problema. El procedimiento que lleva a cabo la búsqueda es denominado programa. En un programa de optimización existen tres elementos importantes a considerar (Schweickardt & Miranda, 2010):

- Un conjunto de posibles soluciones o soluciones válidas, el cual es conocido como espacio de búsqueda.
- Una técnica de búsqueda, la cual puede ser un algoritmo o un modelo de solución.
- Un criterio para determinar cuál es la mejor solución dentro del espacio de búsqueda, para lo que se establece una función objetivo o función de evaluación.

De acuerdo a lo anterior, el planteamiento del problema supone el hallazgo de los valores para los que una determinada función de evaluación (problema mono-objetivo) o un conjunto de funciones de evaluación (problema multi-objetivo), obtiene su valor máximo o mínimo, según se desee. En este sentido, dichas funciones de evaluación, contienen implícitas las restricciones que determinan la validez de una solución y su nivel de eficiencia.

Muchos de los problema de optimización tratan de platearse de manera lineal, ya que permiten ser resueltos mediante métodos que han sido ampliamente probados en diferentes áreas del conocimiento (Urrutia Galicia, Alcérreca Huerta, & Ordaz Alcántara, 2008). No obstante, cuando los problemas no pueden linealizarse, su solución se considera compleja, aludiendo al contexto de la complejidad computacional, el cual está vinculado a la optimización combinatoria. Buscar una solución eficiente a este tipo de problemas, ha dado como resultado el desarrollo de diversos procedimientos en los que es posible la consecución de soluciones, que aunque no alcanzan el máximo valor de optimización, permiten obtener valores muy cercanos, reduciendo la carga computacional que representaría la exploración de todo el espacio de búsqueda. Estos procedimientos, conocidos como metaheurísticas, se basan en el sentido común, permitiendo hallar soluciones a problemas complejos, de forma rápida y eficiente. Las metaheurísticas se constituyen en un proceso repetitivo de una heurística, donde la solución hallada mediante ésta, es modificada en función del cumplimiento de las restricciones establecidas y del valor esperado en la función de evaluación (Mejía Caballero J. M., 2008). Sobre esta idea, se sustentan los aportes propuestos en el presente trabajo.

- A. Problemas de optimización combinatoria:** El problema de asignación de horarios es considerado un problema de optimización combinatoria. La teoría de la complejidad computacional, como parte de la teoría de la computación, estudia los recursos requeridos por un algoritmo para resolver un problema de este tipo. Los recursos más estudiados son el tiempo, que se traduce en el número de pasos de ejecución de un algoritmo para resolver el problema y el espacio, que es la cantidad de memoria utilizada para resolver el problema. En este sentido, se pueden clasificar los problemas en tres clases: P, NP y NP-C (Chávez Bosquez, De los Santos Torres, & Gómez Ramos, 2005).
- **Problemas P:** son el conjunto de problemas que se conocen como polinomiales. Un problema polinomial es aquel que puede ser comprendido y resuelto mediante la descomposición del problema en los términos que lo componen. Este tipo de problemas pueden ser resueltos mediante algoritmos eficientes, en un tiempo polinomial, es decir, un tiempo que es susceptible de ser dividido en años, meses, semanas o horas. Esta definición permite observar que la mayor parte de los problemas tratables tienen soluciones prácticas (Maldonado & Gómez Cruz, 2010).
 - **Problema NP:** son todos aquellos problemas no polinomiales, es decir, aquellos problemas para los que no existe un algoritmo determinista y cuyo tiempo de ejecución no es polinomial, respecto al tamaño de los datos de entrada. La importancia de esta clase de problemas de decisión es que tienen un espacio de búsqueda extenso para el que se desea saber si existe una cierta solución o si existe una mejor solución que las conocidas. Por tanto, se requiere de algoritmos no convencionales mediante los cuales sea posible obtener una solución en tiempos igualmente polinomiales, aunque de forma aproximada (Maldonado & Gómez Cruz, 2010).
 - **Problemas NP-Completos:** son todos aquellos problemas NP para los que no es posible encontrar soluciones óptimas para instancias de gran tamaño en un tiempo de cálculo razonable (Montoya-Torres, Aponte Penagos, & Rosas Castro, 2010). Son los problemas más difíciles en NP y muy probablemente no forman parte de la clase de complejidad P. Demostrar que un problema es NP-completo, requiere comprobar que no tiene una solución determinista en tiempo polinomial, salvo que todos los problemas de NP estén en P, algo que aún no está demostrado (Ibarra Florencio, 2011). Un algoritmo de fuerza bruta para problemas NP-Completos utiliza tiempo exponencial con respecto al tamaño de la entrada. Se desconoce si hay mejores algoritmos, por lo cual, para resolver un problema NP-Completo de tamaño arbitrario se utiliza uno de los siguientes enfoques:
 - ✓ **Aproximación:** algoritmo que rápidamente encuentra una solución no necesariamente óptima con cierto rango de error. Esto se debe, a que en muchos casos, encontrar una buena

aproximación es suficiente para resolver el problema, sin embargo, no todos los problemas NP-Completos tienen buenos algoritmos de aproximación.

- ✓ Probabilidad: se aplica algún tipo de algoritmo probabilístico mediante el cual se obtiene en promedio una buena solución al problema planteado, para una distribución de los datos de entrada dada.
- ✓ Heurísticas: son estrategias de búsqueda basadas en el sentido común, que obtienen soluciones razonables en tiempos relativamente cortos. Dado que se desconoce el óptimo de un problema NP-Completo, estos métodos evolucionan de acuerdo a valores de optimización que se consideran sub-óptimos.

Dadas sus características, el problema de programación de horarios se ubica en la familia de problemas NP-Completos, lo cual representa el elevado nivel de dificultad computacional que representa su solución.

2.3.2 Técnicas para la solución del problema de asignación de horarios

Existen diversas técnicas para la resolución de problemas de asignación de horarios, las cuales se clasifican en dos grandes grupos (Carrasco & Pato, 2001):

A. Técnicas tradicionales: Las técnicas tradicionales, se caracterizan porque recorren todo el espacio de búsqueda, para hallar todas las soluciones posibles del problema abordado, lo cual, aunque es viable, tiene una alta carga computacional y puede invertir mucho tiempo en la medida que se incrementan el número de variables tenidas en cuenta. La ventaja de este tipo de métodos, es que permiten hallar la mejor solución posible, dado que facilita la comparación de todas las opciones válidas en el conjunto de solución, las cuales se pueden ponderar de acuerdo a las restricciones impuestas.

Dentro de las técnicas tradicionales, se destacan los siguientes métodos aplicados al problema de programación de horarios: Programación Entera (Daskalaki, Birbas, & Housos, 2004) (Daskalaki & Birbas, 2005), Programación Lineal (Boland, Hughes, Merlot, & Stuckey, 2008) y Backtracking (Patterson, Talbot, Slowinski, & Weglarz, 1990) (Sadeh, Sycara, & Xiong, 1995), entre otras.

B. Técnicas no tradicionales: Las Técnicas no tradicionales, se caracterizan porque limitan el espacio de búsqueda, de manera tal que no encuentran todas las soluciones posibles, sino un conjunto de soluciones que cumplen con las restricciones establecidas, y que en algunos casos pueden alcanzar el óptimo deseado o un aproximado a este (Mejía Caballero & Paternina Arboleda, 2010). La ventaja de estos métodos está en su alta

efectividad en la consecución de soluciones efectivas con una carga computacional baja y que puede ser resuelta en periodos de tiempo muy cortos.

Dentro de los métodos de solución aplicados al problema de programación de horarios se encuentran principalmente las técnicas inteligentes basadas en heurísticas y metaheurísticas, entre las que se destacan:

- Colonia de hormigas.
- Partículas inteligentes (PSO).
- Recocido simulado.
- Búsqueda tabú.
- Algoritmos evolutivos.

a) Colonia de hormigas: La metaheurística de Colonia de Hormigas (Ant Colony) emplea estrategias inspiradas en el comportamiento colectivo de las colonias de hormigas para descubrir fuentes de alimentación, estableciendo el camino más corto del hormiguero hasta la fuente de alimentación. Cuando las hormigas se mueven inicialmente lo hacen al azar, dejando por el camino una cierta cantidad de feromonas, las siguientes las hacen por el camino que mayor cantidad de feromonas tenga. Se trata de una estrategia de búsqueda distribuida en la que un conjunto de agentes coopera en la búsqueda, transmitiendo e intercambiando información al resto. Aunque la idea inicial parte de los trabajos de M. Dorigo, posteriormente han aparecido otras variantes del método inicial, usando algoritmos de aprendizaje y estrategias de búsqueda local para hallar los mejores recorridos, así como funciones que especifican la intensidad de las feromonas en el tiempo y la persistencia de la información almacenada.

Algunas investigaciones relacionadas con este método de solución y aplicadas al problema de programación de horarios son las siguientes:

- Colonia de hormigas aplicada a la programación óptima de horarios de clase (Peñuela, Franco B, & Toro O, 2008).
- Solving the post enrolment course timetabling problem by ant colony optimization (Nothegger, Mayer, Chwatal, & Raidl, 2012).

b) Partículas Inteligentes (PSO): La Optimización de Partículas Inteligentes (PSO, Particle Swarm Optimization) es una metaheurística evolutiva inspirada en el comportamiento social de las bandadas de pájaros o bancos de peces. Las soluciones, llamadas partículas se echan a volar

en el espacio de búsqueda guiadas por la partícula que mejor solución ha encontrado hasta el momento y que hace de líder de la bandada. Cada partícula evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido y al líder. El procedimiento también tiene en cuenta el mejor valor alcanzado por alguna de las partículas en su entorno. En cada iteración, las partículas modifican su velocidad hacia la mejor solución de su entorno teniendo en cuenta la información del líder.

Algunas investigaciones relacionadas con este método de solución y aplicadas al problema de programación de horarios son las siguientes:

- Using particle swarm optimization to solve effectively the school timetabling problem (Tassopoulos & Beligiannis, 2012).

c) Búsquedas tabú: La búsqueda tabú es una metaheurística que se caracteriza por el uso de memoria adaptativa y de estrategias de aprendizaje para la resolución de problemas. En este sentido la búsqueda tabú hace uso de la historia del proceso atendiendo cuatro dimensiones principales: propiedad de ser reciente, frecuencia, calidad e influencia. Dichas dimensiones facilitan la creación de estructuras lógicas mediante las cuales se evalúan los antecedentes de las diferentes instancias de un problema a través de un enfoque llamado Análisis de objetivo, permitiendo la estructuración de soluciones eficaces (Glover & Melián, 2003).

Algunas investigaciones relacionadas con este método de solución y aplicadas al problema de programación de horarios son las siguientes:

- A Tabu-Search Hyperheuristic for Timetabling and Rostering (Burke, Kendall, & Soubeiga, 2003).
- Adaptive Tabu Search for course timetabling (Lü & Hao, 2010).
- Managing the tabu list length using a fuzzy inference system: an application to examination timetabling (Pais & Amaral, 2012).

d) Algoritmos evolutivos: La mayoría de las implementaciones actuales de algoritmos evolutivos son descendientes de alguno de tres tipos básicos (fuertemente relacionados aunque fueron desarrollados independientemente): algoritmos genéticos, programación evolutiva y estrategias evolutivas. Los algoritmos genéticos fueron propuestos inicialmente por Holland(1975) y estudiados por muchos autores como un modelo genérico de adaptación, pero con un claro acento en aplicaciones de optimización. La programación evolutiva fue

introducida por Fogel (1988) y los trabajos posteriores han seguido su línea inicial de aplicación en dominios donde se intenta crear inteligencia artificial. Esta propuesta inicial se basaba en evolucionar máquinas de estados finitos. Finalmente, las estrategias evolutivas, desarrolladas por Rechenberg (1973) y Schwefel (1965) y objeto actualmente de numerosas extensiones y estudios, fueron diseñadas en un principio para solucionar problemas difíciles de optimización con parámetros discretos y/o continuos (sobre todo en ingeniería)(Alba Torres, 1999).

Dado que la propuesta de la presente investigación se centra en la aplicación de los algoritmos genéticos, a continuación se exponen de manera más amplia sus características.

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Estos se basan en el proceso genético de los organismos vivos, en el que a lo largo de las generaciones, las poblaciones evolucionan en la naturaleza, de acuerdo a los principios de selección natural y la supervivencia de los individuos más fuertes. De esta manera, los AG crean soluciones de un problema determinado, evolucionando hacia valores óptimos, siempre y cuando se haya realizado buena codificación de las variables consideradas a optimizar y el adecuado planteamiento de las funciones de evaluación de los parámetros deseados.

En la naturaleza, los individuos de una población compiten entre sí, por recursos tales como comida, agua y resguardo; aún los miembros de una misma especie se enfrentan en la consecución de un compañero con el cual engendrar un descendiente. Dichos individuos, que tienen más éxito en sobrevivir y reproducirse, tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario, los individuos más débiles producen un menor número de descendientes. Por tanto, los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, facilita que existan descendientes, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven(Moujahid, Inza, & Larrañaga, 2008).

Los AG usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema

dado. A cada individuo se le asigna un valor o puntuación mediante una función de evaluación Fitness, la cual modela adecuadamente las condiciones deseadas como favorables y entrega un resultado que expresa la viabilidad o aptitudes de dicha solución. Esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, es decir, cuanto mayor sea su Fitness, mayor será la probabilidad de que sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos descendientes de los anteriores los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, es decir, cuanto menor sea su Fitness, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se origina una nueva población de posibles soluciones, la cual contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el AG ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

Los AG se caracterizan por ser una técnica robusta, ideal para ser empleadas en diversos problemas para los que el uso de técnicas convencionales de búsqueda implica altos costos computacionales y de procesamiento. Es claro que aunque los AGs no garantizan la solución óptima de los problemas, si permiten el hallazgo de soluciones en un nivel aceptable, en un tiempo justo, teniendo en cuenta que por lo general se trata de problemas de optimización combinatoria (Moujahid, Inza, & Larrañaga, 2008).

Algunas investigaciones relacionadas con este método de solución y aplicadas al problema de programación de horarios son las siguientes:

- A Genetic Algorithm to Solve the Timetabling Problem (Colorni, Dorigo, & Maniezzo, 1990).
- A Parallel Genetic Algorithm for Solving the School Timetabling Problem (Abramson & Abela, 1992).
- A Genetic Algorithm Based University Timetabling System (Burke, Elliman, & Weare, 1994).
- A Genetic Algorithm Solving a Weekly Course-Timetabling Problem (Erben & Keppler, 1996).

- Algoritmo Evolutivo para generar cargas académicas en TIC-SI (Mendoza Crisostomo, Flores Trujillo, & Morales Genis, 2009)
- A genetic algorithm approach to school timetabling (Beligiannis, Moschopoulos, & Likiothanassis, 2009).
- An informed genetic algorithm for the examination timetabling problem (Pillay & Banzhaf, 2010).

2.3.3 Antecedentes del problema en el contexto local

En años recientes existen algunas investigaciones de universidades en Colombia sobre el problema de asignación de horarios y salones. El más reciente es de la Universidad del Norte, en la cual se llevó a cabo una investigación sobre un problema asignación de horarios de clases en el programa de Ingeniería Industrial de la Facultad de Ingeniería de la Universidad de La Guajira, que en ese momento realizaba dicho proceso manualmente bajo algunas restricciones obligatorias y otros parámetros flexibles. En este caso la técnica de solución aplicada fueron los algoritmos evolutivos. El modelo del problema está orientado en obtener una programación de horarios para cada una de las asignaturas que intervienen, considerando cada una de las restricciones que tiene el problema y buscando maximizar el número total de períodos asignados en espacios de tiempo deseados. De manera especial, el modelo también considera que pueden existir asignaciones previas para que algunas asignaturas sean dictadas en períodos y/o salones especiales (Mejía Caballero & Paternina Arboleda, 2010)(Mejía Caballero J. M., 2008).

En la Universidad Tecnológica de Pereira existen dos referentes. El primero propone una metodología para la solución del problema de asignación óptima de salones usando la técnica de optimización Búsqueda Tabú. Esta investigación propone un modelo matemático que muestra las restricciones de forma explícita y compacta. Además, plantean algoritmos constructivos para generar una configuración inicial de buena calidad y se define la estructura de vecindad para la búsqueda local con *Búsqueda Tabú*. El modelo empleado en este trabajo se basa en el utilizado por la comunidad científica del Reino Unido que fue desarrollado para un concurso denominado International Timetabling Competition con el cual se busca resolver el problema University TT, planteando un conjunto de restricciones duras y blandas (Franco Baquero, Toro Ocampo, & Gallego Rendón, 2008).

El segundo referente es una investigación que propone un problema de programación óptima de horarios, planteado bajo las condiciones establecidas en la Universidad Tecnológica de Pereira, en el que en 45 bloques de tiempo (5 días de 9 horas de clase cada uno) se deben programar un conjunto de asignaturas, salones y estudiantes,

atendiendo a las características que deben ser satisfechas por los salones y requeridas por las clases. El objetivo es elegir un salón y un bloque de tiempo para cada clase de forma que se maximicen las preferencias de los estudiantes sin crear conflictos en la programación de estudiantes o salones. La técnica empleada para resolver este problema se basa en la modificación y adecuación del algoritmo genético propuesto por Chu-Beasley (Granada E., Toro Ocampo, & Franco Baquero, 2006).

El último antecedente es de la Universidad EAFIT, en el que se desarrolló una aplicación computacional para la realización de una programación de exámenes, tomando como modelo el problema de la misma universidad, donde dicha programación se realizaba de forma manual, con un alto grado de complejidad debido a la flexibilización de las carreras y a la creación de nuevos programas de pregrado. Para resolver el problema se implementaron dos técnicas: grafos coloreados y Recocido simulado. La solución obtenida se evaluó realizando una comparación entre esta programación computacional y la programación obtenida por Admisiones y Registro de la Universidad, lo cual demostró la efectividad del modelo (Caballero Rosas & Gómez Cardona, 2004).

2.4 Ritmos cognitivos

La presente investigación busca resolver el problema de asignación de horarios en las instituciones de educación media y básica, a partir de una consideración de los ritmos cognitivos que presentan los estudiantes durante la jornada académica, con el interés de que las asignaturas queden ubicadas en los periodos de tiempo en los cuales los estudiantes tienen una mejor disposición para realizar actividades con una mayor demanda cognitiva.

2.4.1 La cronobiología y cronopsicología

La primera evidencia de la estructura temporal es que el ser humano es un sistema oscilante tanto si se analiza a nivel biológico como si se analiza a nivel psicológico. Es por ello que surgen dos campos de estudio para indagar la forma como los individuos responden ante estímulos internos y externos durante dichas variaciones temporales: la cronobiología y la cronopsicología (Estaún Ferrer, 2002). La primera estudia de manera general, los ritmos de las funciones corporales incluidas las psicológicas, por lo cual de esta se desprende la segunda, que como su nombre indica, tiene en cuenta la dimensión temporal en el estudio científico de los comportamientos.

La importancia de analizar las funciones psicológicas humanas en función de la variable temporal surge del hecho de que la actividad mental se encuentra sujeta a ciclos de menor o mayor eficacia, lo cual sugiere la necesidad de conocer los ritmos con

los que estos cambios se efectúan y la forma como los individuos responden y se adaptan a ellos.

2.4.2 Factores asociados con la ritmicidad cognitiva

Aunque los seres humanos presentan continuos altibajos en sus ritmos biológicos, los cuales no son siempre comunes para todos los individuos, es posible la identificación de ciertos factores que determinan las variaciones del rendimiento intelectual, de los cuales se destacan (Hederich Martínez, Camargo Uribe, & Reyes Cuervo, 2004):

- Características de la tarea.
 - ✓ Nivel de complejidad.
 - ✓ Componente motor de la misma.
 - ✓ Carga mnésica.
- Diferencias individuales.
 - ✓ Edad.
 - ✓ Sexo.
 - ✓ Cronotipo (factor de matutinidad).
 - ✓ Factores de personalidad.
 - ✓ Estilo cognitivo.
 - ✓ Niveles de motivación.
- Factores socioculturales.

En este mismo sentido, Maciques Rodríguez (2004), establece un criterio similar asociando el proceso cognitivo a nivel escolar a cuatro factores relacionados con las necesidades del alumno: las preferencias emocionales, el ambiente, las necesidades sociales y las necesidades fisiológicas.

No obstante, dadas las características de la presente investigación, el problema se aborda principalmente sobre las diferencias individuales a nivel biológico, las cuales además de vincular las necesidades alimenticias y las necesidades de movimiento del individuo, resalta la existencia de una hora óptima del día para desarrollar un trabajo. Por tanto, conocer dicha hora favorece la programación de horarios particulares para la consecución de rendimientos óptimos en la asimilación de contenidos (De La Cruz Nogales, 2009).

Tal como señala Fernández Enguita (2000), las investigaciones sobre atención y fatiga sugieren que los estudiantes tienen en la mañana un nivel bajo de atención, el cual sube y se mantiene hasta antes del mediodía, momento en el que vuelve a descender. En la tarde el fenómeno se repite, sólo que el proceso de ascenso se ve reflejado sobre las cuatro de la tarde, lo cual implica que en esta parte de la jornada, el nivel más alto de atención se logra

de manera más tardía. Es preciso señalar, que aunque existe un debate alrededor de la implementación de una jornada escolar continua o partida (Feito Alonso, 2007) y sobre la duración de la jornada (Banco Mundial, 2009), el presente trabajo se orienta en la programación de horarios de forma continua, ya sea en la jornada de la mañana o en la tarde, bajo los parámetros establecidos por el Ministerio de educación de Colombia para las instituciones de carácter público (MEN, 2002).

2.4.3 Consideraciones dentro del modelo educativo colombiano

Las consideraciones sobre las cuales se establecieron las restricciones para el desarrollo del modelo desplegado en esta investigación están sustentadas en el trabajo titulado "Ritmos cognitivos en la escuela" (Hederich Martínez, Camargo Uribe, & Reyes Cuervo, 2004), el cual expone información relevante sobre las características cognitivas de los estudiantes y su relación directa con la planificación de la jornada escolar, dentro del modelo educativo colombiano.

Este trabajo muestra los resultados de una investigación llevada a cabo sobre 149 estudiantes en dos instituciones educativas oficiales de los municipios de Bogotá y Tabio, de contexto urbano y rural respectivamente, con jornadas parciales (mañana - tarde). Inicialmente se tipificaron los aspectos más importantes de la vida cotidiana de los estudiantes, desde la perspectiva de la organización de su tiempo en el colegio y su casa. Seguidamente se abordó el problema de los ritmos atencionales, estableciendo la relación existente con los estados de activación fisiológica y las características individuales como el cronotipo y el estilo cognitivo. Por último, se analizaron los ritmos de la memoria, estableciendo algunas consideraciones sobre la capacidad de retención de información y su relación con periodos específicos del día para la memoria a corto y largo plazo.

Como aspectos relevantes a tener en cuenta de acuerdo a los resultados ofrecidos por la investigación en ritmos cognitivos, se pueden destacar las siguientes conjeturas:

- La primera hora de la jornada (mañana - tarde) presenta los niveles más bajos de activación fisiológica y de control atencional, lo cual es totalmente contrario a la creencia de que al inicio de la jornada los estudiantes tienen mayor capacidad de atención y por tanto pueden comprender con facilidad las asignaturas de mayor complejidad.
- Los momentos ideales para el trabajo intelectualmente exigente pertenecen al intermedio de la jornada, en los cuales los niveles de activación fisiológica son elevados y los de control atencional están en sus niveles ideales.

- Las últimas horas de la jornada tampoco favorecen el trabajo en asignaturas con un alto componente atencional o que requieran de una actividad intelectual sostenida, por lo cual se recomienda que en este periodo se ubiquen asignaturas con un alto componente motor, como educación física, manualidades o artes.
- Dada la dificultad existente con la cantidad de horas y la jornadas parciales, se propone que las asignaturas con mayor demanda cognitiva como matemática y lenguaje se ubiquen preferiblemente al intermedio de la jornada, o en su defecto al inicio de la misma, siempre y cuando se privilegien los momentos de presentación de datos durante las primeras horas y los momentos de interpretación y análisis en las segundas.
- La jornada parcial es menos favorable para el logro educativo que la jornada completa. Sin embargo, dada la necesidad de cobertura desde el sector oficial, sería importante que existiera la posibilidad de que los estudiante asistieran a una de las dos jornadas (mañana - tarde) de acuerdo con su cronotipo y sus hábitos horarios regulares en el contexto cultural del que provienen (rural - urbano).

2.4.4 Propuesta acorde al modelo educativo analizado

Con base en las consideraciones de la sección 2.4.3., se propone un modelo de programación de horarios acogiendo no sólo las restricciones generales sobre docentes, asignaturas y salones, sino incluyendo también un acercamiento a la práctica pedagógica, al considerar los momentos de la jornada en los que es más propicio el aprendizaje para ciertas áreas del conocimiento.

En este caso, se estipula una programación horaria semanal de 30 horas, repartida de lunes a viernes en 6 clases diarias de una hora. La jornada diaria se divide en tres segmentos de dos horas de acuerdo al nivel de atención previsto, determinando un periodo de inicio bajo, un periodo intermedio alto y un tercer periodo de descenso. A cada periodo se asocian un conjunto de asignaturas de acuerdo a las características de sus contenidos, por ejemplo al primer bloque, áreas de ciencias naturales y ciencias sociales; al segundo bloque, áreas de ciencias exactas, lenguaje e idiomas; y al tercer bloque, áreas de deportes, tecnología y manualidades. El objetivo de la programación es ubicar la mayor cantidad posible de asignaturas en el segmento horario que les corresponde, ya que esto permite el desarrollo de actividades propias al nivel cognitivo que el estudiante tiene en los diferentes momentos de la jornada.

3. Materiales y métodos

Dado que el problema de programación de horarios es un problema de complejidad NP-Completo y adicionalmente es un problema de satisfacción de restricciones, la dificultad para su óptima solución se incrementa en la medida que se aumentan los parámetros de entrada y las restricciones impuestas. En este sentido, el planteamiento realizado exhibe además de los parámetros normales de entrada (materias, profesores, salones y grupos) y de las restricciones respecto a los cruces de estos, los parámetros necesarios para la programación horaria de acuerdo a los ritmos de eficiencia cognitiva de los estudiantes y sus restricciones asociadas. Por tanto, se trata de un problema con una complejidad superior a los planteamientos generales, razón por la cual las técnicas aplicadas alternan lo robustez y la eficacia, para la consecución de una solución que satisfaga todos los objetivos establecidos.

A continuación se explica el procedimiento desarrollado para la implementación de tres diferente técnicas inteligentes para la solución del problema propuesto. La aplicación de dichas técnicas se enmarcar dentro de un algoritmo que permite la utilización de cualquiera de ellas para la consecución de una solución de un problema determinado, acorde a las restricciones dispuestas.

3.1 Generalidades del algoritmo

El algoritmo desarrollado para la generación de los horarios fue implementado en el software de cálculo matemático MATLAB Versión 7.13.0.564 (R2011b), debido a la orientación de esta herramienta para el manejo eficiente de algoritmos en términos de matrices y vectores, configuración bajo la cual se estructuran los datos para resolver el problema.

El algoritmo se usa a través una interfaz de usuario (GUI), en la cual se selecciona el método de solución deseado y se ingresa la información base del problema mediante un archivo de Excel (*.xlsx).

Después de que se efectúa la solución del problema bajo los parámetros establecidos, se entrega el resultado final al usuario, en un archivo de Excel, el cual contiene el horario de cada grupo,

los docentes asignados a cada hora y la evaluación de la eficiencia de la programación obtenida bajo la métrica que establece el algoritmo.

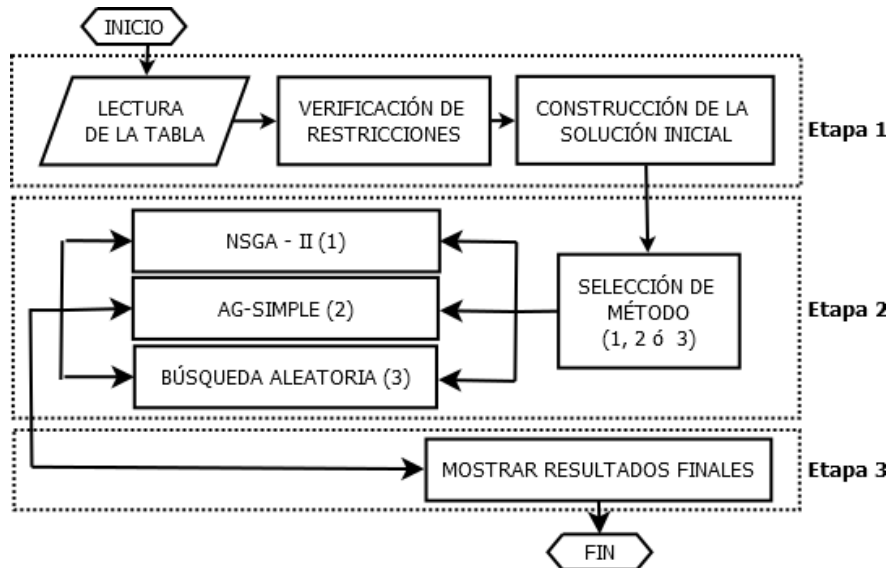
3.2 Estructura general del algoritmo

El algoritmo tiene una estructura general de tres etapas, dos de ellas están vinculadas a la interacción con el usuario y la restante, a la solución del problema bajo los métodos aplicados. Dichas etapas con sus respectivos pasos son:

1. Ingreso y validación de la información.
 - a. Codificación de los datos y construcción de la tabla.
 - b. Ingreso de los datos al algoritmo.
 - c. Verificación de las restricciones de tabla.
 - d. Construcción de la solución inicial.
2. Construcción de la solución de acuerdo al método seleccionado.
3. Presentación de resultados.

La figura 3-1 muestra el diagrama general del algoritmo:

Figura 3-1. Diagrama general del algoritmo.



A continuación se describen cada una de estas etapas.

3.3 Ingreso y validación de la información

Esta etapa se realiza en 3 pasos. El primer paso explica la forma como se organiza la información base que se le suministra al

algoritmo para realizar su ejecución y las características del archivo en que ésta se debe guardar. El segundo paso corresponde a un proceso interno del algoritmo para verificar que la información suministrada cumpla con los parámetros necesarios que permitan ejecutar alguno de los métodos de solución. En el tercer paso el algoritmo construye una solución inicial a partir de la información base, la cual es el punto de partida para cualquiera de los métodos de solución. A continuación se describen cada uno de estos pasos:

3.3.1 Codificación de los datos y construcción de la tabla

Para generar un horario de clase de acuerdo al algoritmo propuesto, es necesario que el usuario suministre la información base, la cual está relacionada con los docentes, su carga académica y las aulas disponibles para cada grupo o asignatura. Para ello, se debe elaborar una tabla con una estructura compuesta por 9 columnas con la siguiente información:

- **Materia:** Nombre de la asignatura.
- **Profesor:** Nombre del profesor.
- **Código materia:** Identificador numérico, único para cada asignatura, indiferente del grupo al que pertenezca.
- **Horas semana:** Intensidad horaria semanal por asignatura.
- **Nombre grupo:** Identificador numérico de cada grupo, de acuerdo al nivel o nomenclatura propia de cada institución (Grado y grupo).
- **Código profesor:** Identificador numérico, único para cada profesor.
- **Código área:** Identificador numérico, asignado a cada asignatura de acuerdo al área del conocimiento a la que pertenece.
- **Salón:** Identificador numérico del salón (o espacio físico) en donde se dicta una asignatura.
- **Número grupo:** Identificador numérico consecutivo de la cantidad de grupos.

Para ilustrar el proceso y explicar los pasos a seguir por el usuario en el ingreso de la información, se tomará la tabla 3-1 como referencia, la cual contiene la información necesaria para generar un horario de tres grupos.

Tabla 3-1. Tabla ejemplo de información base.

Materia	Profesor	Cód. Mat.	Horas Semana	Nombre Grupo	Cód. Prof.	Cód. Área	Salón	Número Grupo
Matemáticas	Profesor 3	1	5	62	3	1	1	1
Ciencias Sociales	Profesor 5	2	3	62	5	4	1	1
Español	Profesor 8	3	5	62	8	2	1	1
Ética y Valores	Profesor 9	4	1	62	9	5	1	1
Religión	Profesor 9	5	1	62	9	5	1	1

Materia	Profesor	Cód. Mat.	Horas Semana	Nombre Grupo	Cód. Prof.	Cód. Área	Salón	Número Grupo
Tecnología	Profesor 10	6	1	62	10	7	1	1
Tecnología-Sala	Profesor 10	7	2	62	10	7	7	1
Ciencias Naturales	Profesor 14	8	4	62	14	3	1	1
Artística	Profesor 15	9	2	62	15	8	1	1
Educación Física	Profesor 18	10	2	62	18	6	1	1
Inglés	Profesor 23	11	4	62	23	2	1	1
Matemáticas	Profesor 3	12	5	72	3	1	2	2
Ciencias Sociales	Profesor 5	13	3	72	5	4	2	2
Ética y Valores	Profesor 5	14	1	72	5	5	2	2
Inglés	Profesor 6	15	4	72	6	2	2	2
Religión	Profesor 9	16	1	72	9	5	2	2
Tecnología	Profesor 10	17	1	72	10	7	2	2
Tecnología-Sala	Profesor 10	18	2	72	10	7	7	2
Ciencias Naturales	Profesor 14	19	4	72	14	3	2	2
Artística	Profesor 15	20	2	72	15	8	2	2
Español	Profesor 8	21	5	72	8	2	2	2
Educación Física	Profesor 19	22	2	72	19	6	2	2
Inglés	Profesor 6	23	4	82	6	2	3	3
Ética y Valores	Profesor 9	24	1	82	9	5	3	3
Religión	Profesor 9	25	1	82	9	5	3	3
Tecnología	Profesor 10	26	1	82	10	7	3	3
Tecnología-Sala	Profesor 10	27	2	82	10	7	7	3
Ciencias Naturales	Profesor 12	28	4	82	12	3	3	3
Algebra	Profesor 13	29	5	82	13	1	3	3
Artística	Profesor 13	30	2	82	13	8	3	3
Español	Profesor 16	31	5	82	16	2	3	3
Ciencias Sociales	Profesor 18	32	3	82	18	4	3	3
Educación Física	Profesor 19	33	2	82	19	6	3	3

Como primera instancia se deben separar las materias por grupos y colocar sus nombres en la primera columna. En la segunda columna se coloca el nombre del profesor que orienta cada materia en el grupo correspondiente. En la tercera columna se debe asignar un código de materia, el cual debe ser diferente para cada asignatura sin importar el grupo. De esta manera, ya que el grupo 1 tiene 11 materias, se asignan códigos de forma ascendente desde el número 1 hasta el número 11; el grupo 2 también tiene 11 materias, entonces se continúa la asignación de códigos desde el número 12 hasta el número 22, y así sucesivamente hasta completar la totalidad de materias de todos los grupos.

Para llenar la columna de Horas Semana basta simplemente con escribir la intensidad horaria de cada materia en su casilla

correspondiente. Así por ejemplo, las asignaturas de código 27 y 28, tienen 2 y 4 horas semanales respectivamente. En la columna Nombre Grupo se colocará de forma numérica el nombre de cada grupo de acuerdo a la nomenclatura o identificador que use cada institución. Por ejemplo el grupo sexto 1 se colocará dentro de la tabla como 61, el grupo octavo 3 como 83, el grupo once 2 como 112 y así, según corresponda.

Cada uno de los profesores de la institución, deben ser codificados con un identificador numérico único. Este identificador es el que debe colocarse en la columna Código Profesor en las asignaturas correspondientes a la carga de cada docente. Por ejemplo, las asignaturas de código 7 y 8, tienen asignados los profesores 10 y 14, respectivamente.

Para llenar la columna de Códigos Área, se debe codificar cada materia de acuerdo al área de conocimiento al que pertenezca. Para ello se utiliza la tabla 3-2 como base para realizar las equivalencias.

Tabla 3-2. Tabla de equivalencias para códigos de área.

CÓDIGO	MATERIAS
Código 1: Ciencias exactas	Matemáticas
	Algebra
	Física
	Calculo
	Geometría
Código 2: Lenguaje e idiomas	Español
	Inglés
Código 3: Ciencias naturales	Biología
	Química
Código 4: Ciencia sociales y filosofía	Sociales
	Geografía
	Ciencias políticas
	Filosofía
Código 5: Educación Religiosa y moral	Historia
	Religión
Código 6: Deportes	Ética
	Educación física
CÓDIGO 7: Tecnología	Danzas
	Electrónica
	Informática
CÓDIGO 8: Manualidades	Tecnología
	Artística

De esta manera, las asignaturas de código 1, 12 y 29 tienen como código de área el número 1, ya que todas pertenecen a la clasificación de Ciencias Exactas; las asignaturas de código 4, 5, 14, 16, 24 y 25 tienen como código de área el número 5, ya que pertenecen a la clasificación de educación religiosa y moral. En caso que el usuario requiera ingresar una asignatura que no se encuentre en el listado, bastará sólo con que realice su

asociación al código de área que mayor relación tenga con las áreas del conocimiento dispuestas.

La columna Salón se llena dando una numeración a los salones y escribiendo en cada fila el número de salón correspondiente en donde se dictará dicha asignatura. Como los horarios escolares tienen una orientación a las clases, las asignaturas de un grupo se orientan en un mismo salón, a excepción de aquellas asignaturas que requieran aulas especiales como informática o deportes, las cuales también deberán estar numeradas para realizar su asignación en el caso que así corresponda. En el ejemplo mostrado existen las aulas 1,2 y 3, asignada a cada una a un grupo diferente. Adicionalmente, está el aula 7, la cual es utilizada para la clase de informática en la que se debe utilizar este recurso.

La columna Número Grupo corresponde a un identificador de cantidad de grupos, el cual sin importar el orden de los grupos en la tabla siempre debe estar en orden ascendente. En el ejemplo este identificador va del 1 al 3.

La tabla 3-1 se debe crear en un archivo de Microsoft Excel, asignándole el nombre que el usuario desee y guardándose bajo la extensión *.xlsx. Aunque en el ejemplo mostrado en la tabla 1, esta contiene los encabezados con los nombres de los datos correspondientes a cada columna, es necesario que esta fila se elimine y sólo se deje la información base. La ubicación del archivo se deja a consideración del usuario.

3.3.2 Evaluación de las restricciones de tabla

Dado que el uso del algoritmo está orientado a instituciones de educación públicas de Colombia, la información base debe ceñirse a dos consideraciones relacionadas con la extensión de la jornada y la disponibilidad máxima de los docente (MEN, 2002), y que en adelante denominaremos *restricciones de tabla*. Ellas son:

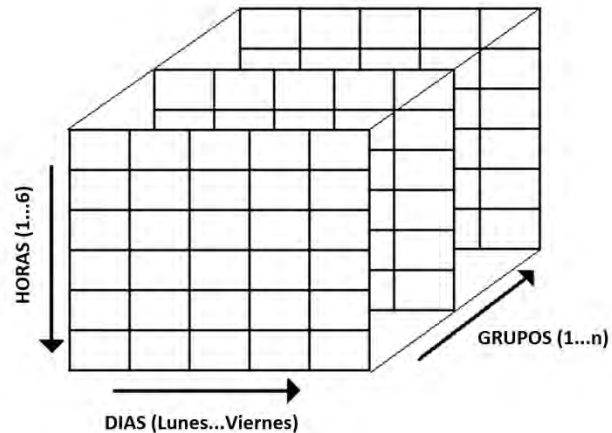
- La sumatoria de las horas de clase del total de materias en cada grupo debe ser igual a 30.
- La sumatoria de las horas de clase que se le pueden asignar a un profesor no debe ser mayor a 25.

Una vez se ingrese la tabla con la información base al algoritmo, se validará el cumplimiento de las restricciones mencionadas. En caso tal de que alguna de ellas se incumpla, el algoritmo detendrá su ejecución y mostrará un mensaje de error, de manera que el usuario verifique la información suministrada.

3.3.3 Construcción de la solución inicial

Una vez ingresada y validada la información base, se procede a la construcción de una solución inicial, las cual corresponde a cuatro hipermatrices (matriz de tres dimensiones) denominadas: Materias, Profesores, Salones y Códigos de Área. En cada una de ellas se plasma la información correspondiente a cada grupo. La figura 3-2 muestra la disposición de dichas hipermatrices.

Figura 3-2. Esquema de hipermatrices.



A continuación se explica el proceso para la generación de la solución inicial tomando como ejemplo la hipermatriz de Materias:

Se genera un vector fila repitiendo el código de cada materia tantas veces como lo indique la intensidad horaria de la asignatura, entendiendo que la cantidad de elementos generados no superará los 30. Por ejemplo, en el grupo 1 la materia 1 tiene una intensidad de 4 horas a la semana, por tanto, se repite el número 1 cuatro veces. El proceso se realiza para todas las materias del grupo siguiendo la misma dinámica. El resultado para los tres grupos es el siguiente:

Grupo 1

```
[1 1 1 1 1 2 2 2 3 3 3 3 3 4 5 6 7 7 8 8 8 8 9 9 10 10 11 11 11 11]
```

Grupo 2

```
[12 12 12 12 12 13 13 13 14 15 15 15 15 16 17 18 18 19 19 19 19 20 20 ...  
... 21 21 21 21 21 22 22]
```

Grupo 3

```
[23 23 23 23 24 25 26 27 27 28 28 28 28 29 29 29 29 29 30 30 31 31 31 ...  
... 31 31 32 32 32 33 33]
```

Con los vectores anteriores se organiza una matriz para cada grupo de acuerdo a la disposición establecida (6 filas, 5 columnas). Esta matriz corresponde a la solución inicial para cada uno de los grupos. Para conformar la hipermatriz, la matriz de cada grupo se coloca en un nivel de profundidad diferente, siguiendo el orden del Número de Grupo. El resultado obtenido es el siguiente:

$$\begin{array}{ccc}
 \begin{array}{c} \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 \\ 3 & 3 & 3 & 4 & 5 \\ 6 & 7 & 7 & 8 & 8 \\ 8 & 8 & 9 & 9 & 10 \\ 10 & 11 & 11 & 11 & 11 \end{array} \right] \\ \text{Matriz grupo 1} \end{array} &
 \begin{array}{c} \left[\begin{array}{ccccc} 12 & 12 & 12 & 12 & 12 \\ 13 & 13 & 13 & 14 & 15 \\ 15 & 15 & 15 & 16 & 17 \\ 18 & 18 & 19 & 19 & 19 \\ 19 & 20 & 20 & 21 & 21 \\ 21 & 21 & 21 & 22 & 22 \end{array} \right] \\ \text{Matriz Grupo 2} \end{array} &
 \begin{array}{c} \left[\begin{array}{ccccc} 23 & 23 & 23 & 23 & 24 \\ 25 & 26 & 27 & 27 & 28 \\ 28 & 28 & 28 & 29 & 29 \\ 29 & 29 & 29 & 30 & 30 \\ 31 & 31 & 31 & 31 & 31 \\ 32 & 32 & 32 & 33 & 33 \end{array} \right] \\ \text{Matriz Grupo 3} \end{array} \\
 \text{Hipermatriz de solución inicial}
 \end{array}$$

Las hipermatrices Profesores, Salones y Códigos de Área se construyen en concordancia con la matriz Materias.

3.4 Construcción de la solución de acuerdo al método seleccionado

El algoritmo desarrollado propone tres métodos de solución para la generación de un horario. Ellos son:

1. NSGA - II (Non-Dominated Sorting Genetic Algorithm - II)
2. Algoritmo genético simple
3. Algoritmo de búsqueda aleatoria

El uso de los diferentes métodos permite comparar la calidad de las soluciones obtenidas, examinando cuál de ellos ofrece un mejor nivel de optimización en sus resultados, de acuerdo a las restricciones impuestas.

Antes de explicar la forma de operación de cada uno de ellos, es necesario definir los elementos bajo los cuales se examina la validez y nivel de optimización de la soluciones. Estos elementos son:

- Tipos de restricciones.
- Funciones de evaluación.

3.4.1 Tipos de restricciones

De acuerdo al planteamiento del problema se consideran dos tipos de restricciones generales: Restricciones Duras (RD) y Restricciones Blandas (Fitness).

A. Restricciones duras: Estas restricciones son de obligatorio cumplimiento, pues están asociadas a la disponibilidad de los

docentes, la carga académica y la eficiencia de procesos pedagógicos. El incumplimiento de alguna de estas restricciones invalida de forma inmediata una solución. Dichas restricciones son:

1. En un mismo día no se deben dictar más de 2 horas de la misma asignatura.
2. Sí en un día se dictan 2 horas de la misma asignatura, éstas deben ser consecutivas.
3. Un profesor no puede dictar 2 materias a la misma hora del día.
4. Dos asignaturas no pueden ser programadas a una misma hora de la jornada en el mismo salón o espacio físico.

B. Restricciones blandas: Las restricciones blandas son un conjunto de condiciones que no son de obligatorio cumplimiento, pero que determinan el nivel de optimización las soluciones encontradas. Estas restricciones están asociadas a los ritmos cognitivos de los estudiantes y se busca obtener un valor mínimo en su evaluación, ya que de esto depende que las asignaturas estén correctamente ajustadas a los momentos en los que la activación atencional de los estudiantes es más adecuada para la realización de una actividad específica. Para ello, la jornada diaria se divide en tres segmentos de dos horas de acuerdo al nivel de atención previsto, determinando un periodo de inicio bajo, un periodo intermedio alto y un tercer periodo de descenso. De acuerdo a esta segmentación, se establecen tres tipos de restricciones:

Restricción 1 - R1: Esta restricción contempla las asignaturas que de acuerdo al tipo de actividades desarrolladas y al nivel atencional requerido por el estudiante, tienen una mejor disposición en la primera y segunda hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 3 y 4.

Restricción 2 - R2: Esta restricción contempla las asignaturas que de acuerdo al tipo de actividades desarrolladas y al nivel atencional requerido por el estudiante, tienen una mejor disposición en la tercera y cuarta hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 1 y 2.

Restricción 3 - R3: Esta restricción contempla las asignaturas que de acuerdo al tipo de actividades desarrolladas y al nivel atencional requerido por el estudiante, tienen una mejor disposición en la quinta y sexta hora del día. Dichas asignaturas son las asociadas a los Códigos de Área 6, 7 y 8.

3.4.2 Funciones de evaluación

Se desarrolló el siguiente modelo matemático para la evaluación de los dos tipos de restricciones en las soluciones obtenidas por cualquiera de los métodos de solución.

Para las restricciones duras, el proceso se efectúa sobre las hipermatrices de Materias, Profesores y Salones. La ecuación 3-1 representa la función de evaluación de estas restricciones:

$$RD = F_1 + F_2 \quad (3-1)$$

Los valores de F_1 y F_2 se describen en las ecuaciones 3-2 y 3-3.

$$F_1 = \sum_{i=1}^{NG} \sum_{j=1}^5 F_{Rest1}(\text{Materias}) \quad (3-2)$$

$$F_2 = \sum_{j=1}^5 \sum_{k=1}^6 F_{Rest2}(\text{Profesores}) + F_{Rest2}(\text{Salones}) \quad (3-3)$$

Donde:

i = variable para recorrer la cantidad total de grupos a programar.

j = variable para recorrer los días de la semana.

k = variable para recorrer las horas del día.

NG= cantidad total de grupos a programar.

La evaluación F_{Rest1} y F_{Rest2} describe en las ecuaciones 3-4 y 3-5:

$$F_{Rest1} = \begin{cases} 1 & \text{Si algún código se repite más de 2 veces en la columna } j \\ 1 & \text{Si al repetirse algún código 2 veces en la columna } j, \text{ estos no están} \\ & \text{consecutivos} \\ 0 & \text{En otro caso} \end{cases} \quad (3-4)$$

$$F_{Rest2} = \begin{cases} 1 & \text{Si se repite algún elemento en el vector de profundidad} \\ & \text{determinado por } (j, k) \\ 0 & \text{En otro caso} \end{cases} \quad (3-5)$$

Es importante tener en cuenta que la solución no es válida en caso de que la función objetivo RD sea diferente de cero.

Para las restricciones blandas, el proceso se efectúa sobre las hipermatriz de Códigos de Área. La ecuación 3-6 representa la función de evaluación de estas restricciones:

$$\text{Fitness} = \min \left(\sum_{i=1}^{NG} \sum_{j=1}^5 \sum_{k=1}^6 F_{\text{Rest}3}(\text{Códigos de Área}) \right) \quad (3-6)$$

Donde $F_{\text{Rest}3}$ evalúa una restricción por iteración, dependiendo del código de área. En la tabla 3-3, se pueden observar los valores de penalización que se tienen para cada restricción de acuerdo a la ubicación de cada asignatura según la hora del día.

Tabla 3-3: Valores de penalización restricciones blandas.

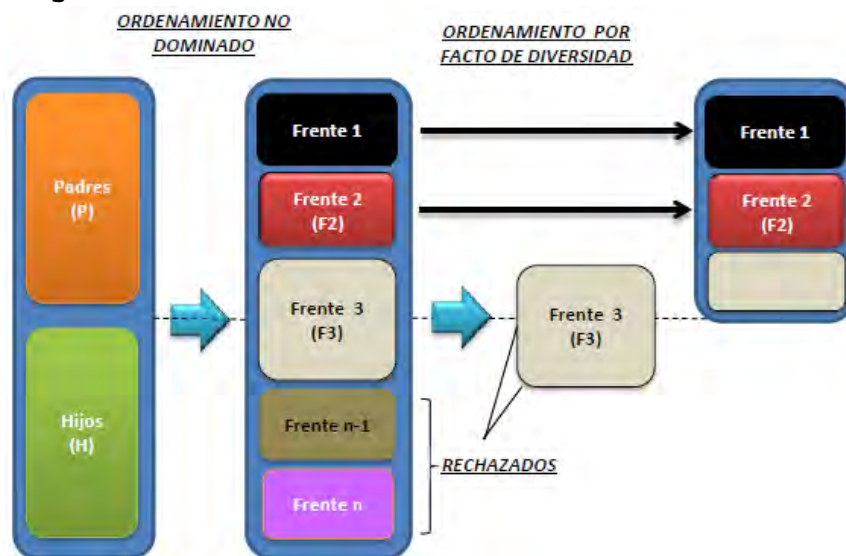
Hora de la jornada	Valor Penalización Restricción 1	Valor Penalización Restricción 2	Valor Penalización Restricción 3
1	0	6	3
2	0	3	3
3	3	0	6
4	3	0	6
5	6	3	0
6	6	6	0

3.4.3 Método de solución 1: Algoritmo genético NSGA-II

El NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm) es un algoritmo de optimización multiobjetivo con elitismo. Tal como se plantea en esta investigación, el problema de la generación de horarios es un problema de optimización multiobjetivo, el cual se centra sobre dos funciones objetivo: las funciones de evaluación RD y Fitness. Se busca que la función RD siempre sea igual a cero y que la función Fitness tenga el menor valor. Por tanto, se puede considerar como orientación específica de la optimización, minimizar ambas funciones.

El mecanismo de evolución del NSGA-II se ilustra en la figura 3-3.

Figura 3-3. Mecanismo de evolución del NSGA-II.

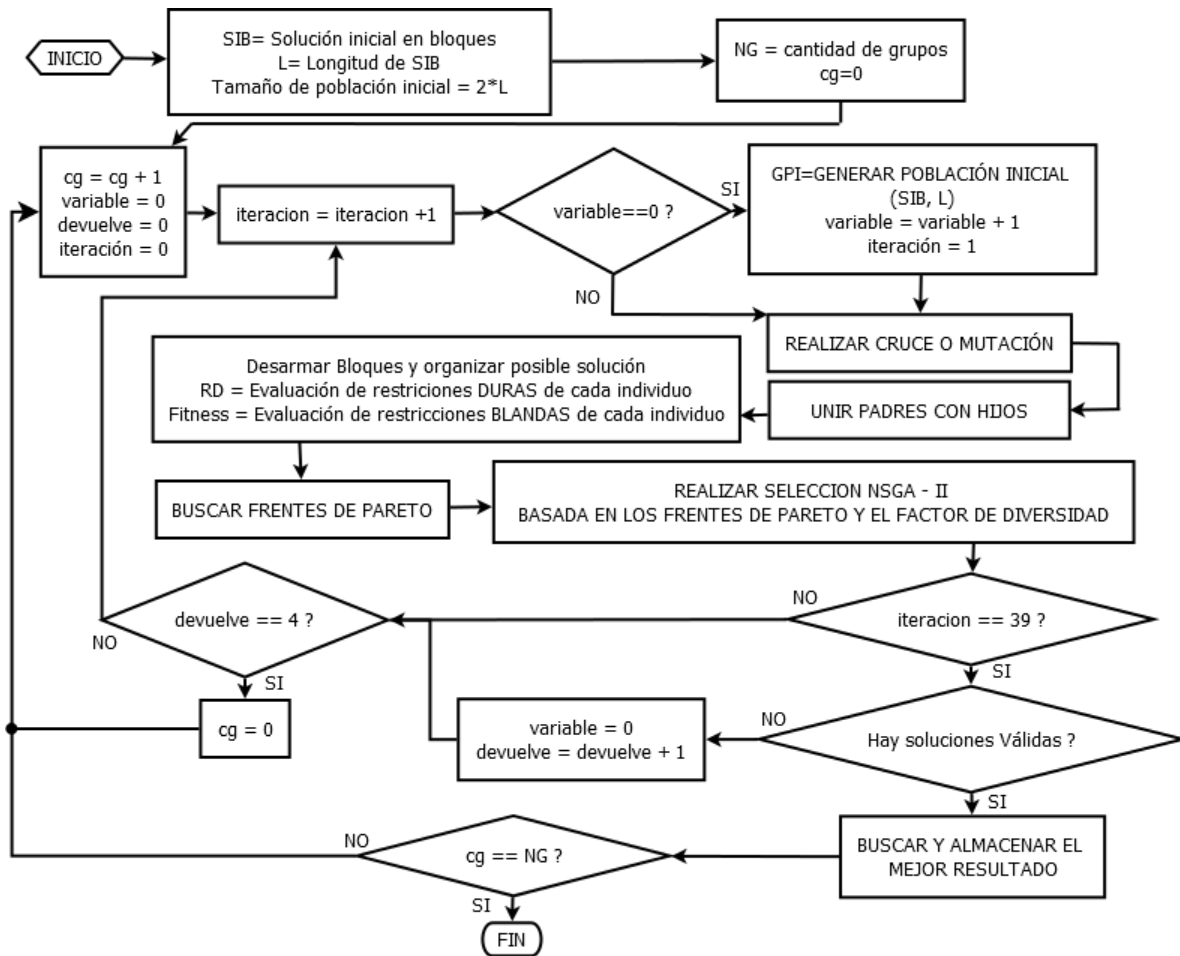


Adaptado de (Correa Flórez, Bolaños, & Molina Cabrera, 2008).

En este algoritmo se parte de una población de N de individuos llamados "Padres", a los cuales se les aplican los operadores genéticos de cruce y mutación, para hallar N nuevos individuos llamado "Hijos". Posteriormente se crea una comunidad en la que se reúne a Padres e Hijos, previa verificación de que no exista ningún individuo repetido en ella. Seguidamente se realiza la evaluación de las restricciones duras y blandas del problema para cada individuo de la comunidad, resultado que se utiliza para clasificar los individuos en Frentes. De dichos frentes se extraen los mejores individuos, que finalmente conforman la población N que pasa a la siguiente generación.

La solución inicial hallada en la sección 3.3.3 constituye la información con la cual se inicia la ejecución del algoritmo NSGA-II. El proceso general de éste se encuentra descrito en la figura 3-4.

Figura 3-4. Diagrama de flujo de operación del NSGA-II.



El algoritmo obtiene la solución del problema a partir de la construcción de soluciones individuales para cada grupo. Por tanto, además del proceso iterativo para hallar una la solución para un grupo, debe repetir este proceso para la cantidad de grupos existentes

A continuación se presentan los pasos para implementar el algoritmo genético NSGA-II.

A. Codificación en bloques: De acuerdo al planteamiento del problema, la jornada está dividida en tres periodos en los que hay diferentes niveles en el proceso de atención que tienen los estudiantes. Por tal motivo, se estableció que la forma más adecuada de aprovechar dicha distribución era que las materias se organizaran en bloque. Cuando se habla de bloques en un horario se hace referencia a 2 horas consecutivas de la misma materia. Para ello se toman todas las materias de un grupo y se conforman bloques de 2 horas. En caso de que la intensidad horaria de una materia sea impar, la hora que queda suelta se considerará independientemente como un bloque más.

El proceso que se lleva a cabo es el siguiente:

1. Se toma la representación del vector de materias de la solución inicial, la cual tiene la siguiente distribución:

[1 1 1 1 1 2 2 2 3 3 3 3 3 4 5 6 7 7 8 8 8 8 9 9 10 10 11 11 11 11]

2. Como la representación del vector de materias es una distribución ordenada, se van tomado grupos de a dos siempre y cuando las asignaturas sean iguales. En caso de que no se pueda conformar una pareja, se agrega un cero como elemento que ajuste el par. Cada pareja armada se asocia a un elemento de un vector de longitud igual a la cantidad de bloques armados y que se ha llenado de forma ascendente con la cantidad de números correspondiente a dichos bloques. El resultado es el siguiente:

[1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18]																		
[1	1	1	1	1	0	2	2	2	0	3	3	3	3	3	0	4	0	5	0	6	0	7	7	8	8	8	8	9	9	10	10	11	11	11	11]

En el caso de la materia 1, esta tiene un total de 5 horas a la semana. Por tanto se le asignan los bloques 1, 2 y 3, teniendo en cuenta que en el bloque 3 al no poderse completar el par, se agrega un cero para indicar que se trata de una sola hora de dicha materia.

El vector en negrilla es la representación en bloques del vector de materias de la solución inicial. El vector que se encuentra abajo, muestra la correspondencia de dicho vector con las materias que conforman el bloque. La respectiva representación con su correspondencia se guarda para poder realizar posteriormente el proceso inverso.

La representación en bloques del vector de materias es la que se ingresa al algoritmo.

B. Población inicial: Una vez la solución inicial de cada grupo se ha convertido a su respectiva representación en bloques, está se toma para generar la población inicial (Padres) de dicho grupo. Para ello, se establece el parámetro L, el cual representa la cantidad de bloques en la solución inicial, en el cual las horas sueltas se han contabilizado como un bloque más. Este parámetros se utiliza para determinar el tamaño de la población inicial, el cual es igual a $2*L$.

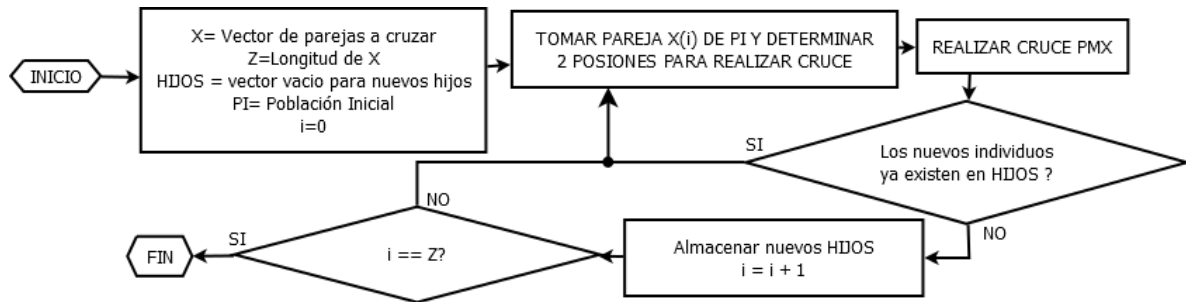
Para generar dicha población se toman la representación en bloques y se halla una distribución aleatoria de números que van de 1 a L, la cual se utiliza para permutar las posiciones de los elementos. El resultado de este proceso se almacena como un nuevo individuo en la población de Padres, previa verificación de que no esté repetido dentro de los individuos ya generados. El proceso se repite hasta alcanzar la cantidad de Padres definida.

A continuación se presenta un ejemplo que muestra cuatro Padres, de los $2*L$ que se generan para cada grupo, los cuales representan 4 individuos de la población inicial.

```
PADRE 1 [6 3 16 11 7 17 14 8 5 15 1 2 4 18 13 9 10 12]
PADRE 2 [14 4 16 17 12 15 10 11 3 7 13 9 8 1 18 5 6 2]
PADRE 3 [4 18 9 15 2 6 5 11 10 12 17 16 1 13 14 7 8 3]
PADRE 4 [1 14 15 6 18 8 16 4 2 9 13 5 10 7 17 11 12 3]
```

C. Operadores genéticos: Los operadores genéticos de cruce y mutación son funciones mediante las cuales se altera la información genética de los individuos de una población (Padres), para generar nuevos individuos (hijos) y mantener así la diversidad genética necesaria para el proceso de evolución. El operador de cruce representa la reproducción sexual y el proceso de mutación representa la mutación biológica. Existen diferentes métodos para aplicar estos operadores, sin embargo, para el problema abordado se tomó la técnica de cruce PMX (Partially Matched Crossover) propuesta por Goldberg y Lingle (1985) y el método de mutación propuesto en la heurística de Lin and Kernighan (1973). A continuación se explican cada uno de ellos.

a) Cruce PMX: La figura 3-5 muestra el proceso del algoritmo para el operador de cruce

Figura 3-5. Proceso para aplicar operador de cruce.

El proceso de cruce conlleva los siguientes pasos:

1. En la primera parte del proceso se genera un vector X el cual contiene las posiciones de los individuos de la población inicial que se van a cruzar. Este es un vector de tamaño L en el que aleatoriamente la posición del elemento en el vector se relaciona con su valor de contenido para conformar una pareja, teniendo en cuenta que el valor máximo de un elemento no será superior a $2 \cdot L$ y que no existe ninguna pareja u elemento repetido en ninguna otra posición del vector. La longitud del vector X determina la cantidad de cruces a realizar.
2. Se toman los dos individuos de acuerdo a las posiciones señaladas en el vector X.
3. Tomando como referencia el valor del parámetro L, se designan dos posiciones aleatorias diferentes, de manera que la posición uno siempre sea menor que la posición dos.
4. Los elementos contenidos entre la posición uno y la posición dos de los dos individuos seleccionados son los elementos que se intercambiarán entre los padres para generar los dos nuevos individuos. Los datos que están por fuera del segmento designado por las posiciones se heredan del padre 1 al hijo 1 y del padre 2 al hijo 2 sin hacer ningún cambio, conservando las mismas posiciones de los padres.
5. Los elementos que faltan en el hijo 1 se toman del padre 2, colocándolos en el orden secuencial en que aparecen en el padre 2. Los genes que faltan en el hijo 2 se toman del padre 1, colocándolos en el orden secuencial en que aparecen en el padre 1.

Veamos el proceso descrito mediante un ejemplo:

- Se tienen 2 individuos (padres), seleccionados de la población inicial de acuerdo al vector de parejas X. El valor de L es de 18.

PADRE 1 [10 9 2 13 11 18 7 12 14 15 4 6 1 8 16 5 17 3]
PADRE 2 [8 12 14 9 15 3 13 11 1 2 4 6 7 16 10 18 5 17]

- Aleatoriamente se designan las posiciones uno y dos, con los valores 8 y 15 respectivamente.
- Los elementos que se heredan de los padres a los hijos conservando su posición son los siguientes:

HIJO 1 [10 9 2 13 11 18 7 5 17 3]
HIJO 2 [8 12 14 9 15 3 13 18 5 17]

- Los nuevos individuos tras cruzar la los elementos de sus padres son los siguientes:

HIJO 1 [10 9 2 13 11 18 7 8 12 14 15 1 4 6 16 5 17 3]
HIJO 2 [8 12 14 9 15 3 13 10 2 11 7 4 6 1 16 18 5 17]

b) Mutación: La figura 3-6 muestra el proceso del algoritmo para el operador de cruce.

Figura 3-6. Proceso para aplicar operador de mutación.



El proceso de cruce conlleva los siguientes pasos:

1. Se toma el individuo de la población inicial al cual se le va aplicar el operador de mutación
2. Tomando como referencia el valor del parámetro L, se designan dos posiciones aleatorias diferentes, de manera que la posición uno siempre sea menor que la posición dos.
3. Los datos que no están comprendidos entre el segmento designado por las posiciones se heredan a los hijos conservando las mismas posiciones.
4. El segmento designado por las posiciones se invierte y se coloca en el hijo conservando las mismas posiciones que tenía en el padre.

Veamos el proceso descrito mediante un ejemplo:

- Se tiene un individuo (padre) que pertenece a la población inicial. El valor de L es 18:

PADRE [18 17 9 11 10 8 5 1 7 4 3 2 16 13 12 15 14 6]

- Aleatoriamente se designan las posiciones uno y dos, con los valores 2 y 10 respectivamente.
- Se invierte la posición de los datos del segmento contenido entre las posiciones designadas de la siguiente forma:

HIJO [18 4 7 1 5 8 10 11 9 17 3 2 16 13 12 15 14 6]

Este proceso se realiza de la misma forma para todos los elementos de la población inicial para generar los nuevos individuos (hijos).

- c) Creación de la comunidad y desagregación de los bloques:**
 Dentro del proceso del algoritmo genético NSGA-II se debe dar la creación de una comunidad la cual integra en un solo conjunto las poblaciones de padres e hijos.

Hasta este punto los individuos (posibles soluciones) están dispuestos en términos de su codificación en bloques. Sin embargo, es necesario regresarlo a su codificación inicial para que mediante las funciones de evaluación se determine si cada uno de ellos cumple con las RD y así mismo conocer su nivel de optimización mediante la función Fitness.

A continuación se muestra como se realiza el proceso descrito:

1. Se tiene una solución cualquiera de un padres o un hijo, representada en bloques de la siguiente manera:

[6 4 8 10 14 17 7 11 5 9 16 3 2 12 18 13 1 15]

2. Para realizar el proceso de desagregación es necesario remitirse al vector de correspondencia de la representación en bloques con el vector de materias agrupado en pares. Para este ejemplo, dicho vector de correspondencia es el siguiente.

[1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18]																		
[1	1	1	1	1	0	2	2	2	0	3	3	3	3	3	0	4	0	5	0	6	0	7	7	8	8	8	8	9	9	10	10	11	11	11	11]

3. Al realizar el proceso se debe tener en cuenta que la posición del elemento en la representación en bloques, indica el número del bloque que se va a tomar para hacer la decodificación, y que el valor de dicho elemento representa la posición que ocupará dicho bloque en la

representación en el vector de materias. Por ejemplo, en la posición 1 de la representación en bloque esta el número 6, lo cual indica que el bloque conformado por los valores [1 1] pasará a ser el bloque número 6 en la representación en el vector de materias, ubicándose por tanto en las posiciones 12 y 13 de este. Otro ejemplo, es el elemento ubicado en la posición 10 de la representación en bloque y que tiene como valor en número 9, lo cual indica que este bloque está conformado por los valores [5 0] pasará a ser el bloque número 9 en la representación del vector de materias, ubicándose por tanto en las posiciones 17 y 18 de este. Las posiciones mencionadas se encuentran en color rojo dentro del vector de bloques desarmados.

```
[11 11 8 8 7 7 1 1 4 0 1 1 3 3 1 0 5 0 2 2 3 0 8 8 10 10 2 0 ...
... 11 11 6 0 3 3 9 9]
```

4. El paso a seguir es retirar los ceros de este vector para que la solución quede con las de 30 posiciones que corresponden. El vector resultante es el siguiente:

```
[ 11 11 8 8 7 7 1 1 4 1 1 3 3 1 5 2 2 3 8 8 10 10 2 11 11 6 3 3 9 9 ]
```

5. El vector anterior se debe acomodar en una matriz de 6 x 5, para lo cual se divide de a seis posiciones, las cuales se ubican como columnas de dicha matriz. Sin embargo, es necesario organizar los bloques para que en el momento de dividir el vector de a seis, estos no se separen. Por ejemplo, el bloque de la materia 11 señalado en rojo en el vector anterior, al dividir el vector de a seis, se fraccionará, quedando una hora en la columna 4 y la otra en la columna 5. Para evitar esto, lo que se hace es mover dicho bloque a la izquierda o a la derecha, intercambiándolo con aquellas horas que no están en bloque. El resultado es el siguiente:

```
[11 11 8 8 7 7 1 1 1 1 3 3 4 1 5 2 2 3 8 8 10 10 11 11 2 6 3 3 9 9]
```

6. El resultado es la solución para la matriz de Materias

$$\begin{bmatrix} 11 & 1 & 4 & 8 & 2 \\ 11 & 1 & 1 & 8 & 6 \\ 8 & 1 & 5 & 10 & 3 \\ 8 & 1 & 2 & 10 & 3 \\ 7 & 3 & 2 & 11 & 9 \\ 7 & 3 & 3 & 11 & 9 \end{bmatrix}$$

A partir de la construcción de la matriz de Materias para cada solución, se construyen las correspondientes matrices de Profesores, Códigos de Área y Salones, las

cuales se emplean para evaluar las diferentes restricciones.

D. Evaluación de las restricciones: La evaluación de las restricciones es una de las partes más importantes dentro de la ejecución del algoritmo, ya que de ello depende la toma de decisiones en cuanto a la validez de las soluciones encontradas, al igual que el nivel de optimización alcanzado en cada una de ellas, de acuerdo a los criterios establecidos.

La evaluación de estas restricciones se hace sobre cada una de las soluciones que conforman la comunidad. Para ello, se aplican sobre dichas soluciones las funciones definidas en las ecuaciones del 3-1 al 3-6.

a) Evaluación de las restricciones duras: Las restricciones duras 1 y 2 se evalúan en F_1 mediante la función F_{REST1} aplicada a la hipermatriz de Materias, mientras que las restricciones duras 3 y 4, se evalúan en F_2 mediante la función F_{REST2} aplicada a las hipermatrices de Profesores y Salones. El valor de F_{REST2} va a ser igual a cero cuando la hipermatriz que evalúa posee información de un solo grupo. El valor de RD que es igual a la suma de F_1 y F_2 , debe ser igual a cero para que la solución se considere válida; de lo contrario ésta se considera inválida.

A continuación se muestra el proceso mediante un ejemplo:

1. Se tiene la siguiente Matriz de Materias

$$\begin{bmatrix} 11 & 1 & 4 & 8 & 2 \\ 11 & 1 & 1 & 8 & 6 \\ 8 & 1 & 5 & 10 & 3 \\ 8 & 1 & 2 & 10 & 3 \\ 7 & 3 & 2 & 11 & 9 \\ 7 & 3 & 3 & 11 & 9 \end{bmatrix}$$

Se realiza la evaluación de la función F_{REST1} y se obtienen como resultado que $F_1=1$. Este valor se da porque el día martes la materia número 11 se presenta más de 2 veces. Aunque es la única infracción que existe, este valor inválida automáticamente la solución así el resto de restricciones se llegasen a cumplir.

2. Se tienen las siguientes hipermatrices de Profesores y Salones correspondientes a un problema de 3 grupos

$$\begin{bmatrix} 23 & 3 & 9 & 14 & 5 \\ 23 & 3 & 3 & 14 & 10 \\ 14 & 3 & 9 & 18 & 8 \\ 14 & 3 & 5 & 18 & 8 \\ 10 & 8 & 5 & 23 & 15 \\ 10 & 8 & 8 & 23 & 15 \end{bmatrix}$$

GRUPO 1

$$\begin{bmatrix} 8 & 10 & 5 & 3 & 14 \\ 5 & 10 & 5 & 9 & 14 \\ 6 & 8 & 3 & 14 & 6 \\ 6 & 8 & 3 & 14 & 6 \\ 5 & 15 & 3 & 19 & 8 \\ 10 & 15 & 3 & 19 & 8 \end{bmatrix}$$

GRUPO 2

$$\begin{bmatrix} 9 & 12 & 19 & 18 & 6 \\ 18 & 12 & 19 & 18 & 6 \\ 16 & 12 & 6 & 16 & 16 \\ 16 & 13 & 6 & 16 & 13 \\ 9 & 13 & 13 & 10 & 13 \\ 12 & 10 & 13 & 10 & 13 \end{bmatrix}$$

GRUPO 3

Hipermatriz de Profesores

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 7 & 1 & 1 & 1 & 1 \\ 7 & 1 & 1 & 1 & 1 \end{bmatrix}$$

GRUPO 1

$$\begin{bmatrix} 2 & 7 & 2 & 2 & 2 \\ 2 & 7 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

GRUPO 2

$$\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 7 & 3 \\ 3 & 3 & 3 & 7 & 3 \end{bmatrix}$$

GRUPO 3

Hipermatriz de Salones

Se realiza la evaluación de la función F_{REST2} para la hipermatrices de Profesores y Salones y se obtienen como resultado que $F_2=1$. Este valor se da porque el profesor número 10 se encuentra asignado a la sexta hora del lunes a los grupos 1 y 2. Para los salones no existe violación a ninguna de las restricciones, por tanto su valor es igual a 0 y no altera el valor de F_2 .

3. De acuerdo a lo anterior, se obtiene que $RD=2$, dando por entendido que se trata de una solución no válida.

- b) **Evaluación de las restricciones blandas:** Las restricciones blandas se evalúan mediante la función F_{REST3} aplicada a la hipermatriz de Códigos de Área. Esta función valora las tres diferentes restricciones existentes acorde con la penalización establecida para cada área del conocimiento en función del momento de la jornada donde se encuentre ubicada. El valor total se almacena en *Fitnees*, el cual determina el nivel de optimización de una solución. Se espera que este valor sea el mínimo posible.

A continuación se muestra el proceso mediante un ejemplo:

1. Se tiene la siguiente Matriz de Códigos de Área.

$$\begin{bmatrix} 2 & 1 & 5 & 3 & 4 \\ 2 & 1 & 1 & 3 & 7 \\ 3 & 1 & 5 & 6 & 2 \\ 3 & 1 & 4 & 6 & 2 \\ 7 & 2 & 4 & 2 & 8 \\ 7 & 2 & 2 & 2 & 8 \end{bmatrix}$$

2. Al aplicar la función F_{REST3} a toda la matriz, ésta entrega el valor de penalización de cada una de las asignaturas de acuerdo a su código y hora del día. Los resultados se muestran en la tabla 3-4:

Tabla 3-4. Resultados Función F_{REST3}

Código	Hora del día	Restricción a la que pertenece	Valor de penalización
2	1	2	6
2	2	2	3
3	3	1	3
3	4	1	3

Código	Hora del día	Restricción a la que pertenece	Valor de penalización
7	5	3	0
7	6	3	0
1	1	2	6
1	2	2	3
1	3	2	0
1	4	2	0
2	5	2	3
2	6	2	6
5	1	0	0
1	2	2	3
5	3	0	0
4	4	1	3
4	5	1	6
2	6	2	6
3	1	1	0
3	2	1	0
6	3	3	6
6	4	3	6
2	5	2	3
2	6	2	6
4	1	1	0
7	2	3	3
2	3	2	0
2	4	2	0
8	5	3	0
8	6	3	0
Fitness total			75

En la primera columna aparece el código de área, en la segunda columna aparece la hora del día en la que se dicta cada materia, en la tercera aparece la restricción a la que está faltando dicho código y en la cuarta columna aparece el valor de penalización correspondiente.

- De acuerdo a lo anterior, Fitness=75, el cual corresponde a la suma de todos los valores de penalización.

E. Selección: La selección es el procedimiento mediante el cual el algoritmo debe decidir cuáles de los elementos de la comunidad tiene las mejores aptitudes para continuar como parte del proceso evolutivo, rumbo a una solución óptima del problema. En el caso del NSGA-II se establecen dos criterios esenciales para dicha tarea, ellos son:

- Rango de dominancia según el frente de Pareto donde este el individuo.
- Distancia local de apilamiento.

A través de estos criterios es posible obtener los mejores individuos de la población, cuya diversidad permita explorar con mayor eficiencia el espacio de búsqueda de la solución,

de manera que sea posible converger más rápido a una solución óptima del problema.

a) Concepto de dominancia: Cuando se trata con problemas de optimización mono-objetivo, una alternativa de solución se considera mejor que otra si su función de evaluación tiene un resultado de menor valor para el caso de minimización, o de mayor valor para el caso de maximización. No obstante en problemas multi-objetivo este criterio debe ser revaluado, ya que se consideran al mismo tiempo varias funciones con objetivos de minimización, maximización o ambos. Además, el orden de magnitud de cada función puede ser diferente, haciendo inviable una comparación directa.

Por tanto, los algoritmos multi-objetivo introducen el concepto de *Dominancia*, el cual define que una solución α_1 domina otra solución α_2 si la solución α_1 no es de menor aptitud que la solución α_2 en todos los objetivos y si la solución α_1 es estrictamente mejor que α_2 en al menos uno de los objetivos. Si se incumple alguna de las condiciones descritas la solución α_1 no domina a la solución α_2 (Correa Flórez, Bolaños, & Molina Cabrera, 2008).

Por tanto, si en un conjunto de soluciones de un problema de optimización multi-objetivo se verifican iterativamente las condiciones mencionadas, se puede establecer cuáles soluciones conforman el Conjunto Dominante. Las soluciones restantes conforman el Conjunto de Soluciones Dominadas. Al establecer el conjunto de Soluciones Dominantes a través de todo el espacio objetivo, entonces se habla de Frente Óptimo de Pareto (Peñuela Meneses & Granada Echeverri, 2007). Una de las técnicas más eficientes para encontrar estos frentes es el método discutido en Kung et al. (1975), el cual se explica en el primer paso del proceso de selección.

b) Obtención de los frentes de Pareto: Según Kung et al. (1975), existen dos aspectos básicos para encontrar un frente: el primero es obtener una secuencia ordenada de individuos y el segundo es tomar esa secuencia ordenada y verificar la dominancia de sus individuos para descartar aquellos que sean dominados. Por tanto, se organizan los individuos de menor a mayor valor con respecto a la función objetivo RD , posteriormente se aplica una división recursiva de la nueva población ordenada hasta que se puedan comparar sub-poblaciones de un individuo, en donde la sub-población derecha es dominada por la sub-población izquierda respecto a la función objetivo RD . El frente se conforma por la sub-población izquierda más los individuos de la sub-población derecha que no son dominados por ningún individuo de la sub-población izquierda con respecto a la función

objetivo Fitness. Luego de encontrar un frente, sus individuos se descartan de la población ordenada y se repite el proceso hasta ubicar todos los individuos en frentes.

A continuación se explica el procedimiento mediante un ejemplo:

1. En la tabla 3-5 se muestra la comunidad de individuos para el desarrollo del horario de un grupo, en el que cada individuo tiene sus respectivos resultados para las funciones RD y Fitness, además de un número de identificación el cual podemos ver en la primera columna de la tabla:

Tabla 3-5. Ejemplo comunidad para selección de frentes.

#	Población concatenada Padres e Hijos																		RD	Fitness
1	18	8	14	6	1	7	9	4	2	11	13	10	17	3	5	15	12	16	0	60
2	16	13	8	7	1	4	14	9	10	2	11	12	3	5	6	18	15	17	0	81
3	15	8	3	13	12	4	1	9	7	6	2	10	11	18	16	17	5	14	10	111
4	17	1	7	5	16	2	14	3	9	8	4	15	11	6	18	10	13	12	20	69
5	18	8	14	6	13	7	1	4	9	10	2	11	3	5	15	17	12	16	10	96
6	16	13	8	7	18	14	6	1	9	4	2	11	10	3	5	12	15	17	10	81
7	1	7	3	9	8	4	15	13	12	6	2	10	11	18	16	17	5	14	20	108
8	3	1	9	7	2	16	17	5	14	8	4	15	11	6	18	10	13	12	30	81

2. Se organiza la comunidad de manera ascendente con respecto a la función RD y esta colocación se almacena en el vector P de acuerdo a su número de identificación, así:

$$P = [1 \ 2 \ 3 \ 5 \ 6 \ 4 \ 7 \ 8]$$

3. Para hallar los frentes se utiliza la función *FND_ordenado*. El proceso de la función se desarrolla por niveles de forma recursiva hasta encontrar un conjunto de individuos que puedan conformar un frente. Su operación se describe a continuación:

- En el primer nivel o iteración, se dividen los elementos del vector P hasta conformar parejas y en ellas se evalúa si el resultado de la función Fitness del primer individuo es mayor o igual que el resultado de la función Fitness del segundo. En caso afirmativo, ambos individuos clasifican al siguiente nivel y en caso negativo, clasifica al siguiente nivel sólo el primero de ellos. En caso de que clasifiquen dos individuos, estos se guardan en una sola variable y se consideran como un individuo con 2 elementos. Este procedimiento se realiza hasta comparar todas las parejas.
- En el segundo nivel o iteración, se toman los individuos clasificados del primer nivel y se les aplica de nuevo el

mismo procedimiento, agrupándolos en parejas y comparándolos de acuerdo a la función Fitness, para escoger cuáles de ellos prosiguen al siguiente nivel. Como es posible hallar que un individuo este conformado por dos elementos, la comparación se realiza sólo con el segundo elemento de este individuo y se aplica el mismo criterio de clasificación.

- La iteración anterior se repite hasta que se tenga un solo individuo en la secuencia, el cual puede estar conformado varios elementos.
- Sí el último individuo de la secuencia sólo tiene un elemento, este es el frente. De lo contrario, si el resultado de la secuencia es individuo con varios elementos, se debe verificar la dominancia dentro de la misma antes de terminar el proceso. Para ello, se compara el valor de la función Fitness del primer individuo del frente con el de los demás individuos. Para que un individuo pertenezca al frente el valor de su función Fitness debe ser menor o igual al del primer elemento del frente. En caso tal de que alguno de los individuos incumpla dicho criterio, este se excluye del frente.
- Para obtener los demás frentes de la población, se retiran los individuos que conforman el frente previamente hallado del vector P y se realiza de nuevo el procedimiento anterior.

De acuerdo al ejemplo, al aplicar la función FND_Ordenado se obtiene el siguiente resultado para la consecución del primer frente (Ver tabla 3-6):

Tabla 3-6: Resultado de aplicación función FND_Ordenado para hallar primer frente.

	[1 2 3 5 6 4 7 8]							
NIVEL 1	[1 2 3 5]				[6 4 7 8]			
	[1 2]	[3 5]	[6 4]	[7 8]				
NIVEL 2	[1]	[3 5]	[6 4]	[7 8]				
NIVEL 3	[1]				[6 4]			
SECUENCIA	[1]							

En el nivel 1 se aplica un proceso recursivo dividiendo la población por la mitad hasta obtener las parejas a comparar. Para el caso de la primera pareja, conformada por los individuos [1 2], el resultado de la función Fitness es 60 para el primer individuo y 81 para el segundo. En este caso se descarta el individuo 2 ya que el valor de la función Fitness es mayor que la del individuo 1. Para la segunda pareja conformada por los individuos [3 5] el resultado de la función Fitness es 111 para el primer individuo y 96 para el segundo. En este caso, ambos individuos clasifican al siguiente nivel ya que el valor de la función Fitness del

primer individuo es mayor que la del segundo. El proceso anterior se repite para todas las parejas.

Una vez efectuada la comparación en todos los niveles posibles, se obtiene una secuencia conformada por un solo individuo, el cual representa el primer frente para realizar la selección.

Para la consecución del segundo frente es necesario retirar el individuo 1 del vector P, el cual queda de la siguiente manera:

$$P = [2 \ 3 \ 5 \ 6 \ 4 \ 7 \ 8]$$

Al aplicar la función FND_Ordenado se obtiene el siguiente resultado para la consecución del segundo frente (Ver tabla 3-7):

Tabla 3-7: Ejemplo de aplicación función FND_Ordenado.

	[2 3 5 6 4 7 8]						
NIVEL 1	[2 3 5 6]				[4 7 8]		
	[2 3]	[5 6]	[4 7]	[8]			
NIVEL 2	[2]	[5 6]	[4]	[8]			
NIVEL 3	[2 5 6]	[4]					
SECUENCIA	[2 5 6 4]						

En el nivel 3 de la tabla 3-7, la comparación de parejas se debe llevar a cabo entre un individuo con tres elementos y un individuo de un solo elemento. En este caso la comparación se realiza entre el último elemento del primer individuo y el segundo individuo, es decir, entre los individuos 6 y el 4. El resultado de la función Fitness es 81 para el primer individuo y 69 para el segundo, por tanto, ambos individuos clasifican al siguiente nivel.

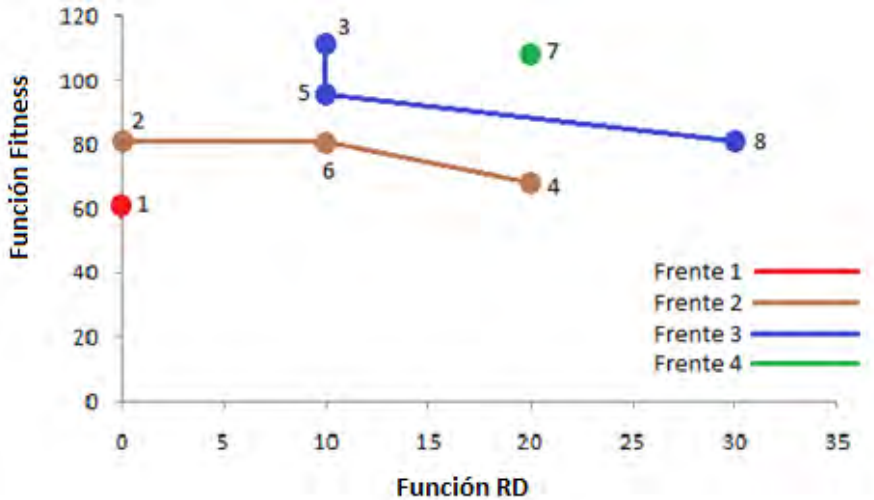
Como el resultado de la secuencia es un individuo con más de un elemento, se debe realizar la verificación de dominancia. El valor de la función Fitness del individuo 2 se debe comparar con el de los individuos [5 6 4]. Como resultado de dicha comparación se encuentra que el valor de la función Fitness del individuo 5 es mayor que la del individuo 2, por tanto, dicho elemento es excluido de la secuencia. De esta manera el segundo frente queda constituido de la siguiente forma: [2 6 4].

Estos individuos que ya fueron ubicados en el segundo frentes se descartan de la población y se repiten todos los pasos anteriores hasta encontrar todos los frentes. El resultado final es el siguiente:

FRENTE	1	1		
FRENTE	2	2	6	4
FRENTE	3	3	5	8
FRENTE	4	7		

En la figura 3-7 se observa la distribución gráfica de cada uno de los frentes obtenidos.

Figura 3-7: Distribución gráfica de los frentes.



c) **Factor de diversidad:** El factor de diversidad permite determinar la distancia de agrupamiento de los individuos, con el fin de medir el grado de diversidad de cada individuo dentro de una población. El grado de diversidad de un individuo es mayor a medida que se aleje de los otros individuos y es menor en la medida que se acerca a los demás. Entre mayor sea el grado de diversidad, es decir, entre mayor sea la distancia entre individuos, estos tienden a explorar mejor el espacio de soluciones para encontrar soluciones nuevas.

El factor de diversidad sólo se mide entre los individuos de un mismo frente. En un frente el grado de diversidad de los individuos de los extremos tiende a infinito.

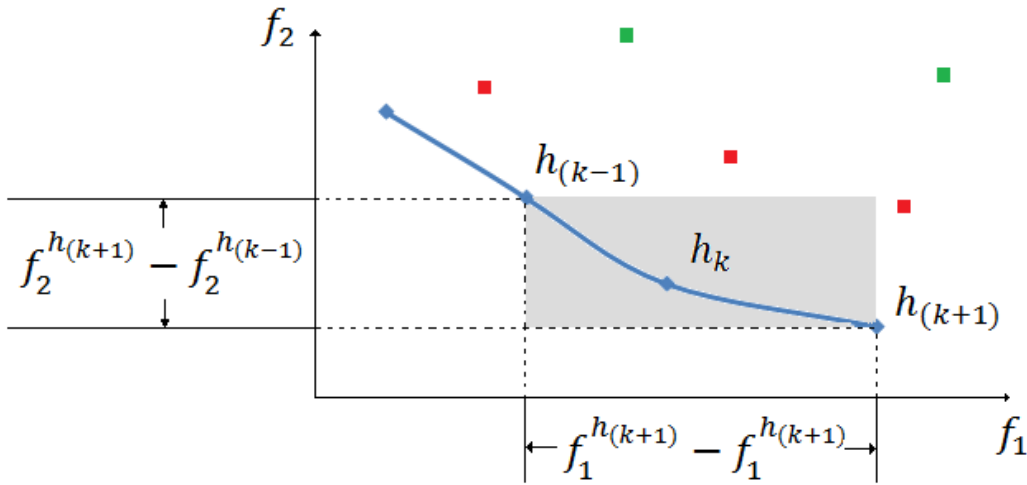
El grado de diversidad para cada individuo se mide de acuerdo a la ecuación 3-7:

$$d_k = \sum_{n=1}^N \left| \frac{f_n^{(i_{k+1}^n)} - f_n^{(i_{k-1}^n)}}{f_n^{max} - f_n^{min}} \right| \tag{3-7}$$

Donde I^n es un vector que indica la alternativa de solución vecina a la alternativa k , f_n^{max} y f_n^{min} son los valores máximo y mínimo sobre todo el espacio de solución de la función objetivo n y N es el número de funciones objetivo optimizadas.

En la figura 3-8 se puede observar la representación gráfica para el cálculo del factor de diversidad para una grupo de soluciones con 2 funciones objetivo, tres frentes y 9 individuos.

Figura 3-8: Representación gráfica factor de diversidad.



En este caso, el factor de diversidad se representa por medio de la ecuación 3-8:

$$d_k = \left| \frac{f_1^{h(k+1)} - f_1^{h(k-1)}}{f_1^{max} - f_1^{min}} \right| + \left| \frac{f_2^{h(k+1)} - f_2^{h(k-1)}}{f_2^{max} - f_2^{min}} \right| \tag{3-8}$$

El cálculo del factor de diversidad solo se realiza en caso de ser necesario ya que el principal criterio para determinarlos es el grado de dominancia que viene dado por los frentes (Peñuela Meneses & Granada Echeverri, 2007).

- d) **Seleccionar nuevos individuos:** La nueva población de individuos se construye agregando los individuos de cada frente hasta completar el tamaño de la población inicial. Sin embargo, puede ocurrir que al agregar un frente a la solución, la cantidad de individuos supere el tamaño de la población inicial. En este caso, si antes del último frente adicionado faltan 1 o 2 individuos, se adicionan el primer individuo o el primero y el último individuos del frente siguiente, respectivamente. En caso de que falten más de 2 individuos, se agrega el primero y el último individuo más los escogidos mediante el factor de diversidad, hasta que el

tamaño sea igual a la población inicial, teniendo prioridad aquellos individuos cuyo factor de diversidad sea más alto. Este factor permite medir la distancia entre los individuos del mismo frente y logra una mejor exploración del espacio de soluciones.

Para el ejemplo, en donde el tamaño de la población inicial debe ser de 4 individuos, se seleccionan dichos individuos de los frentes de acuerdo al criterio anterior, obteniendo que la nueva población se conforma por los siguientes individuos: [1 2 6 4]. En este caso sólo basto agregar los dos primeros frentes para completar el tamaño justo de la población inicial, no siendo necesaria ninguna otra acción.

- e) **Verificación de validez:** Una vez realizadas 40 iteraciones, se verifica dentro la última generación encontrada que haya por lo menos una solución válida para el grupo, es decir, que exista algún individuo cuyo valor en la función RD sea igual a cero. En caso de que no exista ninguna solución válida, se reinicia automáticamente el algoritmo para dicho grupo. Este proceso se repite hasta alcanzar una solución válida.

Cuando se obtiene al menos una solución válida, ésta se guarda dentro de una hipermatriz que contiene la solución general para todos los grupos, y se reinicia el proceso del algoritmo genético NSGA-II para el siguiente grupo. Este proceso se realiza hasta completar la cantidad total de grupos.

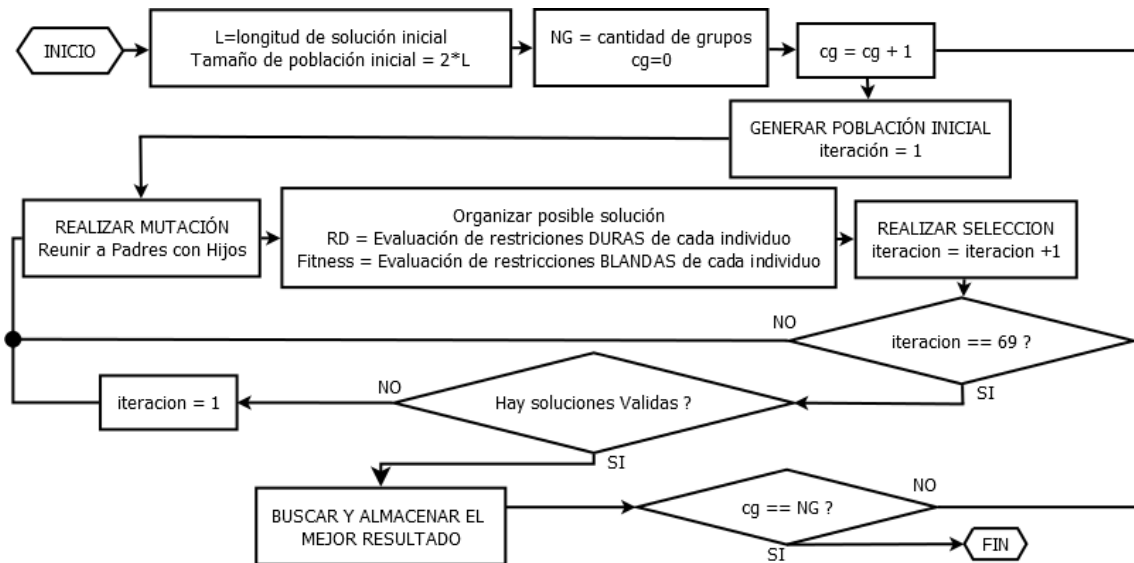
3.4.4 Método de solución 2: Algoritmo genético simple

El algoritmo genético es una técnica empleada para solucionar problemas de alta complejidad computacional, que permite tomar una población inicial de individuos y evolucionarla, con el fin de obtener nuevas generaciones de resultados, que por lo general son mejores que los anteriores, de tal forma que facilita llegar a una solución óptima en un tiempo relativamente corto, comparado con el tiempo que se tomaría un computador en encontrar todas las posibles soluciones al problema abordado.

Este algoritmo genético no se aplica a toda la solución en conjunto, sino que se utiliza para la solución individual de cada grupo, con el fin de que las variaciones sean más particulares, lo que permite tener una evolución consecutiva de la población en función de los mejores resultados de cada grupo. De esta manera, el algoritmo se aplicará tantas veces como grupos hayan para incluir en la solución final.

La solución inicial hallada en la sección 3.3.3 constituye la información con la cual se inicia la ejecución del algoritmo NSGA-II. El proceso general de éste se encuentra descrito en la figura 3-9.

Figura 3-9. Diagrama de flujo de operación del algoritmo genético simple



A continuación se presentan los pasos para su implementación.

- A. Población inicial:** Este paso del algoritmo es homólogo al desarrollado por el algoritmo NSGA-II en la sección 3.4.3 literal B, sólo que para este algoritmo la solución inicial no se encuentra codificada en bloques, por lo cual el parámetro L siempre es igual a 30.
- B. Mutación:** Normalmente en un algoritmo genético se realizan las operaciones de cruce y mutación con el fin de evolucionar la población inicial (Padres) y encontrar las nuevas generaciones (Hijos). Para el presente caso sólo se aplica el proceso de mutación debido a que el cruce alterará la matriz de códigos de materia de cada grupo, modificando la cantidad de horas a la semana de cada materia, lo cual implica el incumplimiento de las restricciones evaluadas en la sección 3.3.2.

A nivel general, la mutación consiste en una alteración o cambio de la información genética, que en este caso está representada por cada uno de los datos que constituyen una de las soluciones iniciales del problema. Existen diferentes maneras de realizar la mutación, de acuerdo con el problema que se esté desarrollando. En el presente caso el proceso realizado es el siguiente:

- Se toma el vector solución y se determina su longitud.
- Se designan dos posiciones diferentes, de forma aleatoria, comprendidas entre 0 y el valor de la longitud del vector solución, de manera que la posición uno siempre sea menor que la posición dos.
- Los datos comprendidos entre las posiciones designadas se invierten de posición y los datos fuera del segmento seleccionado se dejan intactos.

Veamos el proceso descrito mediante un ejemplo:

- Se tiene el siguiente elemento de la población inicial , el cual tiene 30 genes que representa el horario de grupo 1, el cual se muestra a continuación:

```
[11 6 1 1 3 10 2 9 2 11 3 1 2 8 4 3 10 8 7 11 3 8 8 3 7 1 1 5 9 11]
```

- Aleatoriamente se designan las posiciones 7 y 12
- Se invierte la posición de los datos del segmento contenido entre las posiciones designadas de la siguiente forma:

```
[11 6 1 1 3 10 1 3 11 2 9 2 2 8 4 3 10 8 7 11 3 8 8 3 7 1 1 5 9 11]
```

El proceso anterior se realiza de igual forma para toda la población inicial para generar los nuevos individuos (hijos).

Luego de aplicar la mutación se conforma la comunidad agregando en un mismo conjunto a Padres e Hijos, quedando así duplicado el tamaño de la población inicial.

C. Evaluación de las restricciones duras y blandas: Este paso del algoritmo es homologo al desarrollado por el algoritmo NSGA-II en la sección 3.4.3 literal D.

D. Selección: Una vez se tienen los vectores con las variables de verificación de las restricciones, se procede a crear un vector que describe las parejas de soluciones de la población inicial que van a ser enfrentadas mediante torneo, con el fin escoger la mejor de las dos.

Seguidamente se guarda en una matriz toda la población inicial (Padres e Hijos) en forma de vector fila, en la que cada posible solución va acompañada con el valor de evaluación de la función RD y la función Fitness.

Una vez obtenida la matriz anterior se procede a realizar la selección de las mejores soluciones, haciendo uso de los valores de las funciones de evaluación, de la siguiente manera:

- Se toman las soluciones que deban ser enfrentadas, de acuerdo al vector de parejas creado.
- Se comparan los valores de verificación de la función RD, seleccionando el que posea un menor valor (preferiblemente cero). En caso de que sean iguales se realiza el siguiente paso.
- Se comparan los valores de la función Fitness de cada solución y se escoge el que posea menor. En caso de que sean iguales, se escoge cualquiera de las dos soluciones al azar, ya que se consideran semejantes.

A continuación, se ilustra el proceso descrito mediante un ejemplo:

Tabla 3-8. Ejemplo de selección mediante algoritmo genético simple

N	1	2	3	4	5	6	7	8
Solución en forma de vector fila	11	1	1	2	1	2	2	2
	8	10	10	5	10	5	5	5
	11	3	2	1	3	1	1	1
	8	3	3	8	3	8	8	8
	5	3	3	3	3	3	2	3
	11	4	1	1	4	1	4	1
	3	8	2	3	8	3	7	3
	8	1	1	1	1	1	8	1
	3	8	8	8	2	8	1	8
	10	5	9	1	1	1	9	1
	1	9	5	11	3	9	11	11
	2	2	8	3	3	11	3	3
	2	1	1	11	8	3	11	11
	3	8	8	11	1	11	1	9
	2	1	4	7	2	1	8	2
	1	3	3	8	9	8	1	4
	11	3	3	9	5	7	3	7
	3	2	3	3	8	3	1	8
	1	1	1	11	1	2	3	1
	3	2	2	2	2	10	11	11
	6	11	7	4	11	11	10	10
	1	7	11	7	7	2	2	2
	9	7	7	8	7	4	3	3
	1	9	9	1	9	7	7	7
	9	11	11	9	11	11	11	11
	8	10	10	10	10	3	3	3
	10	6	6	2	6	9	9	9
	7	11	11	3	11	8	8	8
7	11	11	10	11	10	10	10	
4	8	8	6	8	6	6	6	
RD	0	0	0	10	0	0	10	10
Fit	45	72	63	63	75	72	42	66
Par	6	3	2	8	7	1	5	4

En la tabla 3-8 se puede observar los valores de verificación de las funciones RD y Fitness de cada solución. También se observa la fila con las parejas con las que enfrentó cada

solución. Esta fila se usa para determinar cuál de las soluciones pareadas continuará el proceso. En el caso de la solución 1, ésta tiene como pareja a la solución 6. Inicialmente se realiza la comparación entre los valores de la columna RD, encontrándose que las dos soluciones tienen un valor igual a cero (lo cual indica que son válidas), por lo cual es necesario comparar el valor de la función Fitness, en el que la solución 1 tiene un valor de 45 y la 6 tiene un valor de 72. Por tanto, de acuerdo al criterio de minimización se escoge la solución 1 y se descarta la 6. El proceso se realiza de igual forma para todas las parejas.

Después de aplicarle el proceso selección, la población inicial queda conformada por los individuos que se muestran en la tabla 3-9:

Tabla 3-9. Ejemplo de selección mediante algoritmo genético simple

N	1	2	3	4
	11	1	2	1
	8	10	5	10
	11	2	1	3
	8	3	8	3
	5	3	3	3
	11	1	1	4
	3	2	3	8
	8	1	1	1
	3	8	8	2
	10	9	1	1
	1	5	11	3
	2	8	3	3
	2	1	11	8
	3	8	11	1
	2	4	7	2
	1	3	8	9
	11	3	9	5
	3	3	3	8
	1	1	11	1
	3	2	2	2
	6	7	4	11
	1	11	7	7
	9	7	8	7
	1	9	1	9
	9	11	9	11
	8	10	10	10
	10	6	2	6
	7	11	3	11
	7	11	10	11
	4	8	6	8
RD	0	0	10	0
Fit.	45	63	63	75

Una vez terminada la selección se procede a repetir el proceso del algoritmo genético 70 veces con el fin de asegurar al menos una solución válida y que el valor de la función Fitness obtenido sea el mínimo posible.

E. Verificación de validez: Una vez realizadas las 70 iteraciones se verifica dentro la última población inicial que haya por lo menos una solución válida para el grupo. En caso de que no exista ninguna solución que cumpla con las restricciones duras, el programa automáticamente reinicia el algoritmo para dicho grupo. Este proceso se repite hasta alcanzar una solución válida.

Cuando se obtiene al menos una solución válida, ésta se guarda dentro de una hipermatriz que contiene la solución general para todos los grupos.

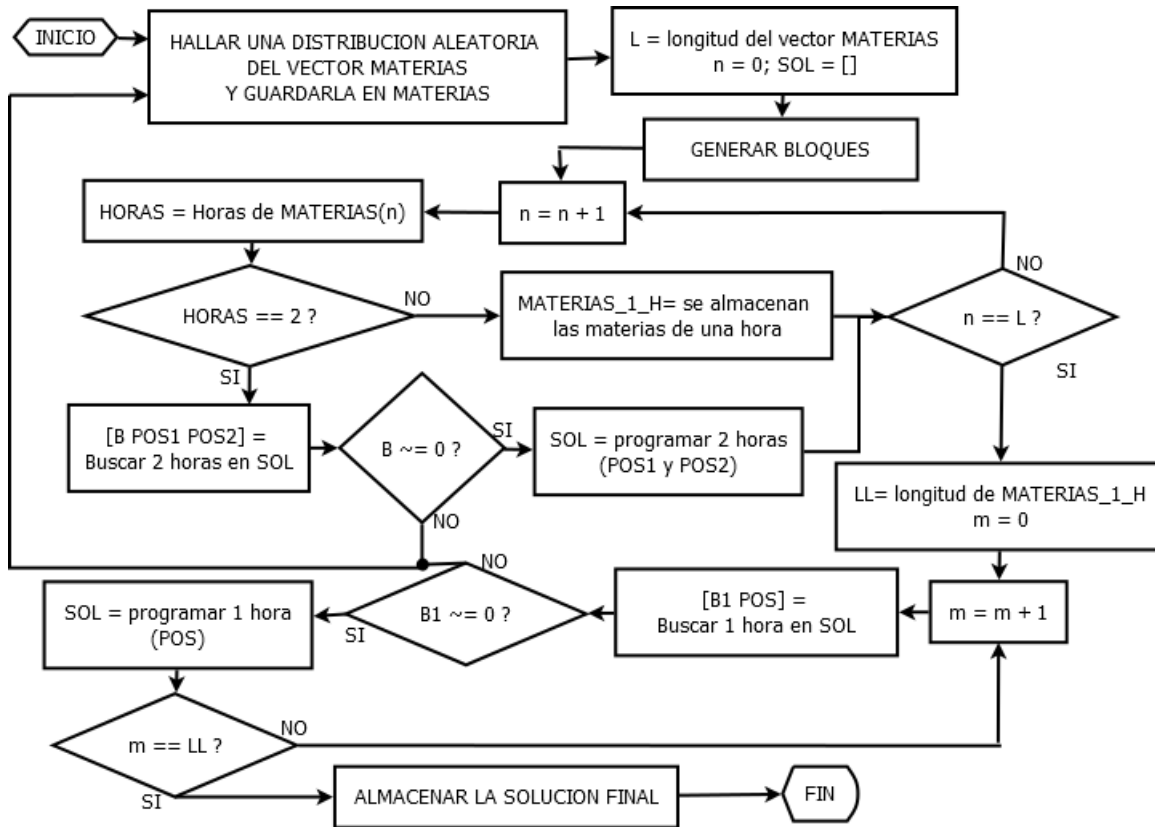
El proceso descrito en la sección 3.4.4 desde el literal A hasta el literal E se repite tantas veces como grupos haya para programar.

3.4.5 Método de solución 3: Búsqueda Aleatoria

La búsqueda aleatoria es una técnica en la que la solución del problema se construye a través de la generación de posiciones indiscriminadas para la ubicación de los diferentes elementos a distribuir. En este sentido, el algoritmo está estructurado para organizar las materias de forma que se cumplan las restricciones duras, garantizando que la solución que se genere sea válida. Sin embargo, este método no tiene procedimiento alguno que optimice la calidad de las soluciones encontradas (Fitness). La razón de aplicar este método, es que su enfoque de solución se realiza bajo el criterio general del problema sin abordar la orientación en los ritmos cognitivos, lo cual brinda la posibilidad de comparar la eficiencia de los métodos anteriormente descritos, con una técnica que no atiende el mismo objetivo de optimización, de manera que se demuestre el carácter robusto de los algoritmos genéticos en la consecución de soluciones que se acoplan perfectamente con el objetivo y complejidad, planteados en esta investigación.

El proceso general de éste se encuentra descrito en la figura 3-10.

Figura 3-10. Diagrama de flujo de operación del algoritmo búsqueda aleatoria



A continuación se presentan los pasos para su implementación.

A. Organizar bloques para todos los grupos: Después de realizar la evaluación de las restricciones de tabla (sección 3.3.2) se toma tabla que el usuario ha ingresado al programa y se duplican las materias que tengan una intensidad superior a dos horas, con el fin de que cada materia quede máximo de dos horas.

En la tabla 3-10 se muestra un ejemplo de la información ingresada por el usuario para programar un solo grupo. Para explicar la organización en bloques se tomará la materia 1 que tiene una intensidad horaria de 5 horas semanales. Ésta se debe duplicar dentro de la tabla, ya que en la primera división se agregará otra materia con el mismo código pero de 2 horas, quedando así una materia 1 de 3 horas y otra materia 1 de 2 horas, por lo cual se toma nuevamente la materia 1 de 3 horas para volverla a dividir, quedando así 3 materias 1, dos de 2 horas y 1 de una hora.

Tabla 3-10. Tabla base para organizar bloques en algoritmo de búsqueda aleatoria

Materia	# Horas	Grupo	Profesor	Área	Salón	# Grupo
1	5	62	3	1	1	1
2	3	62	5	4	1	1
3	5	62	8	2	1	1
4	1	62	9	5	1	1
5	1	62	9	5	1	1
6	1	62	10	7	1	1
7	2	62	10	7	7	1
8	4	62	14	3	1	1
9	2	62	15	8	1	1
10	2	62	18	6	1	1
11	4	62	23	2	1	1

La tabla 11 muestra lo que sucede con esta materia cuando se parte todo en materias de a 2 horas por semana, en la tabla se organizan las materias en orden ascendente de acuerdo a la cantidad de horas a la semana.

Tabla 3-11. Organización de bloques en algoritmo de búsqueda aleatoria

Fila	Materia	# Horas	Grupo	Profesor	Área	Salón	# Grupo
1	1	1	62	3	1	1	1
2	2	1	62	5	4	1	1
3	3	1	62	8	2	1	1
4	4	1	62	9	5	1	1
5	5	1	62	9	5	1	1
6	6	1	62	10	7	1	1
7	7	2	62	10	7	7	1
8	8	2	62	14	3	1	1
9	9	2	62	15	8	1	1
10	10	2	62	18	6	1	1
11	11	2	62	23	2	1	1
12	1	2	62	3	1	1	1
13	2	2	62	5	4	1	1
14	3	2	62	8	2	1	1
15	8	2	62	14	3	1	1
16	11	2	62	23	2	1	1
17	1	2	62	3	1	1	1
18	3	2	62	8	2	1	1

La posición de cada fila de la tabla anterior dentro de la matriz que la contiene se convierte en su identificador.

B. Encontrar un recorrido de forma aleatoria: Este paso consiste en tomar el número de filas de la tabla con las materias distribuidas en bloques y realizar una permutación aleatoria para designar un recorrido de este vector.

En el ejemplo mostrado sólo se tiene 1 grupo, pero si fueran n grupos, todas las materias quedarían ingresadas en la misma matriz sin importar el grupo.

En este caso se tienen 18 filas, por lo cual se determina una permutación aleatoria de 18 que se guarda en una variable llamada "recorrido", así:

```
recorrido [ 1 3 13 12 10 9 18 16 8 7 14 2 4 5 17 15 6 11 ]
```

El vector recorrido muestra el orden en el que se debe ir programando cada materia, que para el ejemplo inicia programando primero la materia de la fila 1, seguida de la materia de la fila 3 y así hasta completar todas las materias.

C. Definición de variables para almacenar la solución final: Se define un conjunto de variables en ceros donde cada que se programe una materia se almacenará el atributo correspondiente según corresponda, estas variables son: *materiash*, *materiasf*, *profesh*, *salonesh*.

D. Buscar dos horas: En este paso comienza la programación del horario de forma aleatoria. Para ello, del vector "recorrido" se toman primero todas las materias que son de dos horas. Si se llega a una materia de 1 hora esta se descarta por el momento. Cuando encuentra la primera materia de 2 horas se genera un número aleatorio entre 1 y 29 el cual se toma como posición para programar la primera hora de dicha materia. En este caso se tienen varias consideraciones en cuenta para saber si la posición encontrada es válida o no, estas son:

- Esta hora no puede estar al final del día es decir en las posiciones 6, 12, 18 y 24 porque de esta manera cuando se programe el horario, el bloque quedara partido entre la última hora de un día y la primera del día siguiente.
- Sí ya existe un bloque de esa materia programado el mismo día no se podrá programar durante ese día porque así se incumpliría una de las restricciones duras.
- Si el profesor que dicta esa materia ya tiene una materia a esa misma hora tampoco se podrá ya que se incumpliría una de las restricciones duras.
- Tampoco se podrá programar esa materia en un espacio en donde ya se haya programado cualquier otra materia en esa hora.

Las horas que se encuentren deben estar libres para ese grupo en específico y además deben cumplir todas las consideraciones anteriores para que sea válido programar la materia en las horas señaladas.

Cada vez que la hora escogida aleatoriamente no cumpla las condiciones descritas, se comienza un ciclo que recorre todas las horas de la semana del grupo al que pertenezca la materia y en cada iteración se busca un espacio adecuado. Sí al terminar el ciclo para ese grupo no se encuentra un espacio válido para programar la materia entonces se reinicia el proceso desde la sección 3.4.5 literal B, estableciendo una nueva distribución aleatoria de materias para programar.

En caso de que se encuentre dos espacios continuos que cumplan las consideraciones anteriores, entonces se pone en 1 una variable llamada "éxito" la cual indica que hay dos horas para programar dicha materia.

E. Programar dos horas: Este paso consiste en almacenar en la variable de solución las dos horas de la materia a programar, las cuales se hallaron en el paso anterior.

En la variable *materiash* se almacena la materia en la hora que se seleccionó para programar. De igual forma ocurre con la variable *materiasf* en donde se guardan los códigos de área correspondientes a la materia a programar en la hora designada.

La variable *profesh* tiene almacenados todos los profesores, cada uno con todas las horas de la semana. El profesor de la materia que se va programar se coloca en 1 a las horas correspondientes, esto para indicar que ese profesor ya está ocupado en esas horas.

De igual forma ocurre con la variable *salonesh*, ya que a cada espacio o salón se tienen asignadas todas las horas de la semana. Entonces, en este caso también se coloca en 1 en las horas correspondientes en el salón o espacio en el cual se dicta la materia a programar.

F. Buscar una hora: Después de haber programado todas las materias de 2 horas, se procede a examinar de nuevo el vector "recorrido", pero en este descartando las materias de 2 horas y buscando 1 hora en la cual se pueda programar cada materia, así:

Primero se genera una posición aleatoria entre 1 y 30 que son las horas de la semana, se analiza si esta posición es válida, evaluando las siguientes condiciones:

- Si ya existe otra hora de esa materia programada el mismo día no se podrá programar durante ese día porque así se incumplirían una de las restricciones duras.

- Si el profesor que dicta esa materia ya tiene una materia a esa misma hora tampoco se podrá porque así se incumpliría una de las restricciones duras.
- Tampoco se podrá programar esa materia en un espacio en donde ya se haya programado cualquier otra materia es esa hora.

En caso de que no se cumplan las condiciones anteriores para la hora hallada aleatoriamente, entonces se debe recorrer todo el horario verificando que exista un espacio libre programar dicha materia.

Sí al terminar el ciclo para ese grupo no se encuentra un espacio válido para programar la materia entonces se reinicia el proceso desde la sección 3.4.5 literal B, estableciendo una nueva distribución aleatoria de materias para programar.

Una vez se tenga la hora en la cual es válido programar la materia se procede a programar dicha materia.

G. Programar una hora: Luego de haber encontrado el espacio correcto para programar dicha hora, se procede programarla siguiendo el mismo proceso llevado a cabo para la programación de las materias de dos horas (Sección 3.4.5 literal E).

H. Armar el horario de todos los grupos: Una vez se hayan programado todas las materias, se almacena todo el horario en el formato de hipermatriz, ya que hasta este momento la programación realizadas se efectuó sobre una matriz de dos dimensiones, en la cual las filas son todas las materias de todos los grupos y las columnas son las 30 horas de la semana.

I. Calcular nivel de optimización: En este paso se calcula el valor de la función Fitness para cada grupo, para establecer el nivel de optimización alcanzado por la solución. Cabe aclarar que este valor es sólo de referencia, ya que la técnica no efectúa ningún procedimiento para mejorarlo.

3.5 Presentación de resultados

Una vez se completa la solución final para todos los grupos se procede a transformar las soluciones, que se encuentran en códigos, a soluciones por nombres mediante una función llamada "caracteres", la cual se encarga de tomar cada código y llevarlo a su nombre correspondiente. De igual manera se hace con los códigos de los profesores mediante una función llamada "caracterespro".

Las soluciones obtenidas se exportan a un archivo de Excel. En la primera hoja del archivo se encuentra la solución por materias, en

la segunda hoja se encuentra la solución por profesores y en la tercera se encuentra el valor de la función Fitness.

En la tabla 3-12 se observan a la izquierda los códigos de materias y a la derecha el nombre de la materia en la forma como aparece en el archivo de Excel que contiene la solución.

Tabla 3-12. Ejemplo resultados de materias

8	2	2	8	5	Ciencias Naturales	Ciencias Sociales	Ciencias Sociales	Ciencias Naturales	Religión
8	2	4	8	1	Ciencias Naturales	Ciencias Sociales	Ética y valores	Ciencias Naturales	Matemáticas
3	3	3	1	1	Español	Español	Español	Matemáticas	Matemáticas
1	3	3	1	11	Matemáticas	Español	Español	Matemáticas	Inglés
10	9	11	7	11	Educación Física	Artística	Inglés	Informática	Inglés
10	9	11	7	6	Educación Física	Artística	Inglés	Informática	Informática

En la tabla 13, al lado izquierdo se observa los códigos de profesores para el horario mostrado en la tabla anterior y a la derecha sus respectivos nombres.

Tabla 3-13. Ejemplo resultado profesores

14	5	5	14	9	Profesor 14	Profesor 5	Profesor 5	Profesor 14	Profesor 9
14	5	9	14	3	Profesor 14	Profesor 5	Profesor 9	Profesor 14	Profesor 3
8	8	8	3	3	Profesor 8	Profesor 8	Profesor 8	Profesor 3	Profesor 3
3	8	8	3	23	Profesor 3	Profesor 8	Profesor 8	Profesor 3	Profesor 23
18	15	23	10	23	Profesor 18	Profesor 15	Profesor 23	Profesor 10	Profesor 23
18	15	23	10	10	Profesor 18	Profesor 15	Profesor 23	Profesor 10	Profesor 10

4. Resultados

Para verificar la estabilidad y eficiencia de los métodos implementados en la solución del problema de programación de horarios bajo las restricciones en ritmos de aprendizaje, se establece una comparación de acuerdo a los resultados de la función de evaluación. Para ello, se formularon 6 problemas con cantidad diferente de grupos, docentes y aulas, los cuales se aplicaron a cada método para ser resueltos durante 10 repeticiones. En cada una de las repeticiones el valor de la función Fitness es tomada como referencia. De igual manera, se presenta un análisis de los tiempos de ejecución de cada uno de los métodos.

En la tabla 4-1 se presenta una descripción de los problemas planteados, los cuales están basados en la carga académica de una institución pública real, la cual se acoge a los parámetros de 5 días de clase y 6 horas diarias de jornada escolar:

Tabla 4-1. Problemas de prueba aplicados

Problema	Cantidad de grupos	Cantidad de docentes	Cantidad de asignaturas	Cantidad de salones	Total de horas programadas
1	4	14	40	5	120
2	5	19	50	6	150
3	6	20	60	7	180
4	7	23	73	8	210
5	8	23	86	9	240
6	9	23	99	10	270

4.1 Estabilidad del sistema

Para verificar la fiabilidad de las soluciones entregadas por cada método en cada una de las repeticiones que realiza el algoritmo, se efectuó una comparación entre los valores de la función Fitness para cada uno de los grupos programados para las 10 repeticiones realizadas con cada problema.

En el Anexo A se puede encontrar la tabla con los resultados para cada una de las repeticiones, en cada uno de los métodos, para cada uno de los problemas, al igual que la tabla con la

información descriptiva que relaciona la media, la mediana, la desviación típica y el valor mínimo y máximo para cada caso.

En este mismo anexo se encuentran las pruebas de normalidad sobre dichos datos. Dada la cantidad de datos, este parámetro se estimó con la prueba de Shapiro-Wilk, encontrándose que no en todos los métodos, ni en todos los problemas, ni en todas las repeticiones, los datos seguían una distribución normal.

Teniendo en cuenta lo anterior, para realizar la comparación entre las soluciones obtenidas por cada repetición por cada uno de los métodos, se aplicó la prueba de Kruskal-Wallis, encontrándose los siguientes resultados (Ver tabla 4-2):

Tabla 4-2. Comparación por repetición para cada método

		Método de solución								
		Aleatorio			AG-Simple			NSGA-II		
Problema	Variable	Chi-cuadrado	Gl	Sig. asintót.	Chi-cuadrado	gl	Sig. asintót.	Chi-cuadrado	gl	Sig. asintót.
1	Valor Fitness	7,614	9	,573	6,257	9	,714	1,600	9	,996
2	Valor Fitness	4,960	9	,838	2,820	9	,971	2,163	9	,989
3	Valor Fitness	5,422	9	,796	4,999	9	,834	,826	9	1,000
4	Valor Fitness	7,879	9	,546	6,558	9	,683	2,901	9	,968
5	Valor Fitness	4,600	9	,868	2,139	9	,989	4,366	9	,886
6	Valor Fitness	6,432	9	,696	1,331	9	,998	2,429	9	,983

Prueba de Kruskal-Wallis

Variable de agrupación: Repetición

La significación de la prueba en todos los casos es mayor al 5%, por tanto los valores de los resultados de la función de valuación Fitness son estadísticamente iguales en cada uno de los métodos para cada repetición realizada en los diferentes problemas planteados. Esto demuestra la alta estabilidad ofrecida por los algoritmos en la consecución de soluciones a los diferentes problemas planteados.

4.2 Comparación por método

La tabla 4-3 muestra el promedio de la función de evaluación Fitness obtenido por cada uno de los métodos para cada problema, durante cada una de las repeticiones establecidas.

Tabla 4-3. Resultados totales de la función Fitness por problema y método

Problema	Método	Valor Fitness total por repetición									
		1	2	3	4	5	6	7	8	9	10
1	Aleatorio	339	336	354	357	294	324	372	285	333	321

Problema	Método	Valor Fitness total por repetición									
		1	2	3	4	5	6	7	8	9	10
2	Aleatorio	477	429	444	492	444	432	426	408	387	423
3	Aleatorio	543	483	486	555	492	462	525	483	513	513
4	Aleatorio	630	573	600	507	567	573	501	543	645	627
5	Aleatorio	699	702	651	696	591	690	714	672	693	675
6	Aleatorio	867	702	771	771	768	762	753	765	789	744
1	AG -Simple	228	273	279	261	252	240	261	276	267	258
2	AG -Simple	327	303	321	279	309	315	324	294	288	303
3	AG -Simple	387	411	330	387	447	423	375	390	384	411
4	AG -Simple	501	474	426	459	408	471	507	444	456	444
5	AG -Simple	537	534	528	552	585	546	573	549	516	564
6	AG -Simple	606	627	642	648	600	627	645	609	648	600
1	NSGA-II	201	177	150	165	165	210	168	189	183	204
2	NSGA-II	183	186	198	168	201	255	222	219	189	189
3	NSGA-II	318	300	258	267	276	285	264	285	276	264
4	NSGA-II	279	321	324	303	363	342	399	357	315	300
5	NSGA-II	312	327	423	426	390	384	396	423	411	348
6	NSGA-II	483	450	456	534	471	432	438	480	534	468

De los valores anteriores se obtiene la siguiente información descriptiva, la cual se encuentra dividida por método y problema solucionado (Ver tabla 4-4).

Tabla 4-4. Descriptivo de resultados totales de la función Fitness por problema y por método

Método de solución	Problema	N	Media	Mediana	Desv. típ.	Mínimo	Máximo
Aleatorio	1	10	331,50	334,50	27,102	285	372
	2	10	436,20	430,50	30,666	387	492
	3	10	505,50	502,50	29,470	462	555
	4	10	576,60	573,00	49,916	501	645
	5	10	678,30	691,50	35,509	591	714
	6	10	769,20	766,50	41,429	702	867
AG-Simple	1	10	259,50	261,00	16,078	228	279
	2	10	306,30	306,00	15,966	279	327
	3	10	394,50	388,50	31,440	330	447
	4	10	459,00	457,50	30,887	408	507
	5	10	548,40	547,50	21,109	516	585
	6	10	625,20	627,00	20,060	600	648
NSGA-II	1	10	181,20	180,00	19,708	150	210
	2	10	201,00	193,50	24,980	168	255
	3	10	279,30	276,00	18,518	258	318
	4	10	330,30	322,50	35,425	279	399
	5	10	384,00	393,00	41,425	312	426
	6	10	474,60	469,50	35,463	432	534

Antes de efectuar la comparación se realizaron pruebas de normalidad y homogeneidad de varianzas. Para la normalidad se utilizó la prueba de Shapiro-Wilk (Tabla 4-5) y para la homogeneidad de varianzas el estadístico de Levene (Tabla 4-6). Los resultados fueron los siguientes:

Tabla 4-5. Prueba de normalidad resultados totales de la función Fitness por problema y método

Método de solución	Variable	Problema.	Shapiro-Wilk		
			Estadístico	G1	Sig.
Aleatorio	Valor Fitness	1	,960	10	,789
		2	,952	10	,694
		3	,952	10	,690
		4	,939	10	,546
		5	,809	10	,019
		6	,849	10	,057
AG-Simple	Valor Fitness	1	,938	10	,528
		2	,958	10	,764
		3	,947	10	,629
		4	,972	10	,905
		5	,983	10	,979
		6	,858	10	,072
NSGA-II	Valor Fitness	1	,955	10	,732
		2	,914	10	,311
		3	,909	10	,277
		4	,968	10	,876
		5	,881	10	,132
		6	,888	10	,160

Tabla 4-6. Prueba de homogenidad de varianzas para resultados totales de la función Fitness por problema y método

Problema	Estadístico de Levene	g1	g2	Sig.
1	1,132	2	27	,337
2	1,001	2	27	,381
3	1,209	2	27	,314
4	1,288	2	27	,292
5	1,714	2	27	,199
6	,443	2	27	,647

La significación de ambas pruebas indica que para cada uno de los métodos en los problemas evaluados se cumplen los supuestos de normalidad y homocedasticidad.

De acuerdo a lo anterior, se efectúa la comparación entre los diferentes métodos de solución para cada uno de los problemas mediante un análisis de Anova (Ver tabla 4-7).

Tabla 4-7. Prueba ANOVA para comparación del resultado total de la función Fitness entre métodos

Problema		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Inter-grupos	113016,600	2	56508,300	122,720	,000
	Intra-grupos	12432,600	27	460,467		
	Total	125449,200	29			
2	Inter-grupos	277603,800	2	138801,900	228,882	,000
	Intra-grupos	16373,700	27	606,433		
	Total	293977,500	29			

3	Inter-grupos	255861,600	2	127930,800	174,459	,000
	Intra-grupos	19799,100	27	733,300		
	Total	275660,700	29			
4	Inter-grupos	303523,800	2	151761,900	96,859	,000
	Intra-grupos	42304,500	27	1566,833		
	Total	345828,300	29			
5	Inter-grupos	435046,200	2	217523,100	190,670	,000
	Intra-grupos	30802,500	27	1140,833		
	Total	465848,700	29			
6	Inter-grupos	434018,400	2	217009,200	192,817	,000
	Intra-grupos	30387,600	27	1125,467		
	Total	464406,000	29			

El análisis de varianza muestra que la significación es inferior al 5%, por tanto existen diferencias estadísticas entre los resultados obtenidos. Para determinar entre qué grupos se presentan las diferencias mencionadas se aplicó la siguiente prueba post hoc (ver tabla 4-8):

Tabla 4-8. Prueba Post hoc para comparación del resultado total de la función Fitness entre métodos

Problema	(I) Tipo de Algoritmo	(J) Tipo de Algoritmo	Diferencia de medias (I-J)	Error típico	Sig.	Intervalo de confianza al 95%	
						Límite inferior	Límite superior
1	Aleatorio	AG-Simple	72,000*	9,597	,000	48,21	95,79
		NSGA-II	150,300*	9,597	,000	126,51	174,09
		AG-Simple	78,300*	9,597	,000	54,51	102,09
2	Aleatorio	AG-Simple	129,900*	11,013	,000	102,59	157,21
		NSGA-II	235,200*	11,013	,000	207,89	262,51
		AG-Simple	105,300*	11,013	,000	77,99	132,61
3	Aleatorio	AG-Simple	111,000*	12,110	,000	80,97	141,03
		NSGA-II	226,200*	12,110	,000	196,17	256,23
		AG-Simple	115,200*	12,110	,000	85,17	145,23
4	Aleatorio	AG-Simple	117,600*	17,702	,000	73,71	161,49
		NSGA-II	246,300*	17,702	,000	202,41	290,19
		AG-Simple	128,700*	17,702	,000	84,81	172,59
5	Aleatorio	AG-Simple	129,900*	15,105	,000	92,45	167,35
		NSGA-II	294,300*	15,105	,000	256,85	331,75
		AG-Simple	164,400*	15,105	,000	126,95	201,85
6	Aleatorio	AG-Simple	144,000*	15,003	,000	106,80	181,20
		NSGA-II	294,600*	15,003	,000	257,40	331,80
		AG-Simple	150,600*	15,003	,000	113,40	187,80

El test de Tukey determinó que dicha diferencia se da entre todos los métodos (sig=0,000). En este sentido, el valor medio de la función de evaluación del método NSGA-II en todos los problemas, presenta una eficiencia 29,31% y 44,74% mayor con respecto al algoritmo genético simple y al algoritmo de búsqueda aleatoria respectivamente. A su vez el algoritmo genético simple es 21,95% más eficiente que el algoritmo de búsqueda aleatoria. De esta manera, se establece que el algoritmo NSGA-II presenta mejores resultados que los otros dos algoritmos implementados para la

solución del problema bajo las restricciones establecidas en esta investigación.

4.3 Comparación por tiempos de ejecución

En el Anexo A, se encuentra la tabla con los tiempos de ejecución de cada uno de los métodos para cada uno de los problemas durante las 10 repeticiones establecidas, al igual que las pruebas de normalidad y de contraste entre grupos. Al evaluar los datos se obtuvieron los resultados que se muestran en la tabla 4-9.

Tabla 4-9. Descriptivo de los tiempos de ejecución de cada uno de los métodos para cada uno de los problemas planteados.

Método	Problema	N	Media	Mediana	Desv. típ.	Mínimo	Máximo
		Válidos					
Búsqueda Aleatoria	1	10	5,1460	5,1750	,34507	4,61	5,84
	2	10	5,6480	5,4350	,66914	4,97	7,02
	3	10	7,2900	6,7150	2,19377	5,55	13,04
	4	10	10,0890	9,2700	4,31926	5,91	19,75
	5	10	48,3570	49,4100	31,42553	7,13	122,69
	6	10	210,8340	101,6100	242,08072	7,12	715,83
Genético Simple	1	10	304,9670	289,7750	31,75913	288,42	365,65
	2	10	379,7520	365,3200	32,07729	360,88	441,06
	3	10	482,2820	478,5600	42,47153	440,64	528,15
	4	10	602,4350	591,7450	80,16718	516,80	768,11
	5	10	673,2570	658,0050	64,80232	615,69	775,34
	6	10	831,4190	736,7350	244,29349	695,05	1485,87
NSGA-II	1	10	308,5620	273,3250	58,46039	260,63	436,71
	2	10	370,7600	347,6700	49,63809	331,97	468,80
	3	10	548,0470	566,2850	98,36912	412,73	679,12
	4	10	776,5490	803,4950	201,63079	473,15	1073,52
	5	10	668,5240	611,0450	177,47130	541,68	1089,32
	6	10	835,4450	794,1900	185,23403	609,84	1254,71

Se observa como el algoritmo de búsqueda aleatoria es el que mejores tiempos de ejecución presenta en la solución de todos los problemas. En este sentido, si se compara la media de tiempo de solución del primer problema entre el algoritmo NSGA-II y el algoritmo de búsqueda aleatoria, este último es aproximadamente 60 veces más rápido. Por el contrario, si se evalúa dicha diferencia sobre el problema de mayor complejidad, ésta se reduce a tan sólo 4 veces. De esta manera, a mayor complejidad del problema, la diferencia entre los tiempos de proceso de los algoritmos también se reduce drásticamente.

Dado que en las pruebas de normalidad, la muestra del algoritmo de búsqueda aleatorio presenta una distribución que no es normal, para efectuar el contraste con los otros dos métodos, se utiliza la prueba de Kruskal Wallis para comparar los tiempos de

ejecución de cada uno de los métodos para los 6 problemas. De este contraste la significación de la prueba ($\text{sig}=000$) indica que existen diferencias entre los tiempos de los tres métodos para todos los problemas. Por tanto, se realiza un contraste uno a uno mediante la prueba U de Mann Whitney determinando que dichas diferencias se presentan entre el algoritmo de búsqueda aleatoria con el algoritmo genético simple y el algoritmo NSGA-II. De su parte, el contraste entre estos dos últimos algoritmos indica que sus tiempos de ejecución no presentan diferencias estadísticamente significativas.

4.4 Prueba piloto

Se realizó una prueba piloto para verificar el efecto de la programación horaria dentro del proceso académico de la institución educativa Colegio San Jorge, ubicada en la ciudad de Manizales, establecimiento de carácter público, que atiende aproximadamente 640 estudiantes, repartidos en 17 grupos de sexto a once. Del grado sexto a décimo existen tres grupos por nivel, mientras que en undécimo sólo son dos grupos.

El experimento se diseñó para comparar dos diferentes grupos del mismo nivel, para todos los niveles de sexto a once, de tal forma que en cada nivel existiera un grupo con un horario programado mediante el algoritmo NSGA-II y otro programado con el algoritmo de búsqueda aleatoria. Dadas las condiciones de planta docente, fue imposible lograr que en todas las asignaturas fueran asignados los mismos profesores para los grupos y niveles en comparación. No obstante, se consiguió que el número máximo de docentes que difirieran por nivel no superara tres asignaturas.

El propósito de la prueba fue verificar la mortalidad académica de todas las asignaturas (Cantidad de asignaturas pérdidas) en cada uno de los grupos y determinar si existe alguna relación con el método de programación horaria utilizada. El problema aplicado a los dos algoritmos tiene los siguientes parámetros: 6 grupos, 23 profesores, 8 salones y 66 asignaturas. Se da la condición de uso de dos salones adicionales a la cantidad de grupos, los cuales corresponden a la sala de cómputo y al patio de deportes, los cuales se deben asignar a cada grupo sin que haya dos o más grupos asignados a estos salones al mismo tiempo.

La programación de horarios generada bajo los dos métodos, fue implementada en la institución de prueba durante dos periodos académicos consecutivos (1 semestre). Una vez aplicada la jornada propuesta bajo cada método (NSGA-II y Búsqueda Aleatoria), se recopiló la información de cada periodo lectivo respecto al rendimiento académico en los grupos evaluados la cual se encuentra resumida en las tablas 4-9 y 4-10. Allí se especifica la cantidad de alumnos por grupo y la cantidad de asignaturas orientadas en

cada nivel. La cantidad de asignaturas pérdidas corresponde a la sumatoria de las asignaturas pérdidas por todos los estudiantes de un grupo durante el periodo evaluado. El valor del porcentaje de pérdida intragrupo representa la proporción de estas asignaturas respecto a la cantidad total de asignaturas orientadas en un grupo. Dadas las condiciones variantes respecto a la permanencia de los alumnos en la institución se tiene en cuenta la cantidad de alumnos por grupo para el cálculo de pérdida intergrupo de manera que no se altere el contraste.

Tabla 4-10. Información académica por niveles primer periodo

Nivel y grupo	Método y valor Fitness	Cantidad alumnos	Cantidad asignaturas orientadas	Total asignaturas orientadas	Cantidad asignaturas pérdidas	% Pérdida intragrupo	% Diferencia entre grupos
6	1 aleatorio	108	49	490	89	18,2%	1,9%
	2 NSGA-II	24	48	480	78	16,3%	
7	1 aleatorio	96	42	420	85	20,2%	8,5%
	2 NSGA-II	54	41	410	48	11,7%	
8	1 aleatorio	84	41	410	68	16,6%	4,4%
	2 NSGA-II	51	42	420	51	12,1%	
9	1 aleatorio	84	41	410	106	25,9%	6,1%
	2 NSGA-II	60	42	420	83	19,8%	
10	1 Aleatorio	99	42	546	129	23,6%	0,7%
	2 NSGA-II	48	39	507	116	22,9%	
11	1 aleatorio	72	38	494	77	15,6%	2,3%
	2 NSGA-II	54	37	481	64	13,3%	
Promedio						18,0%	4,0%

Tabla 4-11. Información académica por niveles segundo periodo

Nivel y grupo	Método y valor Fitness	Cantidad de alumnos	Cantidad de asignaturas orientadas	Total asignaturas orientadas	Cantidad de asignaturas pérdidas	% Pérdida intragrupo	% Diferencia entre grupos
6	1 aleatorio	108	47	470	89	18,9%	2,3%
	2 NSGA-II	24	47	470	78	16,6%	
7	1 aleatorio	96	40	400	91	22,8%	5,2%
	2 NSGA-II	54	41	410	72	17,6%	
8	1 aleatorio	84	41	410	91	22,2%	8,3%
	2 NSGA-II	51	41	410	57	13,9%	
9	1 aleatorio	84	41	410	95	23,2%	2,5%
	2 NSGA-II	60	42	420	87	20,7%	
10	1 aleatorio	99	40	520	151	29,0%	8,1%
	2 NSGA-II	48	39	507	106	20,9%	
11	1 aleatorio	72	38	494	71	14,4%	6,3%
	2 NSGA-II	54	37	481	39	8,1%	
Promedio						19,0%	5,4%

Respecto a la programación realizada con cada uno de los algoritmos, los resultados muestran que los valores de la función

de evaluación Fitness son menores en todos los grupos programados mediante el método NSGA-II.

De cara a los resultados obtenidos por los grupos, durante el primer periodo académico, el valor promedio de asignaturas pérdidas por grupo es del 18% y la diferencia entre grupos muestra un promedio de 4 puntos porcentuales por encima en los grupos programados con el algoritmo de búsqueda aleatoria. En el segundo periodo académico, el valor promedio de asignaturas pérdidas por grupo es del 19% y la diferencia entre grupos muestra un promedio de 5,4 puntos porcentuales por encima en los grupos programados con el algoritmo de búsqueda aleatoria.

Esto indica que los grupos programados con la metodología propuesta, basada en el método NSGA-II, presentan una reducción del 22,22 % y del 28,63% en la cantidad de materias pérdidas por grupo, en el primer y segundo periodo respectivamente.

Por tanto, se evidencia que los grupos donde la programación horaria tiene un valor menor en la función de evaluación Fitness, presentan menor cantidad de materias pérdidas.

5. Discusiones

Aunque el problema de programación de horarios ha sido abordado desde múltiples perspectivas, la presente investigación muestra un elemento innovador al incluir dentro de los factores necesarios para la programación horaria, el ritmo de aprendizaje de los estudiantes durante la jornada escolar. En este sentido, además de elementos tales como profesores, asignaturas, grupos, días hábiles, salones y demás instalaciones (Flores Pichardo, 2011), esta investigación da un enfoque en el que la forma como se desarrolla el proceso educativo en los estudiantes es considerada como una variable adicional para la asignación de las diferentes materias en el horario. Por tanto, la complejidad del problema se incrementa, ya que no sólo se trata de ubicar salones, materias y profesores, evitando los cruces que entre estos se puedan generar y que permitan una solución viable, sino que el orden en el que se distribuyen y organizan las materias debe responder a los periodos de tiempo en los que existen una mayor facilidad para el desarrollo de ciertas actividades de acuerdo a la capacidad intelectual y atencional que poseen los estudiantes en función de sus ritmos biológicos a ciertas horas del día (Hederich Martinez, Camargo Uribe, & Reyes Cuervo, 2004).

De esta manera, el documento expone como a través del empleo de tres diferentes técnicas evolutivas es posible dar solución al problema planteado. La razón de implementar tres técnicas surge de la necesidad de comparar las respuestas en función del nivel de optimización, dado que el enfoque planteado se aparta de los estándares convencionales de programación de horarios escolares que pueden hallarse en la literatura científica, los cuales aunque comprenden la aplicación de diferentes técnicas, en ningún caso atienden alguna condición referente al desempeño cognitivo de los estudiantes, lo cual impide la realización de una comparación ecuánime con alguno de estos métodos.

El modelo de solución desarrollado en esta investigación se asienta sobre la técnica de los algoritmos genéticos. Obviamente existen trabajos previos en esta área y en este mismo tipo de problema, como los presentados por Abramson y Abela, (1992), Colorni et al. (1998) y Beligiannis et al. (2009). Sin embargo, estos modelos trabajan el mismo tipo de consideraciones generales, difiriendo en aspectos más instrumentales como la codificación de

la información, el mecanismo de funcionamiento de los operadores genéticos y el modo aplicación de los mismos, y el planteamiento de la función objetivo bajo el cual miden su nivel de optimización. En este sentido, el problema siempre ha tenido orientación más del lado computacional que operativo, razón por la cual los enfoques previos difieren ampliamente del presente, pues la alternativa presentada se centra en lograr que la programación realizada tenga una consecuencia práctica de su aplicación, la cual es aprovechar la ubicación de las asignaturas en función de los ritmos de eficiencia cognitiva de los estudiantes.

Las tres técnicas implementadas comprenden diferentes niveles de complejidad, acogiendo los parámetros y métricas establecidas, no sólo en la búsqueda de una solución válida, sino también con un nivel alto de optimización.

La técnica más compleja aplicada al problema es el algoritmo NSGA-II, el cual posee una estructura especializada en sus operadores genéticos y método de evolución. Aunque no se trata de un nuevo algoritmo, pues ha sido aplicado a problemas en sistemas de comunicaciones (Carranza, Chávez, Fissore, & Simón, 2008) y de producción (Ruiz, Castrillón, & Sarache, 2012), entre otros, no existen referentes que vinculen ésta técnica al problema de programación de horarios escolares. Su implementación requirió el desarrollo de un sistema de codificación que explotará al máximo sus cualidades, la cual se da en dos etapas, inicialmente en hipermatrices y posteriormente en bloques, mecanismo que también es un diseño propio dentro del proceso de adaptación desarrollado.

El algoritmo genético simple, aunque no posee operadores tan especializados como el método anterior, se basa en los principios generales de funcionamiento de los algoritmos genéticos, permitiendo la consecución de soluciones con un buen nivel de optimización. El algoritmo genético simple facilita su fácil comprensión e implementación pues se basa en subrutinas que se aplican iterativamente y que son comunes en muchos de los artículos relacionados con el tema.

Por último, el algoritmo de búsqueda aleatoria es una representación de una asignación académica basada sólo en las restricciones duras del problema. La razón de utilizar esta técnica es la equiparación con un modelo convencional de programación de horarios escolares, en las que no hay vínculo alguno con el criterio pedagógico seleccionado. Los resultados de este algoritmo aluden sólo a la complejidad básica del problema, eludiendo el nivel de dificultad computacional que impone la variable añadida como enfoque establecido en esta investigación. No obstante, las soluciones obtenidas bajo esta técnica se evalúan bajo el mismo modelo matemático empleado con las soluciones en las técnicas anteriores. Dentro de este contexto, el nivel de

optimización alcanzado por este algoritmo no se equipara con el alcanzado por el NSGA-II o el algoritmo genético simple, precisamente porque este no se encuentra adaptado para responder de manera concreta a los requerimientos de problema, abordándolo sólo de manera general, tal como lo haría cualquier otra técnica.

Todo lo anterior permite destacar los siguientes hechos como elementos diferenciadores de la investigación realizada:

- La inclusión de una variable de orden pedagógico dentro del proceso de programación de horarios escolares.
- La aplicabilidad de técnicas inteligentes a problemas vinculados no sólo a variables instrumentales, sino a elementos de orden más cualitativo dentro de los problemas de programación de horarios escolares.
- El desarrollo de un modelo matemático orientado al cumplimiento de las restricciones duras y blandas que el planteamiento del problema establece, con criterios que facilitan el proceso de optimización de las técnicas implementadas para la solución.
- La adaptación del algoritmo NSGA-II para la solución del problema de programación de horarios bajo el enfoque establecido, para lo cual se implemento un sistema especial de codificación en bloques para la obtención de soluciones con un alto grado de optimización.
- La verificación de la eficiencia del algoritmo propuesto mediante la implementación de otras dos técnicas inteligentes de diferente complejidad, que permitieran medir el nivel de optimización alcanzado por el NSGA-II
- La puesta a prueba del modelo de programación horaria desarrollado en una institución escolar pública, encontrando resultados favorables en la disminución de la mortalidad académica de los grupos con una programación acorde a los ritmos de aprendizaje de los estudiantes.

El proceso desarrollado muestra la reevaluación del problema de programación de horarios al establecer criterios vinculados con los estudiantes y no sólo con la infraestructura, lo cual favorece la programación de horarios particulares para la consecución de rendimientos óptimos en la asimilación de contenidos (De La Cruz Nogales, 2009). De igual manera, aunque los algoritmos genéticos han sido aplicados a este problema, no existe referencia alguna del NSGA-II como técnica de solución. En este sentido, la adaptación realizada permite explotar de manera eficiente el proceso de evolución del algoritmo, no atacando el problema de manera general, sino ofreciendo una construcción de solución sobre cada grupo, de manera que se pueda obtener el mejor resultado posible en cada uno de ellos y consecuentemente en la solución total. Cabe añadir, que el proceso recursivo del algoritmo permite que se evite al máximo la consecución de soluciones sobre óptimos locales, para lo cual tiene la facultad

de reiniciar el proceso en caso tal de que se identifique tal situación.

A nivel escolar es importante resaltar, que aunque el proceso educativo esta circundado por una amplia cantidad de condiciones socioculturales, el modelo educativo, las particularidades propias de cada docente, el ambiente y la situación psicosocial y emocional de los estudiantes (Maciques Rodríguez, 2004), es posible afirmar que un tipo de programación horaria acorde a las características planteadas por esta investigación puede tener un alto grado de incidencia en la respuesta académica de los estudiantes, dada la tendencia que se observa en los resultados experimentales.

6. Conclusiones

A nivel El desarrollo del documento muestra el cumplimiento del objetivo general y los objetivos específicos dispuestos como eje de la presente investigación.

Se estableció un marco teórico en el que se diferenciaron las principales corrientes existentes para el problema de programación de horarios, haciendo énfasis en la programación de horarios educativos y centrándose finalmente en los horarios escolares.

La revisión literaria permitió identificar las principales técnicas computacionales utilizadas para la solución del problema, para lo cual se relacionaron características y antecedentes de las mismas, así como las principales restricciones de los enfoques actuales. En este mismo sentido, se identificaron los referentes teóricos que indican la forma en que la programación horaria de la jornada escolar tiene relación con el desempeño de los estudiantes, aspecto enmarcado dentro de las necesidades fisiológicas de los individuos, en el cual se establecen ritmos para los que ciertos momentos de la jornada permiten de una mejor eficiencia cognitiva.

Todo lo anterior brindó el soporte conceptual que demuestra la pertinencia del enfoque seleccionado, el cual se aparta de los modelos tradicionales y se enmarca en un nivel de complejidad superior para la consecución de una solución válida, al considerar una variable de orden pedagógico dentro del proceso de programación de horarios escolares. También es necesario resaltar la adaptación de una técnica altamente especializada como el NSGA-II, de la cual no existe referente que exponga su aplicación a este tipo de problemas. De esta manera, existe una justificación estructurada que comprueba la innovación del enfoque y de la metodología propuesta.

Dada la dificultad de contraste con otras técnicas existentes debido a las características del enfoque, además del NSGA-II, se implementaron otros dos algoritmos mediante los cuales se pudiera comparar el nivel de optimización logrado por cada una de ellos en la consecución de una programación horaria acorde a los ritmos de aprendizaje. Para ello se diseñó un sistema de codificación basado en hipermatrices, las cuales permiten concentrar la información de los grupos de manera eficiente para la evaluación de las restricciones y demás procesos en los que se manipulan los datos.

Adicionalmente, para el NSGA-II, fue diseñado un sistema especial de codificación en bloques, para potencializar sus características en la exploración del espacio de búsqueda y por ende en la consecución de mejores soluciones.

Las restricciones generales del problema, vinculadas a la asignación sin cruces de materias, salones y profesores, fueron adaptadas para cumplir con los criterios legales de la escuela pública en Colombia. Las restricciones de optimización se fijaron de acuerdo a sugerencias propuestas dentro de la literatura consultada y algunos ajustes en función de las características de la jornada. Estas restricciones se plasmaron en un modelo matemático que es usado por todas las técnicas para verificar la validez de las soluciones y el nivel de optimización de las mismas.

La forma como se aplican las técnicas al problema se explica detenidamente mediante diagramas de flujo y ejemplos que facilitan su comprensión, de manera que puedan ser utilizados en contextos académicos para socialización o replica de los mismos.

A nivel computacional, el algoritmo NSGA-II permite la consecución de soluciones con un mayor grado de optimización respecto a los algoritmos de contraste. De acuerdo a las pruebas realizadas, es posible alcanzar niveles de hasta el 30% y 45% superiores de eficiencia en comparación con el algoritmo genético simple y el algoritmo de búsqueda aleatoria, respectivamente. De hecho, dicha diferencia se da tanto en las soluciones individuales por grupo, como en las soluciones totales para cada problema.

A su vez, el algoritmo genético simple presenta niveles de hasta 22% de eficiencia respecto al algoritmo de búsqueda aleatoria, demostrando que aunque su modelo evolutivo no es tan especializado, presenta una estructura computacional compleja que le permite obtener soluciones con un alto nivel de optimización.

No obstante, respecto al tiempo de ejecución el fenómeno es inverso, siendo el algoritmo de búsqueda aleatoria el que obtiene una solución en el menor tiempo posible, el cual en problemas de pequeña magnitud puede ser hasta 60 veces más rápido que el algoritmo NSGA-II y el algoritmo genético simple.

También es importante destacar que todos los algoritmos presentan un alto nivel de estabilidad en las soluciones que entregan, de manera que entre una y otra ejecución las variaciones en el nivel de optimización no presentan diferencias estadísticamente significativas.

El tipo de programación propuesto fue puesto a prueba dentro de una institución educativa de la ciudad de Manizales. Aunque inicialmente se planteó una comparación con un software comercial, se decidió aplicar dicho contraste con la programación del algoritmo de búsqueda aleatoria, teniendo en cuenta su tipo de respuesta, la cual puede ser equiparable a la conseguida con otras técnicas de solución. Se tomaron dos grupos de cada nivel desde sexto a once, uno con la programación horaria del NSGA-II y el otro con la del algoritmo de búsqueda aleatoria.

La evaluación de los resultados obtenidos durante dos periodos académicos, evidencia una menor mortalidad en el conjunto general de asignaturas para los grupos programados mediante el NSGA-II, hasta en un 28%. En el análisis de dichos resultados se observa como de un periodo académico al otro existe una variación de 6 puntos porcentuales en la reducción de la cantidad de materias pérdidas, lo cual sugiere que en la medida que hay una mayor adaptación de los estudiantes a la jornada establecida y más aun con aquella que atiende al enfoque en los ritmos de aprendizaje, es posible una mejora continua en los resultados de los estudiantes. De acuerdo a estos resultados, es posible afirmar que un tipo de programación horaria acorde a los ritmos cognitivos que presentan los estudiantes durante la jornada escolar, puede tener un alto grado de incidencia en su respuesta académica, mostrando mejores resultados cuanto más tiempo se conservan las condiciones horarias impuestas.

Dadas las restricciones en el tiempo de esta investigación, no es posible examinar la tasa de repitencia dentro de la institución ya que esta medición sólo se puede realizar al finalizar el año lectivo.

Como futuras acciones dentro de esta línea de investigación se propone lo siguiente:

- Aplicar el presente modelo a un conjunto más amplio de instituciones, que permita verificar el impacto que este puede producir en el desempeño académico de los estudiantes. En este sentido sería muy valiosos que dicha prueba también se realizará en instituciones que operen en la jornada de la tarde, con el fin de comprobar sí el patrón de desempeño en esta jornada es similar al presentado por los resultados obtenidos en la jornada de la mañana. Así mismo, sería conveniente la contrastación con instituciones de orden rural, atendiendo las diferencias que este contexto otorga en el desarrollo de la jornada académica.
- Implementación de un conjunto más amplio de técnicas evolutivas para la solución del problema planteado, a través de un sistema iterativo en el que sea posible la alternación entre modelos, buscando la consecución de soluciones óptimas. Así mismo, profundizar en las consideraciones planteadas en el párrafo

anterior, como elementos a considerar dentro de las
restricciones a tener en cuenta.

A. Anexo: Tablas de resultado para pruebas de estabilidad del sistema y comparación de tiempos de ejecución

A continuación se presentan los resultados de las pruebas de estabilidad para los 6 problemas planteados, junto con el análisis descriptivo de los mismos. Las tablas se presentan organizadas por método para facilitar su visualización.

Tabla A-1. Resultado Fitness por grupo, repetición y problema para algoritmo de búsqueda aleatoria

Método	Problema	Grupo.	Fitness por repetición.									
			1	2	3	4	5	6	7	8	9	10
Aleatorio	1	1	60	96	69	78	111	75	87	69	96	90
		2	114	81	102	111	78	75	87	78	99	75
		3	75	78	96	90	72	81	105	72	72	96
		4	90	81	87	78	33	93	93	66	66	60
	2	1	66	99	114	108	120	96	81	108	87	87
		2	111	96	78	108	78	81	69	75	102	75
		3	81	75	93	54	81	96	108	60	33	93
		4	111	87	93	114	78	60	78	63	84	108
		5	108	72	66	108	87	99	90	102	81	60
	3	1	81	69	81	96	81	72	63	72	99	111
		2	84	45	96	78	75	72	114	81	60	93
		3	75	105	39	99	99	75	75	90	108	66
		4	93	78	99	111	96	87	90	69	48	114
		5	117	84	93	81	78	87	102	84	108	54
		6	93	102	78	90	63	69	81	87	90	75
	4	1	114	96	126	66	120	96	78	69	123	114
		2	75	75	84	63	57	63	24	78	63	93
		3	78	108	72	75	66	75	99	54	108	78
		4	102	72	75	78	69	75	57	84	48	99
		5	84	84	69	93	93	87	87	78	96	81
		6	84	66	72	93	96	78	78	84	105	72
		7	93	72	102	39	66	99	78	96	102	90
	5	1	84	102	111	111	87	93	102	78	99	48
		2	63	69	81	69	54	90	78	84	81	81
		3	99	93	84	66	48	99	105	75	99	90
		4	90	96	51	105	90	90	87	108	51	75
		5	108	81	96	69	75	63	63	66	126	99
		6	114	60	72	75	60	84	99	90	87	84

Método	Problema	Grupo.	Fitness por repetición.									
			1	2	3	4	5	6	7	8	9	10
		7	78	105	69	84	87	87	93	93	60	90
		8	63	96	87	117	90	84	87	78	90	108
	6	1	111	93	63	84	96	96	66	54	57	90
		2	90	60	81	63	78	81	63	90	78	66
		3	99	84	117	57	72	84	81	87	99	87
		4	75	81	75	114	108	117	117	60	87	84
		5	96	66	93	81	90	84	93	81	66	105
		6	96	90	57	99	57	75	69	69	99	90
		7	96	54	108	84	87	63	60	99	114	75
		8	96	96	75	87	87	102	90	108	90	99
		9	108	78	102	102	93	60	114	117	99	48

Tabla A-2. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo de búsqueda aleatoria.

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
Aleatorio	1	1	84,75	82,50	23,027	60	114
		2	84,00	81,00	8,124	78	96
		3	88,50	91,50	14,387	69	102
		4	89,25	84,00	15,564	78	111
		5	73,50	75,00	31,984	33	111
		6	81,00	78,00	8,485	75	93
		7	93,00	90,00	8,485	87	105
		8	71,25	70,50	5,123	66	78
		9	83,25	84,00	16,681	66	99
		10	80,25	82,50	16,132	60	96
	2	1	95,40	108,00	20,720	66	111
		2	85,80	87,00	12,112	72	99
		3	88,80	93,00	18,075	66	114
		4	98,40	108,00	24,956	54	114
		5	88,80	81,00	17,824	78	120
		6	86,40	96,00	16,349	60	99
		7	85,20	81,00	14,789	69	108
		8	81,60	75,00	22,188	60	108
		9	77,40	84,00	26,102	33	102
		10	84,60	87,00	18,174	60	108
	3	1	90,50	88,50	14,748	75	117
		2	80,50	81,00	22,242	45	105
		3	81,00	87,00	22,208	39	99
		4	92,50	93,00	12,211	78	111
		5	82,00	79,50	13,506	63	99
		6	77,00	73,50	7,975	69	87
		7	87,50	85,50	18,534	63	114
		8	80,50	82,50	8,361	69	90
		9	85,50	94,50	25,579	48	108
		10	85,50	84,00	24,501	54	114
	4	1	90,00	84,00	13,964	75	114
		2	81,86	75,00	15,170	66	108
		3	85,71	75,00	21,061	69	126
		4	72,43	75,00	18,849	39	93
		5	81,00	69,00	22,583	57	120
		6	81,86	78,00	12,812	63	99
7		71,57	78,00	24,460	24	99	
8		77,57	78,00	13,240	54	96	

Estadísticos								
Valor Fitness								
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo	
		9	92,14	102,00	26,705	48	123	
		10	89,57	90,00	14,223	72	114	
	5		1	87,38	87,00	19,138	63	114
			2	87,75	94,50	16,175	60	105
			3	81,38	82,50	18,102	51	111
			4	87,00	79,50	20,846	66	117
			5	73,88	81,00	17,415	48	90
			6	86,25	88,50	10,607	63	99
			7	89,25	90,00	13,864	63	105
			8	84,00	81,00	12,928	66	108
			9	86,63	88,50	23,537	51	126
			10	84,38	87,00	17,960	48	108
	6		1	96,33	96,00	10,332	75	111
			2	78,00	81,00	14,925	54	96
			3	85,67	81,00	20,573	57	117
			4	85,67	84,00	18,069	57	114
			5	85,33	87,00	14,782	57	108
			6	84,67	84,00	18,235	60	117
			7	83,67	81,00	21,448	60	117
			8	85,00	87,00	21,319	54	117
9			87,67	90,00	17,986	57	114	
10			82,67	87,00	17,436	48	105	

Tabla A-3. Resultado Fitness por grupo, repetición y problema para algoritmo genético simple.

Método	Problema	Grupo.	Fitness por repetición.										
			1	2	3	4	5	6	7	8	9	10	
Genético Simple	1	1	54	69	57	42	48	51	66	63	75	69	
		2	63	81	78	81	57	60	63	75	63	66	
		3	63	72	102	72	93	66	75	75	63	66	
		4	48	51	42	66	54	63	57	63	66	57	
	2		1	78	39	54	36	72	69	66	42	60	48
			2	78	72	72	87	66	75	48	78	48	45
			3	51	72	81	75	81	69	78	81	81	102
			4	57	69	54	39	36	51	69	45	42	60
			5	63	51	60	42	54	51	63	48	57	48
	3		1	33	75	57	36	48	96	66	30	57	69
			2	72	72	66	72	72	36	60	87	81	69
			3	102	72	66	84	96	90	81	66	57	75
			4	45	54	30	54	54	69	60	60	45	57
			5	54	69	54	54	69	57	48	69	63	54
			6	81	69	57	87	108	75	60	78	81	87
	4		1	51	60	57	30	51	57	66	69	48	57
			2	78	66	57	81	54	69	78	72	87	66
			3	108	75	96	69	105	102	105	66	72	72
			4	54	63	60	69	60	60	57	48	51	51
			5	63	60	54	45	51	60	42	54	51	63
6			93	81	45	108	33	60	93	90	90	78	
7			54	69	57	57	54	63	66	45	57	57	
5		1	30	45	69	54	42	48	57	42	39	69	
		2	81	75	63	90	69	81	66	69	72	51	

Método	Problema	Grupo.	Fitness por repetición.									
			1	2	3	4	5	6	7	8	9	10
		3	72	99	78	60	81	87	114	81	87	84
		4	60	66	60	63	66	60	42	60	60	60
		5	54	48	63	57	72	72	63	69	48	60
		6	99	84	57	63	102	84	72	69	72	84
		7	63	60	54	57	66	51	42	63	75	81
		8	78	57	84	108	87	63	117	96	63	75
	6	1	42	42	81	90	66	63	81	39	42	48
		2	81	93	60	60	75	51	75	99	72	69
		3	81	81	84	75	72	102	69	42	99	63
		4	54	57	57	60	51	63	54	60	66	60
		5	45	54	54	39	33	60	57	48	51	57
		6	72	93	60	96	54	75	93	81	84	72
		7	63	66	69	57	57	54	60	54	60	66
		8	81	39	81	81	99	78	90	84	84	93
		9	87	102	96	90	93	81	66	102	90	72

Tabla A-4. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo genético simple.

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
AG-Simple	1	1	57,00	58,50	7,348	48	63
		2	68,25	70,50	12,580	51	81
		3	69,75	67,50	26,082	42	102
		4	65,25	69,00	16,681	42	81
		5	63,00	55,50	20,347	48	93
		6	60,00	61,50	6,481	51	66
		7	65,25	64,50	7,500	57	75
		8	69,00	69,00	6,928	63	75
		9	66,75	64,50	5,679	63	75
		10	64,50	66,00	5,196	57	69
	2	1	65,40	63,00	12,260	51	78
		2	60,60	69,00	14,910	39	72
		3	64,20	60,00	11,925	54	81
		4	55,80	42,00	23,488	36	87
		5	61,80	66,00	17,441	36	81
		6	63,00	69,00	11,225	51	75
		7	64,80	66,00	10,941	48	78
		8	58,80	48,00	19,045	42	81
		9	57,60	57,00	14,910	42	81
		10	60,60	48,00	23,850	45	102
	3	1	64,50	63,00	25,367	33	102
		2	68,50	70,50	7,450	54	75
		3	55,00	57,00	13,236	30	66
		4	64,50	63,00	19,877	36	87
		5	74,50	70,50	23,424	48	108
		6	70,50	72,00	22,025	36	96

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
		7	62,50	60,00	10,803	48	81
		8	65,00	67,50	19,596	30	87
		9	64,00	60,00	14,408	45	81
		10	68,50	69,00	12,062	54	87
	4	1	71,57	63,00	22,142	51	108
		2	67,71	66,00	7,910	60	81
		3	60,86	57,00	16,222	45	96
		4	65,57	69,00	25,245	30	108
		5	58,29	54,00	22,239	33	105
		6	67,29	60,00	15,766	57	102
		7	72,43	66,00	21,454	42	105
		8	63,43	66,00	15,725	45	90
		9	65,14	57,00	17,809	48	90
		10	63,43	63,00	9,396	51	78
	5	1	67,13	67,50	20,594	30	99
		2	66,75	63,00	18,406	45	99
		3	66,00	63,00	10,392	54	84
		4	69,00	61,50	19,376	54	108
		5	73,13	70,50	17,635	42	102
		6	68,25	67,50	15,021	48	87
		7	71,63	64,50	29,105	42	117
		8	68,63	69,00	15,666	42	96
		9	64,50	67,50	15,464	39	87
		10	70,50	72,00	12,524	51	84
	6	1	67,33	72,00	16,978	42	87
		2	69,67	66,00	23,420	39	102
		3	71,33	69,00	14,680	54	96
		4	72,00	75,00	19,092	39	96
		5	66,67	66,00	20,827	33	99
		6	69,67	63,00	16,000	51	102
7		71,67	69,00	14,107	54	93	
8		67,67	60,00	24,285	39	102	
9		72,00	72,00	18,914	42	99	
10		66,67	66,00	12,530	48	93	

Tabla A-5. Resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II.

Método	Problema	Grupo.	Fitness por repetición.									
			1	2	3	4	5	6	7	8	9	10
NSGA-II	1	1	18	18	18	21	15	15	12	24	15	24
		2	51	48	36	45	54	60	51	54	36	30
		3	96	84	42	66	45	87	78	57	78	105
		4	36	27	54	33	51	48	27	54	54	45
	2	1	24	24	15	21	18	27	24	18	15	15
		2	30	48	51	36	48	33	57	78	45	45
		3	57	63	51	57	78	105	72	84	69	78

Método	Problema	Grupo.	Fitness por repetición.									
			1	2	3	4	5	6	7	8	9	10
		4	30	27	45	21	24	33	39	21	36	30
		5	42	24	36	33	33	57	30	18	24	21
	3	1	21	30	27	18	27	18	24	18	39	21
		2	39	42	54	36	45	48	30	54	42	51
		3	102	78	57	84	81	84	63	96	84	84
		4	36	39	30	39	27	36	48	33	39	36
		5	54	27	21	33	30	30	27	39	27	21
		6	66	84	69	57	66	69	72	45	45	51
	4	1	24	21	21	15	21	15	24	15	18	18
		2	33	54	48	48	39	54	51	51	54	42
		3	54	75	69	72	105	81	96	90	63	45
		4	48	36	39	36	51	54	48	51	42	54
		5	24	27	30	42	30	51	45	24	30	27
		6	60	54	78	42	69	51	84	84	60	69
		7	36	54	39	48	48	36	51	42	48	45
	5	1	15	18	21	27	21	15	27	21	21	21
		2	33	33	48	72	45	63	45	60	72	18
		3	63	81	84	57	72	93	81	78	63	48
		4	24	42	39	30	18	36	24	60	42	45
		5	30	18	24	54	30	27	42	30	30	30
		6	63	45	69	84	66	57	45	81	54	51
		7	36	42	66	42	48	42	45	30	60	54
		8	48	48	72	60	90	51	87	63	69	81
	6	1	30	27	15	27	12	15	21	27	24	27
		2	45	51	51	30	66	57	45	51	51	45
		3	75	72	87	78	51	87	78	84	63	87
		4	42	39	42	78	51	24	39	27	36	33
		5	21	33	45	69	36	33	27	24	36	21
		6	48	45	51	57	66	57	66	60	90	78
		7	33	36	48	45	69	39	36	54	48	36
		8	93	78	57	72	51	42	69	69	90	72
		9	96	69	60	78	69	78	57	84	96	69

Tabla A-6. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
NSGA-II	1	1	50,25	43,50	33,350	18	96
		2	44,25	37,50	29,330	18	84
		3	37,50	39,00	15,000	18	54
		4	41,25	39,00	19,190	21	66
		5	41,25	48,00	17,896	15	54
		6	52,50	54,00	29,850	15	87
		7	42,00	39,00	28,879	12	78
		8	47,25	54,00	15,564	24	57
		9	45,75	45,00	26,763	15	78
		10	51,00	37,50	37,068	24	105

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
	2	1	36,60	30,00	13,145	24	57
		2	37,20	27,00	17,570	24	63
		3	39,60	45,00	15,060	15	51
		4	33,60	33,00	14,758	21	57
		5	40,20	33,00	23,962	18	78
		6	51,00	33,00	32,311	27	105
		7	44,40	39,00	19,832	24	72
		8	43,80	21,00	34,047	18	84
		9	37,80	36,00	20,849	15	69
		10	37,80	30,00	25,154	15	78
	3	1	53,00	46,50	28,566	21	102
		2	50,00	40,50	24,714	27	84
		3	43,00	42,00	19,504	21	69
		4	44,50	37,50	23,037	18	84
		5	46,00	37,50	22,821	27	81
		6	47,50	42,00	24,914	18	84
		7	44,00	39,00	20,229	24	72
		8	47,50	42,00	26,659	18	96
		9	46,00	40,50	19,596	27	84
		10	44,00	43,50	23,749	21	84
	4	1	39,86	36,00	14,358	24	60
		2	45,86	54,00	18,792	21	75
		3	46,29	39,00	20,557	21	78
		4	43,29	42,00	16,958	15	72
		5	51,86	48,00	28,074	21	105
		6	48,86	51,00	20,029	15	81
		7	57,00	51,00	24,617	24	96
		8	51,00	51,00	28,036	15	90
		9	45,00	48,00	16,340	18	63
		10	42,86	45,00	16,767	18	69
	5	1	39,00	34,50	17,566	15	63
		2	40,88	42,00	19,960	18	81
		3	52,88	57,00	23,400	21	84
		4	53,25	55,50	19,689	27	84
		5	48,75	46,50	25,695	18	90
		6	48,00	46,50	24,054	15	93
		7	49,50	45,00	22,904	24	87
		8	52,88	60,00	22,956	21	81
		9	51,38	57,00	18,593	21	72
		10	43,50	46,50	20,473	18	81
	6	1	53,67	45,00	27,595	21	96
		2	50,00	45,00	18,675	27	78
		3	50,67	51,00	18,822	15	87
		4	59,33	69,00	20,664	27	78
		5	52,33	51,00	18,802	12	69
		6	48,00	42,00	23,953	15	87
		7	48,67	45,00	19,831	21	78
		8	53,33	54,00	23,516	24	84
		9	59,33	51,00	26,865	24	96

Estadísticos							
Valor Fitness							
Método	Probl.	Grup	Media	Mediana	Desv. típ.	Mínimo	Máximo
		10	52,00	45,00	24,602	21	87

A continuación se presentan los resultados de las pruebas de normalidad para los datos obtenidos de las pruebas de estabilidad del sistema.

Tabla A-7. Análisis descriptivo de resultado Fitness por grupo, repetición y problema para algoritmo NSGA-II.

Pruebas de normalidad						
Método de solución	Problema	Grupo	Variable	Shapiro-Wilk		
				Estadístico	gl	Sig.
Aleatorio	1	1	Valor Fitness	,986	4	,937
		2	Valor Fitness	,773	4	,062
		3	Valor Fitness	,941	4	,662
		4	Valor Fitness	,838	4	,189
		5	Valor Fitness	,972	4	,851
		6	Valor Fitness	,827	4	,161
		7	Valor Fitness	,827	4	,161
		8	Valor Fitness	,971	4	,850
		9	Valor Fitness	,852	4	,233
		10	Valor Fitness	,950	4	,717
	2	1	Valor Fitness	,804	5	,087
		2	Valor Fitness	,903	5	,424
		3	Valor Fitness	,966	5	,847
		4	Valor Fitness	,638	5	,002
		5	Valor Fitness	,712	5	,013
		6	Valor Fitness	,816	5	,109
		7	Valor Fitness	,949	5	,732
		8	Valor Fitness	,867	5	,254
		9	Valor Fitness	,821	5	,119
		10	Valor Fitness	,994	5	,993
	3	1	Valor Fitness	,893	6	,336
		2	Valor Fitness	,946	6	,707
		3	Valor Fitness	,810	6	,072
		4	Valor Fitness	,963	6	,846
		5	Valor Fitness	,942	6	,673
		6	Valor Fitness	,810	6	,072
		7	Valor Fitness	,987	6	,979
		8	Valor Fitness	,925	6	,540
		9	Valor Fitness	,854	6	,168
		10	Valor Fitness	,929	6	,571
	4	1	Valor Fitness	,922	7	,485
		2	Valor Fitness	,894	7	,297
		3	Valor Fitness	,806	7	,046
		4	Valor Fitness	,926	7	,518
		5	Valor Fitness	,889	7	,269
		6	Valor Fitness	,944	7	,674

Pruebas de normalidad							
Método de solución	Problema	Grupo	Variable	Shapiro-Wilk			
				Estadístico	gl	Sig.	
		7	Valor Fitness	,868	7	,179	
		8	Valor Fitness	,947	7	,704	
		9	Valor Fitness	,888	7	,264	
		10	Valor Fitness	,965	7	,862	
		5	1	Valor Fitness	,941	8	,617
			2	Valor Fitness	,894	8	,254
			3	Valor Fitness	,989	8	,994
			4	Valor Fitness	,854	8	,103
			5	Valor Fitness	,839	8	,074
			6	Valor Fitness	,848	8	,092
	7		Valor Fitness	,934	8	,550	
	8		Valor Fitness	,962	8	,827	
	9		Valor Fitness	,961	8	,817	
	10		Valor Fitness	,929	8	,511	
	6	1	Valor Fitness	,894	9	,217	
		2	Valor Fitness	,932	9	,500	
		3	Valor Fitness	,959	9	,785	
		4	Valor Fitness	,959	9	,789	
		5	Valor Fitness	,965	9	,845	
		6	Valor Fitness	,964	9	,843	
		7	Valor Fitness	,897	9	,233	
		8	Valor Fitness	,972	9	,912	
		9	Valor Fitness	,949	9	,679	
		10	Valor Fitness	,939	9	,576	
	AG-Simple	1	1	Valor Fitness	,860	4	,262
			2	Valor Fitness	,939	4	,650
			3	Valor Fitness	,980	4	,904
			4	Valor Fitness	,926	4	,571
			5	Valor Fitness	,795	4	,094
			6	Valor Fitness	,927	4	,577
7			Valor Fitness	,982	4	,911	
8			Valor Fitness	,729	4	,024	
9			Valor Fitness	,791	4	,086	
10			Valor Fitness	,840	4	,195	
2		1	Valor Fitness	,881	5	,314	
		2	Valor Fitness	,820	5	,117	
		3	Valor Fitness	,873	5	,281	
		4	Valor Fitness	,822	5	,121	
		5	Valor Fitness	,964	5	,837	
		6	Valor Fitness	,817	5	,111	
		7	Valor Fitness	,955	5	,775	
		8	Valor Fitness	,791	5	,069	
		9	Valor Fitness	,934	5	,623	
		10	Valor Fitness	,731	5	,020	
3		1	Valor Fitness	,977	6	,934	
		2	Valor Fitness	,769	6	,030	
		3	Valor Fitness	,803	6	,062	
		4	Valor Fitness	,927	6	,560	
		5	Valor Fitness	,934	6	,615	
		6	Valor Fitness	,965	6	,855	

Pruebas de normalidad							
Método de solución	Problema	Grupo	Variable	Shapiro-Wilk			
				Estadístico	gl	Sig.	
		7	Valor Fitness	,899	6	,368	
		8	Valor Fitness	,917	6	,486	
		9	Valor Fitness	,891	6	,324	
		10	Valor Fitness	,949	6	,730	
	4	1	Valor Fitness	,875	7	,203	
		2	Valor Fitness	,909	7	,392	
		3	Valor Fitness	,715	7	,005	
		4	Valor Fitness	,980	7	,958	
		5	Valor Fitness	,777	7	,024	
		6	Valor Fitness	,652	7	,001	
		7	Valor Fitness	,974	7	,924	
		8	Valor Fitness	,944	7	,677	
		9	Valor Fitness	,843	7	,105	
		10	Valor Fitness	,968	7	,880	
	5	1	Valor Fitness	,980	8	,965	
		2	Valor Fitness	,953	8	,746	
		3	Valor Fitness	,920	8	,428	
		4	Valor Fitness	,748	8	,008	
		5	Valor Fitness	,961	8	,821	
		6	Valor Fitness	,925	8	,471	
		7	Valor Fitness	,849	8	,093	
		8	Valor Fitness	,949	8	,696	
		9	Valor Fitness	,965	8	,856	
		10	Valor Fitness	,908	8	,340	
	6	1	Valor Fitness	,887	9	,186	
		2	Valor Fitness	,923	9	,416	
		3	Valor Fitness	,911	9	,326	
		4	Valor Fitness	,934	9	,524	
		5	Valor Fitness	,971	9	,907	
		6	Valor Fitness	,923	9	,421	
		7	Valor Fitness	,940	9	,578	
		8	Valor Fitness	,903	9	,270	
		9	Valor Fitness	,969	9	,888	
		10	Valor Fitness	,937	9	,549	
	NSGA-II	1	1	Valor Fitness	,943	4	,674
			2	Valor Fitness	,925	4	,564
			3	Valor Fitness	,982	4	,911
			4	Valor Fitness	,981	4	,909
			5	Valor Fitness	,804	4	,111
			6	Valor Fitness	,991	4	,965
7			Valor Fitness	,973	4	,863	
8			Valor Fitness	,708	4	,014	
9			Valor Fitness	,997	4	,991	
10			Valor Fitness	,821	4	,145	
2		1	Valor Fitness	,892	5	,367	
		2	Valor Fitness	,815	5	,106	
		3	Valor Fitness	,836	5	,155	
		4	Valor Fitness	,871	5	,271	
		5	Valor Fitness	,908	5	,457	

Pruebas de normalidad							
Método de solución	Problema	Grupo	Variable	Shapiro-Wilk			
				Estadístico	gl	Sig.	
		6	Valor Fitness	,795	5	,074	
		7	Valor Fitness	,938	5	,654	
		8	Valor Fitness	,734	5	,021	
		9	Valor Fitness	,965	5	,841	
		10	Valor Fitness	,898	5	,397	
		3	1	Valor Fitness	,937	6	,634
			2	Valor Fitness	,831	6	,110
			3	Valor Fitness	,901	6	,381
			4	Valor Fitness	,921	6	,513
			5	Valor Fitness	,848	6	,152
	6		Valor Fitness	,955	6	,781	
	7		Valor Fitness	,885	6	,294	
	8		Valor Fitness	,901	6	,378	
	9		Valor Fitness	,776	6	,035	
	10		Valor Fitness	,895	6	,348	
	4	1	Valor Fitness	,915	7	,431	
		2	Valor Fitness	,929	7	,543	
		3	Valor Fitness	,932	7	,566	
		4	Valor Fitness	,927	7	,528	
		5	Valor Fitness	,917	7	,447	
		6	Valor Fitness	,918	7	,453	
		7	Valor Fitness	,897	7	,312	
		8	Valor Fitness	,929	7	,541	
		9	Valor Fitness	,941	7	,647	
		10	Valor Fitness	,967	7	,874	
	5	1	Valor Fitness	,925	8	,472	
		2	Valor Fitness	,886	8	,215	
		3	Valor Fitness	,926	8	,479	
		4	Valor Fitness	,962	8	,830	
		5	Valor Fitness	,947	8	,685	
		6	Valor Fitness	,973	8	,919	
		7	Valor Fitness	,846	8	,086	
		8	Valor Fitness	,889	8	,229	
		9	Valor Fitness	,921	8	,438	
		10	Valor Fitness	,936	8	,572	
	6	1	Valor Fitness	,886	9	,182	
		2	Valor Fitness	,909	9	,306	
		3	Valor Fitness	,914	9	,342	
		4	Valor Fitness	,840	9	,057	
		5	Valor Fitness	,837	9	,054	
6		Valor Fitness	,959	9	,793		
7		Valor Fitness	,956	9	,752		
8		Valor Fitness	,897	9	,233		
9		Valor Fitness	,898	9	,243		
10		Valor Fitness	,907	9	,293		

*. Este es un límite inferior de la significación verdadera.

Tabla A-8. Tiempos de ejecución por método y por repetición

Método	Probl.	Tiempo de ejecución [Seg] X Repetición									
		1	2	3	4	5	6	7	8	9	10
NSGA-II	1	265,92	267,66	335,33	260,63	268,95	273,21	273,44	348,34	436,71	355,43
	2	420,89	426,1	357,99	363,71	332,49	468,8	337,35	336,3	332	331,97
	3	472,32	426,67	459,29	578,69	412,73	553,88	679,12	655,22	615,4	627,15
	4	806,67	498,65	473,15	976	1073,52	633,33	979,5	800,32	812,46	711,89
	5	1089,32	839,24	614,6	545,84	613,67	608,42	743,94	546,83	541,68	541,7
	6	756,19	754,37	915,84	1254,71	695,83	609,84	882,95	832,19	683,3	969,23
Genético Simple	1	293,62	291,02	288,55	289,42	365,65	364,66	290,13	288,42	289,36	288,84
	2	439,97	360,88	364,01	362,93	366,92	441,06	366,19	364,35	364,45	366,76
	3	442,91	443,32	528,15	440,96	440,64	442,86	522,6	513,8	522,79	524,79
	4	516,8	675,88	575,19	657,6	608,3	768,11	616,46	545,07	530,33	530,61
	5	615,69	702,66	620,77	616,52	617,33	698,85	695,24	775,34	772,55	617,62
	6	774,16	695,05	699,31	695,52	777,48	857,83	1485,87	696,5	936,11	696,36
Aleatorio	1	5,84	5,4	5,21	5,26	5,17	5,18	5,04	5,05	4,7	4,61
	2	7,02	5,91	5,48	5,39	4,97	5,22	5,14	6,58	5,58	5,19
	3	6,91	6,52	6,42	8,69	13,04	5,93	6,91	5,55	6,01	6,92
	4	11,8	19,75	5,91	6,21	10,21	10,99	7,62	8,33	6,22	13,85
	5	40,84	48,38	54,8	55,47	9,87	50,51	7,13	50,44	43,44	122,69
	6	345,12	293,19	7,12	480,39	11,31	715,83	75,25	18,86	33,3	127,97

Tabla A-9. Normalidad de tiempo de ejecución por método

Método		Kolmogorov-Smirnov ^a		
		Estadístico	gl	Sig.
Búsqueda Aleatoria	Tiempo de ejecución [Seg]	,361	60	,000
Genético Simple	Tiempo de ejecución [Seg]	,109	60	,073
NSGA-II	Tiempo de ejecución [Seg]	,096	60	,200*

Tabla A-10. Comparación tiempo de ejecución por método y por problema mediante Kruskal Wallis

Problema	Tiempo de ejecución [Seg]		
	Chi-cuadrado	gl	Sig. asintót.
1	20,015	2	,000
2	20,604	2	,000
3	20,392	2	,000
4	21,099	2	,000
5	20,604	2	,000
6	16,423	2	,000

Tabla A-11. Comparación tiempo de ejecución por método y problema entre pares

Problema		Aleatorio Vs Genético Simple		Aleatorio Vs NSGA-II		Genético simple Vs NSGA-II	
		U de Mann-Whitney	Sig. asintót. (bilateral)	U de Mann-Whitney	Sig. asintót. (bilateral)	U de Mann-Whitney	Sig. asintót. (bilateral)
Tiempo de ejecución [Seg]	1	0,000	,000	0,000	,000	34,000	,226
	2	0,000	,000	0,000	,000	28,000	,096
	3	0,000	,000	0,000	,000	30,000	,131
	4	0,000	,000	0,000	,000	24,000	,049
	5	0,000	,000	0,000	,000	28,000	,096
	6	5,000	,001	3,000	,000	47,000	,821

B. Anexo: Productos Académicos

En este anexo se presentan los productos académicos obtenidos a partir de esta investigación:

- Artículo: Programación de horarios escolares basados en ritmos cognitivos usando un algoritmo NSGA-II. Revista Información Tecnológica 24(1), en prensa (2013). Revista categoría A1.
- Artículo: Assignment of school class schedules based on students learning rhythms using a genetic algorithm. Revista ingeniería y ciencia. Universidad EAFIT. Artículo en proceso de evaluación. Revista categoría B.
- Ponencia: Diseño de una metodología basada en técnicas inteligentes para la distribución de procesos académicos en ambientes de trabajo job shop. 5th International Conference on Industrial Engineering and Industrial Management. Del 7 al 9 de Septiembre de 2011, Cartagena - España.
- Ponencia: Metodología basada en partículas inteligentes para la optimización de la producción en ambientes job shop. Conferencia Iberoamericana de Complejidad, Informática y Cibernética. Del 27 al 30 de marzo de 2012, Orlando, Florida - USA
- Registro lógico de software.



Información Tecnológica

Centro de Información Tecnológica (CIT)
c/ Monseñor Subercaseaux 667
La Serena - Chile <http://www.citrevistas.cl>

Incluida en
SciELO Chile

La Serena - Chile, 04 de Octubre de 2012

Omar D. Castrillón
Universidad Nacional de Colombia
Manizales-Colombia

Estimado(a) autor(a):

Informamos a Ud. que el artículo titulado "*Programación de Horarios Escolares Basados en Ritmos Cognitivos usando un Algoritmo NSGA-II*" de los autores Víctor F. Suarez, Álvaro Guerrero y Omar D. Castrillón, ha sido aceptado para publicación en nuestra revista internacional "*INFORMACIÓN TECNOLÓGICA*", y será publicado en el volumen 24 número 1 (Enero-Febrero) del año 2013.

Si debe citar el trabajo en otras publicaciones puede hacerlo colocando: Autores, título, Inf. Tecnol. 24(1), en prensa (2013)

Agradeciendo su interés en nuestra revista, le saluda muy atentamente,



Dr. José O. Valderrama
Editor

PROGRAMACIÓN DE HORARIOS ESCOLARES BASADOS EN RITMOS COGNITIVOS USANDO UN ALGORITMO NSGA-II

Victor F. Suarez., Álvaro Guerrero., Omar D. Castrillón*.

Universidad Nacional de Colombia, Manizales.

vfsuarezc@unal.edu.co, aguerreroa@unal.edu.co, odcastrillong@unal.edu.co

RESUMEN

Se plantea la solución del problema de programación de horarios escolares bajo las características de la escuela pública Colombiana, estableciendo como restricciones la asignación de salones y docentes, e incluyendo como factor primordial dentro del proceso de optimización, la ubicación horaria de las asignaturas en función los ritmos cognitivos que presentan los estudiantes en la jornada escolar. Se propone como método de solución el algoritmo NSGA-II, a través del cual se busca la minimización de una función de evaluación que representa la calidad de la misma de acuerdo al enfoque establecido. Los resultados muestran una mayor eficiencia del algoritmo en comparación con otros aplicados al mismo problema y evaluados con la misma métrica. A nivel experimental, la metodología evidencia que los grupos programados mediante la orientación descrita presentan una reducción en los niveles de mortalidad académica en comparación con una programación horaria que sólo atiende las condiciones de salones y docentes.

Palabras claves: Programación de horarios escolares, NSGA-II, ritmos de cognitivos, restricciones, jornada escolar.

SCHOOL SCHEDULE PROGRAMMING BASED ON COGNITIVE RHYTHMS USING NSGA-II ALGORITHM

ABSTRACT

We propose solving the school timetabling problem under the characteristics of the Colombian public schools, establishing as constraints the allocation of rooms and teachers, and including as a key factor in the optimization process, the location time of the subjects according to the cognitive rhythms that students present on the day. The algorithm NSGA-II is proposed as a solution method, through which seeks minimizing an evaluation function that represents the quality according to of the approach established. The results show a higher efficiency compared to other algorithm applied to the same problem and evaluated with the same metric. For the experiment, is evident that programmed groups through the methodology proposed, describe a reduction in academic mortality rates compared with timetabling that only addresses the conditions of classrooms and teachers.

Keywords: School Timetabling, NSGA-II, Cognitive Rhythms, Constraints, School Day.

INTRODUCCION

Según Wren (1996), el problema de programación de horarios consiste en la disposición de objetos dentro de un patrón en el tiempo o el espacio, de tal manera que se logre el cumplimiento de unos objetivos determinados y que las restricciones sobre la forma en que se deben disponer dichos objetos se satisfagan. Es así como se identifican tres elementos esenciales dentro de cualquier proceso de programación: objetos, patrones y restricciones. Los objetos se refieren al recurso a organizar o disponer, los cuales pueden ser personas, máquinas, vehículos, clases, trabajos en una fábrica, etc.; el patrón es el orden que deben seguir los eventos y se crea como parte del proceso de programación; y las restricciones se definen como las relaciones tangibles o intangibles entre objetos y representan las reglas que se deben seguir para la construcción de un patrón. En el contexto educativo, este problema se enmarca en la disposición de asignaturas, estudiantes, profesores y salones, dentro de periodos de tiempo establecidos (jornadas, semanas, trimestres, semestres), los cuales están sujetos a unas condiciones instauradas de acuerdo a las características de la organización (colegios, universidades) y al tipo de evento que se desea programar (jornada, exámenes, uso de instalaciones).

La programación de horarios académicos generalmente se dirige a tres contextos: horarios universitarios, horarios escolares y horarios para exámenes. El presente artículo se centra en el problema de programación de horarios a nivel escolar. La asignación de horarios escolares tiene un enfoque en las clases y no en los estudiantes, por lo que la jornada diaria (mañana o tarde) se programa de forma continua, dividida en periodos de tiempos preestablecidos, para el desarrollo de actividades académicas en un número dado de asignaturas. Estas asignaturas se asocian a un nivel o grado respectivo, razón por la cual son cursadas por un mismo grupo de estudiantes, los cuales las toman a modo de curso obligatorio y no por demanda, como sucede con los cursos universitarios. En este sentido, la planta física también cumple con la orientación de clases, ya que los salones se asignan de forma permanente a un mismo grupo y se propicia de manera sistemática la rotación de los docentes por cada una de las aulas, según la distribución del horario y su asignación. No obstante, se deben tener en cuenta la existencia de aulas multipropósito o de uso especial, las cuales también pueden ser incluidas dentro de la programación para ser dispuestas de manera permanente en un horario dado para alguno de los grupos. En cuanto a los docentes, la asignación debe garantizar que un docente no esté en dos grupos diferentes al mismo tiempo.

En una proporción alta de instituciones, la solución a este problema es construida de manera manual, lo cual además de tomar días o semanas de trabajo, no permite establecer cierto tipo de restricciones que optimicen el uso del recurso humano y de infraestructura, al igual que da lugar a errores por traslapes en asignaturas y en la asignación de los docentes (Flores, 2011). Sin embargo, a nivel investigativo se han aplicado diversas técnicas analíticas y heurísticas a la solución del problema de programación de horarios escolares tales como: grafos coloreados (Čangalović y Schreuder, 1991), recocido simulado (Abramson, 1991), algoritmos genéticos (Colomi et al., 1998; Abramson y Abela, 1992; Beligianni et al., 2009), programación con restricciones (Valouxis y Housos, 2003), algoritmos de búsqueda local (Avela et al., 2007), programación lineal (Birbas et al., 2009), búsqueda tabú (Minh et al., 2010) y enjambres de partículas (Tassopoulos y Beligiannis, 2012). Igualmente, es importante resaltar que en la búsqueda de una metodología más integral ha sido necesario analizar la aplicación de estas técnicas inteligentes en una gran variedad de problemas como: Optimización de ambientes Job shop (Ruiz et al., 2012), Optimización de redes eléctricas por algoritmos genéticos (Anaut et al., 2009), tratamiento de emisiones atmosféricas (Pérez et al., 2004), procesos de simulación en degradación aeróbica (Rosa et al., 2004), etc.

Aunque los métodos mencionados contemplan las limitaciones generales del problema, se encuentran diferentes enfoques en los que se establecen restricciones especiales sobre la infraestructura, la inclusión de asignaturas electivas, el balanceo de la carga académica diaria de los docentes y otras condiciones particulares de acuerdo al tipo de modelo académico acogido experimentalmente. No obstante, a pesar de las consideraciones expuestas, ninguno de los métodos aborda de manera explícita la relación existente entre los horarios programados y el proceso pedagógico, dejando de lado el papel vital que este asunto tiene dentro de la gestión del tiempo dentro del aula, la cual está directamente relacionada con el grado de aprendizaje de un estudiante (Murillo, 2008; Cornejo y Redondo, 2007). En este sentido, es necesario aclarar que el proceso cognitivo a nivel escolar depende de cuatro factores relacionados con las necesidades del alumno: las preferencias emocionales, el ambiente, las necesidades sociales y las necesidades fisiológicas (Maciques, 2012). Este artículo se centra en el último factor, el cual además de vincular las necesidades alimenticias y las necesidades de movimiento del individuo, resalta la existencia de una hora óptima del día para desarrollar un trabajo. Por tanto, conocer dicha hora favorece la programación de horarios particulares para la consecución de rendimientos óptimos en la asimilación de contenidos (De La Cruz Nogales, 2009).

Surgen entonces dos preguntas: ¿Cuándo enseñar? y ¿qué enseñar? Algunas creencias a nivel escolar indican que en las primeras horas de la mañana son las más adecuadas para ubicar las asignaturas más complejas, pues suponen que los estudiantes tienen una mayor disposición y atención. No obstante, como señala Fernández (2000), las investigaciones sobre atención y fatiga sugieren que los estudiantes tienen en la mañana un nivel bajo de atención, el cual sube y se mantiene hasta antes del mediodía, momento en el que vuelve a descender. En la tarde el fenómeno se repite, sólo que el proceso de ascenso se ve reflejado sobre las cuatro de la tarde, lo cual implica que en esta parte de la jornada, el nivel más alto de atención se logra de manera más tardía. Es preciso señalar, que aunque existe un debate alrededor de la implementación de una jornada escolar continua o partida (Feito, 2007) y sobre la duración de la jornada (Banco Mundial, 2009), el presente trabajo se orienta en la programación de horarios de forma continua, ya sea en la jornada de la mañana o en la tarde, bajo los parámetros establecidos por el Ministerio de educación de Colombia para las instituciones de carácter público (MEN, 2012).

En concordancia con lo mencionado, Hederich et al. (2004), presentaron el trabajo titulado “Ritmos cognitivos en la escuela”, una investigación que expone las características cognitivas de los estudiantes y su relación directa con la planificación de la jornada escolar, dentro del modelo educativo Colombiano. De sus resultados se destaca: a) La primera hora de la jornada (mañana – tarde) presenta los niveles más bajos de activación fisiológica y de atención. b) Los momentos ideales para el trabajo, intelectualmente exigente, pertenecen al intermedio de la jornada, en los cuales los niveles de activación fisiológica son elevados con un nivel de atención ideal. c) Las últimas horas de la jornada tampoco favorecen el trabajo en asignaturas con un alto componente de atención o que requieran de una actividad intelectual sostenida, por lo que se recomienda ubicar asignaturas de un alto componente motor. d) Dada la dificultad existente con la cantidad de horas de clase semanales y la necesidad de jornadas continuas como parte de la optimización de la infraestructura, se propone que las asignaturas con mayor demanda cognitiva como matemática y lenguajes se ubiquen preferiblemente al intermedio de la jornada, o en su defecto al inicio de la misma, siempre y cuando se privilegien los momentos de presentación de datos durante las primeras horas y los momentos de interpretación y análisis en las segundas horas.

Con base en lo anterior, se propone un modelo de programación de horarios acogiendo no sólo las restricciones generales sobre docentes, asignaturas y salones, sino incluyendo también un acercamiento a la práctica pedagógica, al considerar los momentos de la jornada en los que es más propicio el aprendizaje para ciertas áreas del conocimiento. En este caso, se estipula una programación horaria semanal de 30 horas, repartida de lunes a viernes en 6 clases diarias de una hora. La jornada diaria se divide en tres segmentos de dos horas de acuerdo al nivel de atención previsto, determinando un periodo de inicio bajo, un periodo intermedio alto y un tercer periodo de descenso. A cada periodo se asocian un conjunto de asignaturas de acuerdo a las características de sus contenidos, por ejemplo al primer bloque, áreas de ciencias naturales y ciencias sociales; al segundo bloque, áreas de ciencias exactas, lenguaje e idiomas; y al tercer bloque, áreas de deportes, tecnología y manualidades. El objetivo de la programación es ubicar la mayor cantidad posible de asignaturas en el segmento horario que les corresponde, ya que esto permite el desarrollo de actividades propias al nivel cognitivo que el estudiante tiene en los diferentes momentos de la jornada.

La estructura de la metodología presentada en este documento emplea como estrategia de solución el algoritmo NSGA-II, un algoritmo genético multiobjetivo con elitismo (Carranza et al., 2008; Ruizet et al., 2012), el cual ha sido adaptado para la solución de este problema. La ventaja de su uso está en que permite abordar el problema general a partir de la construcción de soluciones individuales para cada grupo, dando cumplimiento a las restricciones de carácter obligatorio y seleccionando aquellas que incumplen en menor proporción las restricciones blandas, de tal manera que se dé prioridad a las soluciones con mejor disposición de acuerdo al enfoque en los ritmos de aprendizaje. Para mostrar la alta eficiencia del algoritmo bajo el enfoque establecido, se realiza la comparación de los resultados de su función de evaluación, con los entregados por un algoritmo genético simple y un algoritmo de búsqueda aleatoria, en la solución de un mismo problema.

METODOLOGÍA

La metodología planteada para el desarrollo de esta investigación se describe de manera general en la figura 1 Y párrafos siguientes de esta sección:

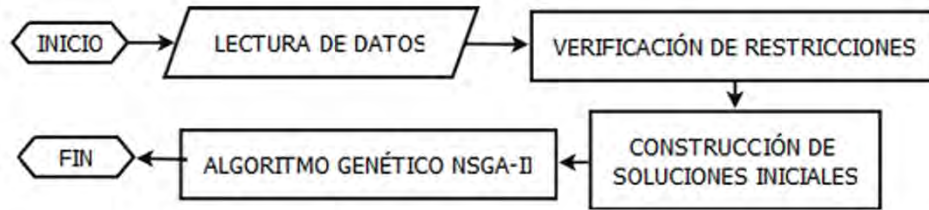


Fig. 1: Esquema general del programa

El proceso requiere del ingreso de la información básica sobre la cual se realiza la construcción del horario. Estos datos son: nombre de las materias y un código único de identificación sin importar el grupo, nombre del profesor que orienta cada materia y su respectivo código único de identificación, intensidad horaria semanal de cada materia, nombre del grupo en el que se orientará dicha materia, código de área a la que pertenece la materia (este código es un identificador del área del conocimiento en el que se ubica una asignatura, por ejemplo, materias como matemáticas, cálculo, geometría pertenecen al área de ciencias exactas), y por último, un número de identificación para el salón donde se orientará la asignatura. Se debe verificar que la intensidad horaria semanal para cada grupo sea de 30 horas de clase y que el tiempo total de la asignación académica semanal de cada docente sea de 22 horas efectivas.

Una vez validados los datos se procede a la construcción de soluciones iniciales, las cuales corresponden a cuatro hipermatrices (matriz de tres dimensiones) denominadas: Materias, Profesores, Salones y Códigos de Área. En cada una de ellas se plasma de manera aleatoria la información correspondiente a cada grupo. La figura 2 muestra la disposición de dichas hipermatrices.

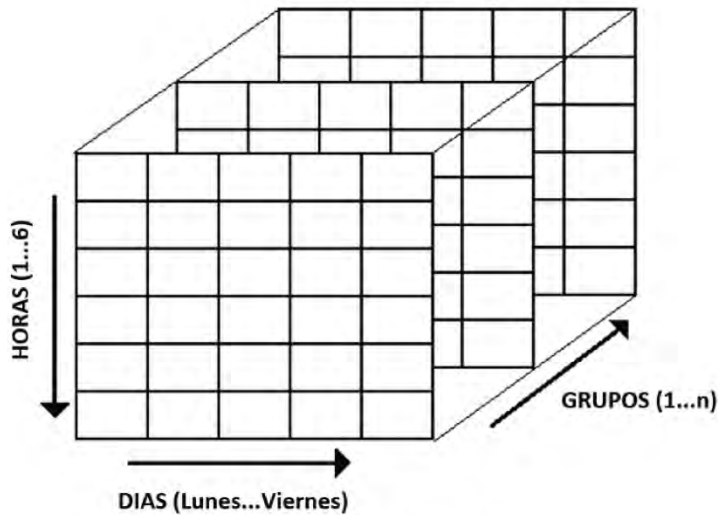


Fig. 2: Esquema de hipermatriz

Las soluciones iniciales constituyen la entrada a la sección que ejecuta el algoritmo NSGA-II, descrito en la figura 3. El algoritmo busca una solución óptima para cada grupo y a partir de estas construye la solución final del problema planteado.

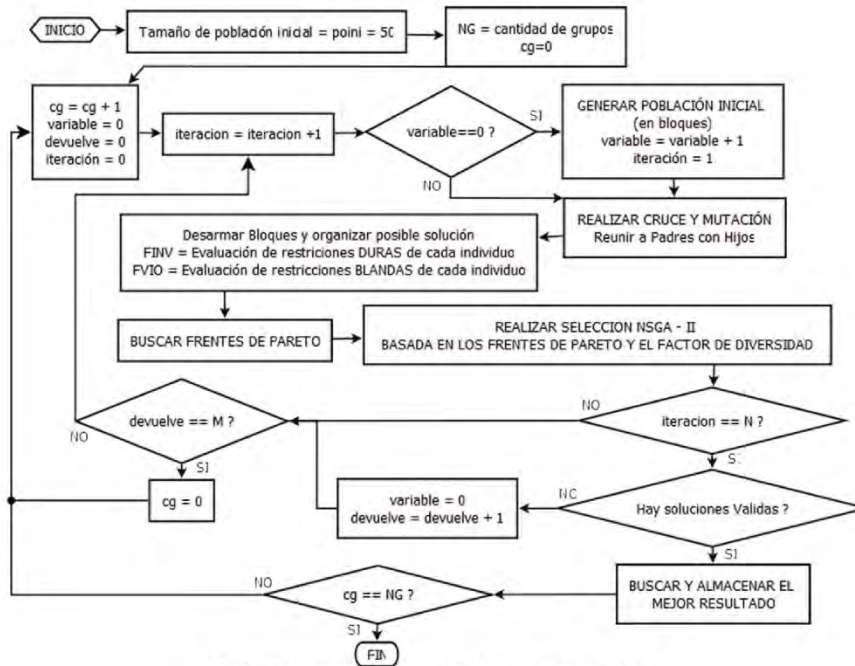


Fig 3: Esquema de ejecución del algoritmo genético

Paso 1: Población Inicial.

Se toman todas las materias de un grupo y se conforman bloques de 2 horas de la misma materia. En caso de que la intensidad horaria de una materia sea impar, además de los bloques correspondientes quedara una hora suelta. Posteriormente se establece el parámetro L, éste representa la cantidad de bloques, en el cual las horas sueltas se incluyen como un bloque más. Además, es necesario determinar el tamaño de la población inicial, la cual es igual $2 \cdot L$. Esto último permitirá generar soluciones aleatorias a partir de una solución inicial, hasta alcanzar el tamaño de la población inicial.

Paso 2: Operadores Genéticos.

Los operadores genéticos de cruce y mutación son aplicados en cada individuo de la población inicial, con el fin de encontrar los nuevos individuos (Hijos). Para el cruce se aplica la técnica llamada Cruce PMX (Partially Matched Crossover) propuesta por Goldberg y Lingle (1985), el cual se realiza sobre las posiciones de los elementos en los individuos y no sobre sus valores. Para ello, se determinan dos posiciones dentro del vector que contiene el individuo; los elementos que se encuentran por fuera del segmento determinado por las posiciones se dejan igual en los hijos, luego los elementos que faltan en el hijo 1 se colocan del padre 2 en el orden secuencial de aparición dentro del individuo, y de igual forma los elementos que faltan en el hijo 2 se toman del padre 1 en el orden secuencial en el que aparecen, tal como se puede observar en la tabla 1. En este ejemplo se tienen dos padres y se determinaron las posiciones 8 y 15 dentro del vector.

Tabla 1: Ejemplo Cruce PMX

PADRE 1	10	9	2	13	11	18	7	12	14	15	4	6	1	8	16	5	17	3
PADRE 2	8	12	14	9	15	3	13	11	1	2	4	6	7	16	10	18	5	17
HIJO 1	10	9	2	13	11	18	7								5	17	3	
HIJO 2	8	12	14	9	15	3	13									18	5	17
HIJO 1	10	9	2	13	11	18	7	8	12	14	15	1	4	6	16	5	17	3
HIJO 2	8	12	14	9	15	3	13	10	2	11	7	4	6	1	16	18	5	17

La mutación se aplica a cada individuo de la población inicial determinando dos posiciones aleatorias en sus elementos. Aquellos que se encuentran por fuera de las posiciones se pasan al hijo como están en el padre y el sub-vector contenido dentro de las dos posiciones se invierte y se pasa al nuevo hijo.

Paso 3: Selección.

Se reúne la población de padres y la población de hijos, siendo necesario desarmar los bloques con el fin de formar cada una de las posibles soluciones. Estas soluciones son evaluadas mediante dos funciones objetivo, las cuales describen las restricciones del problema y se deben cumplir en su totalidad para que la solución generada sea válida. Funciones duras (*RD*) y blandas (*FITNESS*), las cuales están definidas de acuerdo a las condiciones específicas del planteamiento realizado en esta investigación y con base en algunos trabajos previos (Suarez y Castrillón, 2011). La función *RD* contempla las siguientes restricciones: a) En un mismo día no se deben dictar más de 2 horas de la misma materia. b) Si en un día se dictan 2 horas de la misma materia estas deben ser consecutivas. c) Un profesor no puede dictar 2 materias a la misma hora del día. d) A una misma hora no pueden estar programadas dos clases en un mismo salón.

Las siguientes ecuaciones (1-5) representan las funciones de evaluación de estas restricciones, sobre las cuales operan las hipermatrices de Materias, Profesores y Salones:

$$RD = F1 + F2 \quad (1)$$

Los valores de *F1* y *F2* se describen de la siguiente manera.

$$F1 = \sum_{i=1}^{NG} \sum_{j=1}^5 F_{Rest1}(Materias) \quad (2)$$

$$F2 = \sum_{j=1}^5 \sum_{k=1}^6 F_{Rest2}(Profesores) + F_{Rest2}(Salones) \quad (3)$$

Donde:

i = variable para recorrer la cantidad total de grupos a programar.

j = variable para recorrer los días de la semana.

k = variable para recorrer las horas del día.

NG = cantidad total de grupos a programar

La evaluación F_{Rest1} y F_{Rest2} se realizan de la siguiente manera:

$$F_{Rest1} = \begin{cases} 1 & \text{Si algún código se repite más de 2 veces en la columna } j \\ 1 & \text{Si al repetirse algún código 2 veces en la columna } j, \text{ estos no están consecutivos} \\ 0 & \text{En otro caso} \end{cases} \quad (4)$$

$$F_{Rest2} = \begin{cases} 1 & \text{Si se repite algún elemento en el vector de profundidad determinado por } (j, k) \\ 0 & \text{En otro caso} \end{cases} \quad (5)$$

Es importante tener en cuenta que la solución no es válida en caso de que la función objetivo *RD* sea diferente de cero.

La función objetivo *FITNESS* hace referencia al enfoque en los ritmos de aprendizaje y su evaluación determina la calidad de las soluciones. El algoritmo busca que este valor se minimice. La evaluación de esta función se realiza con base a la segmentación de la jornada, con tres tipos de restricciones según la ubicación de cada asignatura durante el día:

Restricción 1: Asignaturas donde los estudiantes tienen una mejor disposición para recibir su contenido en la primera y segunda hora de la jornada.

Restricción 2: Asignaturas donde los estudiantes tienen una mejor disposición para recibir su contenido en la tercera y cuarta hora de la jornada.

Restricción 3: Asignaturas donde los estudiantes tienen una mejor disposición para recibir su contenido en la quinta y sexta hora de la jornada.

Estas restricciones se evalúan en la hipermatriz de Códigos de Área, a través de la función de evaluación presentada en la ecuación 6:

$$FITNESS = \min \left(\sum_{i=1}^{NG} \sum_{j=1}^5 \sum_{k=1}^6 F_{Rest3}(\text{Códigos de Área}) \right) \quad (6)$$

Donde F_{Rest3} evalúa una restricción por iteración, dependiendo del código de área. En la tabla 2, se pueden observar los valores de penalización que se tienen para cada restricción de acuerdo a la ubicación de cada asignatura según la hora del día.

Tabla 2: Valores de penalización restricciones blandas

Hora de la jornada	Valor Penalización Restricción 1	Valor Penalización Restricción 2	Valor Penalización Restricción 3
1	0	6	3
2	0	3	3
3	3	0	6
4	3	0	6
5	6	3	0
6	6	6	0

Una vez evaluadas todas las restricciones descritas, se procede a realizar la selección de los individuos para la nueva población. Dicha selección está basada en dos criterios: el rango de dominancia según los frentes óptimos de Pareto y el factor de diversidad de cada frente. Estos se calculan utilizando los resultados de las funciones objetivo, con el fin de encontrar grupos de soluciones en donde el primero es dominante (mejor calidad) sobre los demás.

La obtención de los frentes de Pareto consiste en organizar los individuos de menor a mayor valor con respecto a la función objetivo RD , aplicar una división recursiva de la nueva población ordenada hasta que se puedan comparar sub-poblaciones de tamaño 1, en donde la sub-población derecha es dominada por la sub-población izquierda respecto a la función objetivo RD . El frente se conforma por la sub-población izquierda más los individuos de la sub-población derecha que no son dominados por ningún individuo de la sub-población izquierda con respecto a la función objetivo $FITNESS$ (Peñuela y Granada, 2007).

Luego de encontrar un frente, sus individuos se descartan de la población ordenada y se repite el proceso hasta ubicar todos los individuos en frentes. La nueva población de individuos se construye agregando los individuos de cada frente hasta completar el tamaño de la población inicial. Sin embargo, puede ocurrir que al agregar un frente a la solución, la cantidad de individuos supere el tamaño de la población inicial. En este caso, si antes del último frente adicionado faltan 1 o 2 individuos, se adicionan el primer individuo o el primero y el último individuos del frente siguiente respectivamente. En caso de que falten más de 2 individuos, se agrega el primero y el último individuo más los escogidos mediante el factor de diversidad, hasta que el tamaño sea igual a la población inicial, teniendo prioridad aquellos individuos cuyo factor de diversidad sea más alto. Este factor permite medir la distancia entre los individuos del mismo frente y logra una mejor exploración del espacio de soluciones.

El proceso descrito desde el cruce y la mutación se repite hasta completar N iteraciones, después se verifica si en la última población seleccionada hay soluciones válidas, es decir, si el vector de la función objetivo RD tiene al menos un cero. En caso de que no se encuentre una solución válida se reinicia el algoritmo generando una nueva población inicial para el grupo actual. Si se debe generar más de M veces la población inicial de un grupo, el algoritmo vuelve a iniciar el proceso desde el grupo 1. Sino, se escoge la solución con el mejor valor de evaluación de la función $FITNESS$ y se procede con el siguiente grupo.

Paso 4: Condición de parada.

El proceso debe continuar hasta completar la totalidad de grupos, en este momento el algoritmo entrega la solución final escogida.

EXPERIMENTACION

Paso 1: Pruebas.

Con el propósito de mostrar la eficiencia de la metodología propuesta, se establece la comparación de los resultados de su función de evaluación, con los obtenidos de la ejecución de un algoritmo genético simple y un algoritmo de búsqueda aleatoria, cuyo diseño y estructura también atienden a las condiciones específicas del planteamiento realizado en esta investigación. El algoritmo genético simple se diferencia del propuesto en la complejidad de sus operadores genéticos (no se manejan frentes) y el algoritmo de

búsqueda aleatoria realiza la programación mediante una asignación recursiva en la que sólo examina el cumplimiento de las restricciones duras.

Se plantean 6 problemas con diferente número de profesores, asignaturas, salones y total de horas programadas (Tabla 3), los cuales son resueltos por cada algoritmo durante 10 repeticiones.

Tabla 3: Parámetros de los problemas de prueba

Problema	Cantidad de grupos	Cantidad de docentes	Cantidad de asignaturas	Cantidad de salones	Total de horas programadas
1	4	14	40	5	120
2	5	19	50	6	150
3	6	20	60	7	180
4	7	23	73	8	210
5	8	23	86	9	240
6	9	23	99	10	270

En cada una de las repeticiones, el resultado obtenido respecto a la función *FITNESS* es tomado como referencia. Se obtienen los siguientes valores (Tabla 4):

Tabla 4: Resultados función *FITNESS* para comparación de eficiencia de algoritmos

Problema	Método	Valor Fitness total por repetición									
		1	2	3	4	5	6	7	8	9	10
1	Aleatorio	339	336	354	357	294	324	372	285	333	321
2	Aleatorio	477	429	444	492	444	432	426	408	387	423
3	Aleatorio	543	483	486	555	492	462	525	483	513	513
4	Aleatorio	630	573	600	507	567	573	501	543	645	627
5	Aleatorio	699	702	651	696	591	690	714	672	693	675
6	Aleatorio	867	702	771	771	768	762	753	765	789	744
1	AG –Simple	228	273	279	261	252	240	261	276	267	258
2	AG –Simple	327	303	321	279	309	315	324	294	288	303
3	AG –Simple	387	411	330	387	447	423	375	390	384	411
4	AG –Simple	501	474	426	459	408	471	507	444	456	444
5	AG –Simple	537	534	528	552	585	546	573	549	516	564
6	AG –Simple	606	627	642	648	600	627	645	609	648	600
1	NSGA-II	201	177	150	165	165	210	168	189	183	204
2	NSGA-II	183	186	198	168	201	255	222	219	189	189
3	NSGA-II	318	300	258	267	276	285	264	285	276	264
4	NSGA-II	279	321	324	303	363	342	399	357	315	300
5	NSGA-II	312	327	423	426	390	384	396	423	411	348
6	NSGA-II	483	450	456	534	471	432	438	480	534	468

Se puede observar que en cada uno de los problemas la mejor solución hallada fue encontrada con el algoritmo NSGA-II, mientras que la peor solución está asociada al método de búsqueda aleatoria. Estas soluciones fueron comparadas mediante un análisis Anova (Tabla 5), para determinar si existen diferencias estadísticamente significativas entre los resultados entregados por cada algoritmo. Posteriormente, se aplicaron pruebas post hoc para establecer entre qué métodos existen dichas diferencias (Tabla 6).

Tabla 5: Análisis de varianza entre métodos de solución

Problema		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Inter-grupos	113016,600	2	56508,300	122,720	,000
	Intra-grupos	12432,600	27	460,467		
	Total	125449,200	29			
2	Inter-grupos	277603,800	2	138801,900	228,882	,000

Problema		Suma de cuadrados	gl	Media cuadrática	F	Sig.
	Intra-grupos	16373,700	27	606,433		
	Total	293977,500	29			
3	Inter-grupos	255861,600	2	127930,800	174,459	,000
	Intra-grupos	19799,100	27	733,300		
	Total	275660,700	29			
4	Inter-grupos	303523,800	2	151761,900	96,859	,000
	Intra-grupos	42304,500	27	1566,833		
	Total	345828,300	29			
5	Inter-grupos	435046,200	2	217523,100	190,670	,000
	Intra-grupos	30802,500	27	1140,833		
	Total	465848,700	29			
6	Inter-grupos	434018,400	2	217009,200	192,817	,000
	Intra-grupos	30387,600	27	1125,467		
	Total	464406,000	29			

Tabla 6: Prueba Post hoc. Test de Tukey

Problema	(I) Tipo de Algoritmo	(J) Tipo de Algoritmo	Diferencia de medias (I-J)	Error típico	Sig.	Intervalo de confianza al 95%	
						Límite inferior	Límite superior
1	Aleatorio	AG-Simple	72,000 [*]	9,597	,000	48,21	95,79
		NSGA-II	150,300 [*]	9,597	,000	126,51	174,09
	AG-Simple	NSGA-II	78,300 [*]	9,597	,000	54,51	102,09
2	Aleatorio	AG-Simple	129,900 [*]	11,013	,000	102,59	157,21
		NSGA-II	235,200 [*]	11,013	,000	207,89	262,51
	AG-Simple	NSGA-II	105,300 [*]	11,013	,000	77,99	132,61
3	Aleatorio	AG-Simple	111,000 [*]	12,110	,000	80,97	141,03
		NSGA-II	226,200 [*]	12,110	,000	196,17	256,23
	AG-Simple	NSGA-II	115,200 [*]	12,110	,000	85,17	145,23
4	Aleatorio	AG-Simple	117,600 [*]	17,702	,000	73,71	161,49
		NSGA-II	246,300 [*]	17,702	,000	202,41	290,19
	AG-Simple	NSGA-II	128,700 [*]	17,702	,000	84,81	172,59
5	Aleatorio	AG-Simple	129,900 [*]	15,105	,000	92,45	167,35
		NSGA-II	294,300 [*]	15,105	,000	256,85	331,75
	AG-Simple	NSGA-II	164,400 [*]	15,105	,000	126,95	201,85
6	Aleatorio	AG-Simple	144,000 [*]	15,003	,000	106,80	181,20
		NSGA-II	294,600 [*]	15,003	,000	257,40	331,80
	AG-Simple	NSGA-II	150,600 [*]	15,003	,000	113,40	187,80

El análisis de varianza muestra que la significancia de la prueba es inferior al 5% en los 6 problemas, por tanto, existen diferencias estadísticas entre los resultados obtenidos por los diferentes algoritmos. El test de Tukey determinó que dicha diferencia se da entre todos los métodos (sig=0,000). En este sentido, el valor medio de la función de evaluación del método NSGA-II en todos los problemas, presenta una eficiencia 29,31% y 44,74% mayor con respecto al algoritmo genético simple y al algoritmo de búsqueda aleatoria respectivamente. De esta manera, se establece que el algoritmo NSGA-II presenta los mejores resultados.

Paso 2: Prueba piloto.

En esta sección se realizó una prueba piloto para verificar el efecto de la programación de horario dentro del proceso académico de la institución educativa Colegio San Jorge, en la ciudad de Manizales, institución de

carácter público, que atiende aproximadamente 640 estudiantes repartidos en 17 grupos de sexto a once. Del grado sexto a décimo existen tres grupos por nivel, mientras que en undécimo sólo son dos grupos. Para lograr una comparación objetiva, se tomaron dos grupos por nivel, realizando la programación de estos con los algoritmos NSGA-II y Búsqueda Aleatoria. El propósito es verificar la mortalidad académica de todas las asignaturas (Cantidad de asignaturas perdidas) en cada uno de los grupos y determinar si existe alguna relación con el método de programación horaria utilizada. El problema aplicado a los dos algoritmos tiene los siguientes parámetros: 6 grupos, 23 profesores, 8 salones y 66 asignaturas. Se da la condición de uso de dos salones adicionales a la cantidad de grupos, los cuales corresponden a la sala de cómputo y al patio de deportes, los cuales se deben asignar a cada grupo sin que haya dos o más grupos asignados a estos salones al mismo tiempo.

RESULTADOS Y DISCUSIONES

Con base en la metodología propuesta se solucionó el problema planteado en el paso 2 de la experimentación. La programación de horarios generada bajo los dos métodos, fue implementada en el colegio mencionado. Una vez aplicada la jornada propuesta bajo cada método (NSGA-II y Búsqueda Aleatoria), se recopiló la información de un periodo lectivo respecto al rendimiento académico en los grupos evaluados la cual se encuentra resumida en la tabla 7. Allí se especifica la cantidad de alumnos por grupo y la cantidad de asignaturas orientadas en cada nivel. La cantidad de asignaturas perdidas corresponde a la sumatoria de las asignaturas perdidas por todos los estudiantes de un grupo durante el periodo evaluado. El valor del porcentaje de pérdida intragrupo representa la proporción de estas asignaturas respecto a la cantidad total de asignaturas orientadas en un grupo.

Tabla 7. Resultados experimentación

Nivel y grupo	Método y valor Fitness	Cantidad alumnos	Cantidad asignaturas orientadas	Total asignaturas orientadas	Cantidad asignaturas perdidas	% Perdida intragrupo	% Diferencia entre grupos	
6	1 aleatorio	108	49	10	490	89	18,2%	1,9%
	2 NSGA-II	24	48	10	480	78	16,3%	
7	1 aleatorio	96	42	10	420	85	20,2%	8,5%
	2 NSGA-II	54	41	10	410	48	11,7%	
8	1 aleatorio	84	41	10	410	68	16,6%	4,4%
	2 NSGA-II	51	42	10	420	51	12,1%	
9	1 aleatorio	84	41	10	410	106	25,9%	6,1%
	2 NSGA-II	60	42	10	420	83	19,8%	
10	1 aleatorio	99	42	13	546	129	23,6%	0,7%
	2 NSGA-II	48	39	13	507	116	22,9%	
11	1 aleatorio	72	38	13	494	77	15,6%	2,3%
	2 NSGA-II	54	37	13	481	64	13,3%	
Promedio						18,0%	4,0%	

Los resultados muestran que los valores de la función *FITNESS* son menores en todos los grupos programados mediante el algoritmo NSGA-II. Adicional a esto, el valor promedio de asignaturas perdidas por grupo es del 18% y la diferencia entre grupos muestra un promedio de 4 puntos porcentuales por encima en los grupos programados con el algoritmo de búsqueda aleatoria. Esto indica que los grupos programados con la metodología propuesta basada, en el método NSGA-II, presentan una reducción del 22,22 % en la cantidad de materias perdidas por grupo.

Finalmente, se resalta que en la revisión literaria no se encuentra una metodología para la programación de horarios escolares que tenga como objetivo reducir la mortalidad académica o que contemple un conjunto de restricciones asociadas a los ritmos cognitivos de los estudiantes o alguna otra variable relacionada con los procesos de aprendizaje. Por lo general, las investigaciones existentes centran el problema en la infraestructura y la carga académica docente, desvinculando el proceso de programación horaria del proceso pedagógico. Por tanto, el modelo propuesto además de permitir la búsqueda de resultados con una alta eficiencia a través del algoritmo NSGA-II, aborda restricciones para obtener un mayor beneficio de las horas de día en las que los estudiantes son más productivos.

CONCLUSIONES

Los resultados permiten obtener las siguientes conclusiones: 1) El enfoque en ritmos cognitivos busca que el problema de programación de horarios escolares se vincule a los procesos pedagógicos relacionados con las necesidades fisiológicas de los estudiantes, en los que determinados momentos de la jornada subrayan un mayor o menor nivel de atención para el desarrollo de cierto tipo de actividades. En este sentido, el uso del algoritmo NSGA-II para la resolución de un problema de esta complejidad, permite obtener un conjunto de soluciones que acogen las múltiples restricciones impuestas y que en comparación con otros algoritmos utilizados (Algoritmo genético simple, Algoritmo de búsqueda aleatoria), alcanza un grado de optimización de hasta un 44,74% superior, en la ubicación de las asignaturas en la jornada, en función de la capacidad intelectual que requieren. 2) La aplicación de la metodología a la programación horaria de una institución educativa, muestra que los grupos, en los que su distribución horaria tiene un enfoque basado en los ritmos de aprendizaje, presentaron un descenso en la mortalidad académica hasta de un 22,22%, en relación con a los demás grupos. 3) El algoritmo permite tener en cuenta características como el uso de salones adicionales a los correspondientes para cada grupo, conservando las limitaciones de traslape; además de habilitar la asignación de múltiples materias para un docente dentro de un mismo grupo. 4) La metodología propuesta está diseñada de acuerdo a las características de la escuela pública en Colombia, lo cual permite que pueda ser aplicada en diferentes instituciones educativas cuya jornada sea continua (mañana, tarde), empleando diversas técnicas inteligentes, con el fin de obtener un mejor análisis sobre su verdadero impacto en la disminución de la mortalidad académica.

AGRADECIMIENTOS

Este artículo es parte del proyecto denominado "*Disminución de la mortalidad académica en instituciones de educación básica y media mediante el empleo de Técnicas Inteligentes en la asignación de horarios*" (código Hermes 13744), Financiado por la Universidad Nacional de Colombia.

REFERENCIAS

- Abramson, D, Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*: 37(1) 98-113 (1991).
- Abramson, D., y J. Abela, A Parallel Genetic Algorithm for Solving the School Timetabling Problem. 15 *Australian Computer Science Conference*, 1-9 hobart, australia feb (1992).
- Anaut, D. y otros tres autores, Optimización de redes eléctricas mediante la aplicación de algoritmos genéticos, *Información Tecnológica*, 20 (4), 137-148 (2009).
- Avella, P. y otros tres autores, A computational study of local search algorithms for Italian high-school timetabling. *Journal of Heuristics*, 13(6) 543-556 (2007).
- Banco Mundial, La calidad de la educación en Colombia: Un análisis y algunas opciones para un programa de política. Misión residente en Colombia, Bogota, Colombia (2009).
- Beligiannis, G., C. Moschopoulos. y S. Likothanassis, A genetic algorithm approach to school timetabling. *Journal of the Operational Research Society*, 60(1) 23-42 (2009.)
- Birbas, T., S. Daskalaki. y E. Housos, School timetabling for quality student and teacher schedules. *Journal of Scheduling*, 12(2) 177-197 (2009).
- Čangalović, M. y J. A. Schreuder, Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research*, 51(2) 248-258 (1991).
- Carranza, H. E. y otros tres autores, Un enfoque multiobjetivo para la asignación de canales en sistemas celulares. *Información Tecnológica*: 19(1) 87-96 (2008).
- Colomi, A., M. Dorigo. y V. Maniezzo. Metaheuristics for high school timetabling. *Computational Optimization and Applications*: 9 (3) 275-298 (1998).
- Cornejo R., y J.M. Redondo, Variables y factores asociados al aprendizaje escolar. Una discusión desde la investigación actual. *Estudios Pedagógicos*, 33(2) 155-175 (2007).

- De La Cruz F. M. Las estrategias de aprendizaje: Técnicas para el estudio. *Revista Digita Innovación y Experiencias Educativa*, 1(15) 1-10 (2009).
- Feito A. R, Tiempos escolares: El debate sobre la jornada escolar continua y partida. *Cuadernos de pedagogía*, 1(365) 74-79 (2007).
- Fernández E. M, La hora de la escuela. Análisis y valoración de los procesos, los efectos y las opciones de implantación de la jornada continua. Universidad de Salamanca, Salamanca, España (2000).
- Flórez P, Revisión de algoritmos genéticos aplicados al problema de la programación de cursos universitarios. *Programación Matemática y Software*:3(1) 49-65 (2011).
- Goldberg, D. E. y J. R. Lingle, Loci and the traveling salesman problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*,154-159, (1985).
- Hederich M, C., A. Camargo. y C. Reyes, Ritmos cognitivos en la escuela. Universidad Pedagógica Nacional, Bogota, Colombia (2004).
- Maciques R, Trastornos del aprendizaje. Estilos de aprendizaje y el diagnóstico psicopedagógico. (2004), http://www.sld.cu/galerias/pdf/sitios/rehabilitacion-equino/trastornos_del_aprendizaje_y_estilos_de_aprendizaje_1.pdf. Acceso: 8 de julio de 2012.
- MEN, Decreto 1850. Bogotá: Ministerio de Educación Nacional. (2002). <http://www.mineducacion.gov.co/1621/article-103274.html>. Acceso: 8 de julio de 2012.
- Minh, K. y otros tres autores, Using tabu search for solving a high school timetabling problem. *Studies in Computational Intelligence*: 1(283) 305-313 (2010).
- Murillo, F. J, Enfoque, situación y desafíos de la investigación sobre eficacia escolar en América Latina y el Caribe. En OREALC/UNESCO, y LLECE, Eficacia escolar y factores asociados en América Latina y el Caribe (págs. 17-48). Santiago: Oficina Regional de Educación de la UNESCO para América Latina y el Caribe y el Laboratorio Latinoamericano de Evaluación de la Calidad de la Educación. (2008).
- Pérez, S., Z. Niño. y M. Llobregat. Desarrollo de un programa de simulación de procesos para el tratamiento de emisiones atmosféricas, *Información Tecnológica*, 15 (2), 55-62 (2004).
- Rosa, M. y otros tres autores, Selección mediante simulación de una estructura de modelo cinético para la degradación aeróbica de efluentes de una industria láctea, *Información tecnológica*, 15 (2), 75-78 (2004).
- Ruiz, S., O. D. Castrillón. y W. Sarache, Una metodología multiobjetivo para optimizar un ambiente job shop. *Información Tecnológica*, 35-46.(2012).
- Peñuela M. y M. Granada, Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no-dominado (NSGA-II). *Scientia et Technica*:1(35) 175-180. (2007).
- Suarez V. y Castrillón O, Diseño de una metodología basada en técnicas inteligentes para la distribución de procesos académicos en ambientes de trabajo Job Shop. XV Congreso de Ingeniería de la Organización, 285 – 289, Cartagena, España 7 -9 Septiembre de 2011.
- Tassopoulos, I. X., y G. Beligiannis, Using particle swarm optimization to solve effectively the school timetabling problem. *Soft Computing*: 16(7) 1229-1252 (2012).
- Valouxis, C. y E. Housos, Constraint programming approach for school timetabling. *Computers and Operations Research*:30(10) 1555–1572 (2003).
- Wren, A, Scheduling, Timetabling and Rostering – A Special Relationship. *Practice and theory of automated timetabling*: 1153(1) 46-75 (1996).

ASSIGNMENT OF SCHOOL CLASS SCHEDULES BASED ON STUDENTS LEARNING RHYTHMS USING A GENETIC ALGORITHM

Universidad Nacional de Colombia, Dpto. Ingeniería Industrial, Manizales, Colombia.

GTA Innovación y Desarrollo Tecnológico (Cat A)

{vfsuarezc, odcastrillong, aguerrero}@unal.edu.co

Summary. The objective of this proposal is to implement a school day agenda focused on the learning rhythms of students of elementary and secondary schools using a genetic algorithm. The methodology takes into account legal requirements and constrains on the assignment of teachers and classrooms, for a regular school day, in public educational institutions in Colombia. In addition, this proposal provides a set of constrains focused on cognitive rhythms, scheduling certain school subjects at the most convenient times according to the area of knowledge. The genetic algorithm evolves through a process of mutation and selection and builds a total solution based on the best solutions for each group. A school of 16 groups is tested and the results of class schedule assignments are presented. The quality of the solution obtained through the established approach is validated by comparing the results to the solutions obtained using an additional algorithm.

Keywords: Learning rhythm, Genetic algorithm, School day, constrains.

1. Introduction

School day is the time that primary and secondary schools establish for the direct rendering of educational services. The assignment of class schedules must comply with current legal requirements, with the institutional educational project and also with the curriculum of the school. In general terms, the planning of the school day is exclusively associated with the distribution of classrooms and teachers, but this only takes into account the constrains of the school, the overlapping of class assignments and some special conditions of each school. However, the assignment of class schedules not only involves the management of infrastructure resources and faculty, but also the academic performance of students. Many schools fail to see the importance of academic performance when assigning class schedules, thus the learning rhythms of the students are not taken into account. The mental activity of human beings is subject to cycles of greater or lesser degree of accuracy when performing tasks, which vary in frequency having as a result high, medium and low rhythm frequencies. So, students do not always have the cognitive capacity to assimilate certain knowledge or to carry out certain learning activity at any given time of the day. There has been a lot of research done on the learning rhythms of students, also known as circadian rhythms (periods between 20 and

28 hours), which facilitate the analysis of performance according to factors such as characteristics of the task (complexity, motor component), individual differences (age, sex, chronotype, cognitive style, level of motivation) and socio-cultural factors (urban, rural) [1].

The assigning of school schedules mainly focuses on the distribution of classes, and students learning rhythms are ignored. The school day (morning or afternoon) is a continuous program divided into predetermined time periods for the carrying out of academic activities in a given number of subjects. These subjects are associated with a specific grade level, and are taken by a group of students not by choice, as university courses, but as mandatory courses. Class arrangements take into consideration the conditions of the school facilities and classrooms are permanently assigned to the same group, systematically favoring the rotation of teachers between classrooms according to the distribution of class schedules. Multi-purpose classrooms or classrooms for special purposes can also be considered when assigning class schedules as these may turn into permanent classrooms for any of the groups at any given time. Moreover, the assigning of schedules must ensure that a teacher is not placed in two different groups at the same time, or that the weekly schedule assignment for each teacher does not exceed a preset number of hours of supervised work.

This proposal is not beyond the scope of previous work in which a wide variety of techniques have been applied to solve scheduling problem [2] based on models such as integer programming [3,4] linear programming [5], Backtracking [6,7] multi objectives scheduling [8] evolutionary algorithms [9,10,11] metaheuristics [12], mathematical models based on polynomial reduction [13], taboo search [14] and expert systems [15] among others [16,17,18].

This research establishes a class schedule structure that embraces not only the constraints of infrastructure and general capacity of the school, but also includes an approach to a pedagogical practice that considers the times of the day that are more conducive to the learning of certain subjects. The questions of when to teach, or what is the best time to carry out certain academic activities become an important focal point. There are different approaches specifically addressing three types of lines: the first refers to the implementation of a double shift school day or a split shift [19]; the second refers to the duration of the school day [20]; and the third refers to the best time of the day to carry out specific learning actions [1].

The approach of this proposal deals with the most appropriate time of the day for teaching and the considerations made by Hederich [1] were adapted and adjusted to the general parameters of public educational institutions in Colombia, where the school day is continuous and lasts six hours. Three instances of time in which attention and cognitive activity are more conducive to certain types of subjects were determined. The chosen schedule, which attempts to concentrate the most difficult subjects in the middle of the

day, is presented in the methodology section and is identified as the time window in which students have a greater attention span and a better cognitive disposition.

The computational model presented in this paper is a genetic algorithm, which although it has been used repeatedly to solve the traditional approach to the problem [21,22,23,24], has properties that allow to cover comprehensively the multiple needs presented, including the set of constraints associated with cognitive rhythms, which increase the complexity of the problem.

In summary, the assigning of class schedules will be focused on the learning rhythms of the students. This is done through the use of a genetic algorithm which seeks to solve and optimize a class schedule in which the location of subjects will depend on constraints of the activities planned, so that these match the time of the day when the cognitive ability of the students is more suitable.

2. Methodology

Figure 1 generally describes the methodology developed for the solution of the problem in this research:

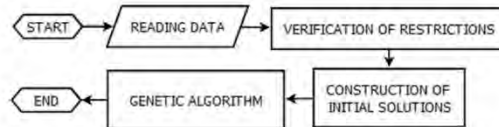


Figure 1. General scheme of the algorithm.

Basic information such as names of the subjects, identification codes for each subject (regardless of group), names of the teachers teaching each subject, identification codes for each teacher, current weekly hours per subject, name of the group taking each subject, area identification codes for the area of each subject (this code is an identifier of the area of knowledge in which a subject is located, for example, subjects like math, calculus, and geometry belong to the area of sciences), and identification numbers for the classrooms where each subject is to be taught, was taken into account to structure class schedules.

Since this is a program focused on the characteristics of educational public institutions, the information provided must adhere to two basic conditions, which are previously verified before giving way to finding a solution [25]: a) Each group must take 30 hours of class a week. B) A total of 22 hours of class will be assigned to a teacher each week

Once the data is validated, initial solutions are structured and these correspond to four hipermatrices ¹ in which the information of each group concerning subjects, teachers, codes of knowledge area, and code of classrooms are randomly added. Figure 2 shows the arrangement in which such hipermatrices are constructed.

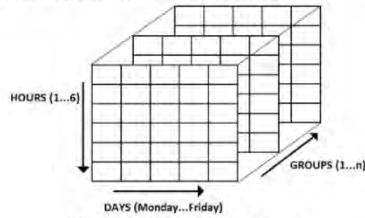


Figure 2. Hypermatrix scheme.

The initial solutions provide the input for the implementation of the genetic algorithm, which is described in Figure 3:

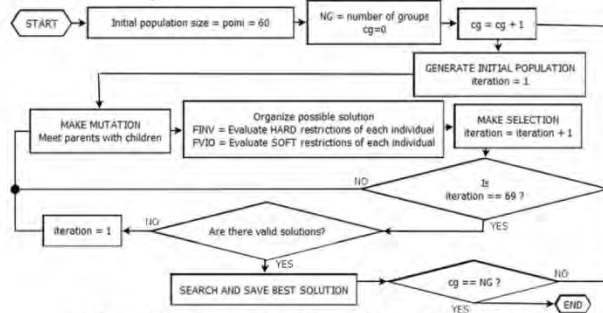


Figure 3. Implementation scheme of the genetic algorithm

The first step of the algorithm is to determine the initial population size (Parents). The dimension must be between L and 2^3L , where L is the size of the input vector, which in this case is 30 positions corresponding to each class hour to be scheduled in each group. The next step is to specify the number of groups (NG) and the counter begins (cg) shifting the group on which the algorithm is applied. This mechanism allows the best solutions for each group and not just an overall optimal solution.

¹ Three-dimensional matrix

The initial solution from each group is taken. Other solutions are randomly generated using the elements of this initial solution, until the size determined in the initial population is reached. The maximum size of the initial population, that is $2 * L$, was considered in order to cover a wider solution space and to allow the attaining of a more optimal solution. The counting of the iterations was then reset to zero, which represents the number of generations to be found before evaluating whether or not there are any valid solutions. For this case, a parameter of 70 iterations was chosen, which was found experimentally by selecting the average value at which the algorithm guarantees at least one valid solution for the group.

The next step is to apply mutation to each individual in the initial population in order to find new individuals (children). Two random positions within the vector of each individual are determined, and the sub-vector contained within those two positions is inverted. This particular genetic algorithm does not perform crossover as the genetic content of the individuals would be altered by using this coding method, resulting in invalid solutions.

Once the parents and children are obtained, they are gathered in one community, to further evaluate the hard and soft constrains and thus verify the validity and level of optimization of the solutions. Hard constrains take into consideration general conditions of the school scheduling problem and are as follows: a) No more than 2 hours on the same subject matter should be taught in one single day. b) A block should be scheduled when two hours on the same subject are to be taught in one day. c) A teacher can not teach 2 subjects at the same time of day. d) Two classes cannot be scheduled at the same time in the same classroom.

These constrains are evaluated in the Hypermatrix of subjects, teachers and classrooms. The solution is invalidated when there is not full compliance with these constrains. Soft constrains refer to learning rhythms and its evaluation determines the quality of the solutions, according to the location of each subject during the time of day. Taking into account the information in Table 1:

Table 1. Description of soft constrains

Constrains	Description
R1	It covers subjects in which students are more willing to receive their content in the first and second hour of the school day. Subjects from the areas of natural sciences and social sciences are examined here.
R2	It covers the subjects in which students are more willing to receive their content in the third and fourth hour of the school day. Subjects from the areas of science and languages are examined here.
R3	It covers subjects in which students are more willing to receive their content in the fifth and sixth hour of the day. Subjects from the areas of sports, technology and crafts are examined here.

These constrains are evaluated in the hypermatrix of knowledge area codes. Using the evaluation function in equation (1):

$$\text{Evaluation function (fitness)} = \min \left(\sum_{i=1}^{NG} \sum_{j=1}^5 \sum_{k=1}^6 F_{Soft_Const}(\text{codes}(i,j,k)) \right) \quad (1)$$

Where:

i = variable to traverse the total number of groups to be programmed.

j = variable to traverse the days of the week.

k = variable to traverse the hours of the day.

$F_{Soft_Const}(\text{codes}(i,j,k))$ = Evaluation function of soft constrains in the hypermatrix of the knowledge area codes. Depending on the knowledge area code, only one constrain per iteration is evaluated.

Table 2 shows the penalized low grade values held by each constrain according to the location of each subject at a certain time of the school day.

Table 2. Soft constrain penalized values

Time of the school day	Penalized value Constrain 1	Penalized value Constrain 2	Penalized value Constrain 1
1	0	6	3
2	0	3	3
3	3	0	6
4	3	0	6
5	6	3	0
6	6	6	0

The selection of new individuals takes place once the constrains described above are evaluated and the results are stored in their respective variables. Therefore, a vector that describes pairs with individuals of the community is generated in order to confront each other in a duel in which one of them is rejected and the other is selected for the next generation, thus: a) The solutions to be confronted are taken according to the vector of pairs. b) The values of verification of the soft constrains are compared and the lowest (preferably zero) is chosen. If both are of the same value, the next step is taken. c) The values of verification of the soft constrains are compared in each solution and the lowest is chosen. If both are of the same value, one of the solutions is randomly chosen, since they are considered genetically equal.

The above process is repeated until 70 iterations are completed, and then it is corroborated whether valid solutions are found in the last selected population. That is to say, the verification vector of the hard constrains has at least one zero (0), if not the count is reset

and another 70 iterations are completed. If the result is positive, the best result is selected and the process continues for the next group. Upon completion of all groups, the final chosen solution is shown.

3. Experimentation

The previous methodology was tested for the assignment of class schedules in a high school² in the city of Manizales. This particular school has 16 groups between sixth and eleventh grade. There are three groups per level in sixth, seventh, eighth and ninth grade, and two groups per level in tenth and eleventh grades. The algorithm was implemented during 4 cycles, and 10 solutions to the problem were found in each cycle.

4. Results

Table 3 presents the results of the evaluation function of the 10 total solutions of the problem in each of the four cycles. The best solution found during the 40 iterations corresponded to a value of 1068 in its evaluation function.

Table 3. Results of the evaluation function in 4 cycles

	Iterations									
	1	2	3	4	5	6	7	8	9	10
Cicle 1	1131	1137	1155	1107	1098	1167	1131	1128	1110	1086
Cicle 2	1113	1068	1125	1134	1143	1128	1113	1128	1137	1164
Cicle 3	1107	1143	1167	1131	1137	1113	1128	1068	1134	1107
Cicle 4	1131	1068	1125	1131	1143	1113	1128	1110	1164	1131

These solutions were compared by ANOVA analysis, in order to establish whether there are significant differences between them (Table 4).

Table 4. Analysis of variance-own methodology

	Sum of squares	gl	Mean square	F	Sig.
Inter-groups	18,900	3	6,300	,010	,999
Intra-groups	23265,000	36	646,250		
Total	23283,900	39			

The variance analysis in Table 4 shows that the significance value is higher than 5%, and consequently it can be said that the results obtained are statistically equal, which proves the stability of the algorithm.

² Educational Institution San Jorge School.

Subsequently, another variance analysis was performed to determine if both methods have significant differences (table 6) using the solutions generated in cycles 1 to 3 of table 3, and also 10 values of the evaluation function of the solutions found by the commercial software (SOS) [26] used for the generation of class schedules in the school under study (Table 5). With the problem (Table 5), we performed another variance analysis to determine if both methods have significant differences (Table 6).

Table 5. Evaluation function results in SOS solutions

Repeticiones										
	1	2	3	4	5	6	7	8	9	10
SS	1314	1257	1284	1323	1218	1350	1314	1305	1284	1290

Table 6. Variance analysis-own methodology Vs. SOS

	Sum of squares	gl	Mean square	F	Sig.
Inter-grupos	214987,275	3	71662,425	86,146	,000
Intra-grupos	29947,500	36	831,875		
Total	244934,775	39			

Table 7. Post hoc analysis. Tukey test

(I) Sample	(J) Sample	Mean difference (I-J)	Estándar Error	Sig.
1	3	1,5	12,899	0,999
	SC	-168,900	12,899	0,000
2	1	0,3	12,899	1
	3	1,8	12,899	0,999
3	SC	-168,600	12,899	0,000
	1	-1,5	12,899	0,999
SS	2	-1,8	12,899	0,999
	SC	-170,400	12,899	0,000
SS	1	168,900	12,899	0,000
	2	168,600	12,899	0,000
	3	170,400	12,899	0,000

Table 6 shows that the significance of the test is less than 5%, therefore there are statistical differences between the means of the results obtained. The Tukey test was applied to confirm that the difference are between the samples of the own model and the SOS (Table 7). It was found in the contrast tests that the results obtained of the evaluation function in the execution cycles always showed significant differences with the SOS.

5. Conclusions

The results show that the methodology developed achieves an improvement of 12.3% in the assignment of class schedules based on the model of student learning rhythms when compared to the SOS, program currently used by the school under study. Additionally, the algorithm shows high stability in the found solutions, ensuring an efficiency of over 92%, in the results obtained in each run. Finally, this methodology is to be implemented as a future line of research in a large number of schools in order to analyze its impact.

6. Acknowledgements

This article is part of the project entitled "Disminución de la mortalidad académica en instituciones de educación básica y media mediante el empleo de Técnicas Inteligentes en la asignación de horarios ", funded by the Universidad Nacional de Colombia, Manizales.

References

- [1]. Hederich Martinez, Christian, Camargo Uribe, Ángela y Reyes Cuervo, María Emma. Ritmos cognitivos en la escuela. Bogotá, Colombia : Universidad Pedagógica Nacional, 2004.
- [2]. An introduction to timetabling. De Werra, D. 1985, European Journal of Operational Research, págs. 151-162.
- [3]. An integer programming formulation for a case study in university timetabling. Daskalaki, S, Birbas, T y Housos, E. 2004, European Journal of Operational Research, págs. 117-135.
- [4]. Efficient solutions for a university timetabling problem through integer programming. Daskalaki, S y Birbas, T. 2005, European Journal of Operational Research, págs. 106-120.
- [5]. New integer linear programming approaches for course timetabling. Boland, Natashia, y otros. 2008, Computers & Operations Research, págs. 2209-2233.
- [6]. Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. Patterson, James H, y otros. 1990, European Journal of Operational Research, págs. 68-79.
- [7]. Backtracking techniques for the job shop scheduling constraint satisfaction problem. Sadeh, Norman, Sycara, Katia y Xiong, Yalin. 1995, Artificial Intelligence, págs. 455-480.
- [8]. Recent research directions in automated timetabling. Kieran Burke, Edmund y Petrovic, Sanja. 2002, European Journal of Operational Research, págs. 266-280.
- [9]. Programación óptima de horarios de clase usando un algoritmo memético. Granada E., Mauricio, Toro Ocampo, Eliana M y Franco Baquero, John F. 2006, Scientia et Technica, págs. 255-260.

- [10]. An informed genetic algorithm for the examination timetabling problem. Pillay, N y Banzhaf, W. 2010, *Applied Soft Computing*, págs. 457-467.
- [11]. Asignación de horarios de clases universitarias mediante algoritmos evolutivos. Mejía Caballero, José María y Paternina Arboleda, Carlos. 2010, *Educación en Ingeniería*, págs. 140-149.
- [12]. A decomposed metaheuristic approach for a real-world university timetabling problem. De Causmaecker, Patrick, Demeester, Peter y Vanden Berghe, Greet. 2009, *European Journal of Operational Research*, págs. 307-318.
- [13]. Polynomial reduction of time space scheduling to time scheduling. Studenovský, J. 2009, *Discrete Applied Mathematics*, págs. 1364-1378.
- [14]. Adaptive Tabu Search for course timetabling. Lü, Zhipeng y Hao, Jin-Kao. 2010, *European Journal of Operational Research*, págs. 235-244.
- [15]. Parallelizing a CLIPS-based course timetabling expert system. Wu, Chao-Chin. 2011, *Expert Systems with Applications*, págs. 7517-7525.
- [16]. An integrated hybrid approach to the examination timetabling problem. Turabieh, Hamza y Abdullah, Salwani. 2011, *Omega*, págs. 598-607.
- [17]. Solving timetabling problems using a cultural algorithm. Soza, Carlos, y otros. 2011, *Applied Soft Computing*, págs. 337-344.
- [18]. Lower bounds for the ITC-2007 curriculum-based course timetabling problem. Hao, Jin-Kao y Benlic, Una. 2011, *European Journal of Operational Research*, págs. 464-472.
- [19]. Tiempos escolares: El debate sobre la jornada escolar continua y partida. Feito Alonso, Rafael. 2007, *Cuadernos de pedagogía*, págs. 74-79.
- [20]. Banco Mundial. La calidad de la educación en Colombia: Un análisis y algunas opciones para un programa de política. Bogotá: Misión residente en Colombia, 2009.
- [21]. A Genetic Algorithm to Solve the Timetabling Problem. Colomi, Alberto, Dorigo, Marco y Maniezzo, Vittorio. 1990, *Computational Optimization and Applications Journal*.
- [22]. A Genetic Algorithm Solving a Weekly Course-Timetabling Problem. Erben, Wilhelm y Keppler, Jürgen. 1996, *Lecture Notes in Computer Science*, 1996, Volume 1153, *Practice and Theory of Automated Timetabling*, págs. 192-211.
- [23]. Algoritmo Evolutivo para generar cargas académicas en TIC-SI. Mendoza Crisostomo, Patricia, Flores Trujillo, Ivan Antonio y Morales Genis, Delfino. Xalapa, México: s.n., 2009. 4to Simposio Internacional en Sistemas Inteligentes y Organizaciones Inteligentes.
- [24]. A genetic algorithm approach to school timetabling. Beligiannis, GN, Moschopoulos, C y Likothanassis, SD. 2009, *Journal of the Operational Research Society*, págs. 23-42.
- [25]. MEN. Decreto 1850. Bogotá: Ministerio de Educación Nacional, 2002.
- [26]. Software, Peñalara. GHC. Generador de horarios para centros de enseñanza. 2008.

*5th International Conference on Industrial Engineering and Industrial Management
XV Congreso de Ingeniería de Organización
Cartagena, 7 a 9 de Septiembre de 2011*

Diseño de una metodología basada en técnicas inteligentes para la distribución de procesos Académicos en ambientes de trabajo job shop.

Víctor Fabio Suarez, Omar Danilo Castrillón¹,

¹ Dpto. de Ingeniería Industrial, Universidad Nacional de Colombia.
vfsuarez@unal.edu.co, odcastrillong@unal.edu.co

Palabras Claves: Técnicas Inteligentes, Job Shop, ritmos cognitivos.

1. Introducción

La programación de horarios dentro de las instituciones educativas ha sido examinada desde diversas perspectivas, principalmente a nivel universitario. Sin embargo, se ha dejado de lado el problema de asignación de horarios en las instituciones de educación básica y media, en las cuales es necesario el establecimiento de múltiples parámetros que permitan una asignación que facilite la organización de restricciones especiales que favorezcan los momentos del aprendizaje de acuerdo a las necesidades de cada una de las áreas del conocimiento, sin afectar una equitativa repartición en la carga de los docentes y sin sobrepasar los límites de horas diarias y semanales que son necesarias dentro de cada materia.

A diferencia de la asignación de horarios universitarios, los horarios escolares tienen un enfoque orientado a las clases y no a los estudiantes, por lo cual la carga académica se programa de una manera intensiva en periodos de una jornada diaria, para un grupo de estudiantes que deben cursar unos contenidos estructurados en diferentes áreas y repartidos en intensidades no superiores a las dos horas diarias en la misma asignatura. De esta manera, la planta física también cumple con la orientación a las clases, por lo cual los salones se asignan de forma permanente a un mismo grupo de estudiantes y se propicia de una manera sistemática la rotación de los docentes por cada una de las aulas según la distribución que se haga de la carga en la respectiva programación. En cuanto a la disponibilidad de los maestros, se da por hecho que no hay ningún tipo de conflicto en la asignación horaria durante la jornada, y que deben dictar un número mínimo de horas durante la programación semanal.

De igual forma, es necesario tener en cuenta que hay grupos en los que la jornada académica puede tener una extensión más prolongada y en la que las horas extendidas deben ser realizadas en una asignatura específica. En cuanto a la planta física, aunque cada grupo tiene un aula asignada por defecto para la gran mayoría de las asignaturas, es necesario contemplar el uso de aulas auxiliares tales como las salas de sistemas, aulas o espacios para las clases de deporte, laboratorios, salones de música y salas de multimedia.

Desde la anterior perspectiva, el problema planteado no escapa de los alcances hechos por trabajos anteriores. Sin embargo, las limitaciones surgen al establecer un conjunto de

restricciones que deben ser acogidas con tal de lograr adecuar la programación a estrategias y modelos pedagógicos sobre los cuales se instauran consideraciones respecto a los momentos de la jornada en los que es más propicio el aprendizaje para algunas áreas del conocimiento. En general, las investigaciones a nivel pedagógico se han centrado en responder a preguntas como ¿qué enseñar?, ¿cómo enseñar? o ¿dónde enseñar?, mas sin embargo la pregunta ¿cuándo enseñar? no ha sido abordada de una manera específica y mucho menos, se han desarrollado herramientas que permitan planificar este tiempo de una forma acorde con los ritmos cognitivos (Hederich Martínez, Camargo Uribe, & Reyes Cuervo, 2004). Es así, que buscar que el establecimiento de la jornada académica tenga como condición la variación del rendimiento intelectual, permitirá no solo que el nivel de aprendizaje se incremente, sino que consecuentemente se reduzca la mortalidad académica en las asignaturas de mayor dificultad.

El planteamiento inicial del problema supone la existencia de las aulas suficientes para los grupos que se desean establecer, es decir, la planta física debe contar mínimo con aulas disponibles y con la capacidad suficiente para albergar a los grupos de alumnos, cuya cantidad está determinada por las políticas y carácter de la institución. Las aulas para la orientación de actividades que requieren espacios o materiales de trabajo diferentes, son contadas como adicionales, y en ningún caso se les podrá asignar un grupo de forma permanente durante toda la jornada. De igual forma, se parte del hecho de que los docentes deben permanecer en la institución durante toda la jornada, no obstante su carga académica no debe superar el 85% del total de horas de todas las jornadas de una semana. A la vez, se supone que la cantidad de maestros es suficiente para cumplir con la carga académica necesaria en cada área del conocimiento en la cantidad total de grupos a crear. La asignación de docentes debe contemplar también la posibilidad de asociar más de una asignatura a un solo docente.

En este sentido, el desarrollo de modelos computacionales ofrece ventajas que permiten abarcar con detalle las múltiples necesidades planteadas, para lo cual el uso de metaheurísticas surge como una respuesta a la complejidad que encierra este problema.

El término metaheurística se obtiene de anteponer a heurística el sufijo meta que significa más allá" o a un nivel superior". Las concepciones actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las metaheurísticas son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El término metaheurística apareció por primera vez en el artículo seminal sobre búsqueda tabú de Fred Glover en 1986 (Brito Santana, y otros, 2004). Antes de que el término fuese aceptado completamente por la comunidad científica, estas técnicas eran denominadas *heurísticas modernas* (Reeves, 1993). Esta clase de algoritmos incluye técnicas como algoritmos genéticos (Chao-Hsien Pan & Huang, 2009) (Manikas & Chang, 2009), lógica difusa (Yun, 2002), búsquedas tabú (Buscher & Shen, 2009), colonias de hormigas (Xing, Chen, Wang, Zhao, & Xiong, 2010), sistemas artificiales inmunes (Ge, Sun, Liang, & Qian, 2008), partículas inteligentes (Eberhart & Kennedy, 1995) y células de aprendizaje automático (Jafarpour, Meybodi, & Shiry, 2007). De igual forma existen modelos en los que se han desarrollado sistemas híbridos de estas metaheurísticas (Fang Ming & Qiong, 2009).

De acuerdo a Luna Valero (2008), de las diferentes descripciones de metaheurísticas que se encuentran en la literatura se pueden destacar ciertas propiedades fundamentales que caracterizan a este tipo de métodos:

- Las metaheurísticas son estrategias o plantillas generales que guían el proceso de búsqueda.
- El objetivo es una exploración eficiente del espacio de búsqueda para encontrar soluciones (casi) óptimas.
- Las metaheurísticas son algoritmos no exactos y generalmente son no deterministas.
- Pueden incorporar mecanismos para evitar regiones no prometedoras del espacio de búsqueda.
- El esquema básico de cualquier metaheurística tiene una estructura predefinida.
- Las metaheurísticas pueden hacer uso de conocimiento del problema que se trata de resolver en forma de heurísticos específicos que son controlados por la estrategia de más alto nivel.

Lo anterior se puede sintetizar en precisar a una metaheurística como una estrategia de alto nivel que usa diferentes métodos para explorar el espacio de búsqueda. En otras palabras, una metaheurística es una plantilla general no determinista que debe ser rellenada con datos específicos del problema (representación de las soluciones, operadores para manipularlas, etc.) y que permiten abordar problemas con espacios de búsqueda de gran tamaño.

En la literatura se pueden encontrar gran variedad de técnicas que han sido diseñadas para la programación de horarios (De Werra, 1985), basados en modelos como: Profesor clase, pre asignaciones, heurísticas, reglas de decisión, programación de restricciones (Valouxis & Housos, 2003), programación de horarios multiobjetivos (Kieran Burke & Petrovic, 2002), Algoritmos genéticos (Granada E., Toro Ocampo, & Franco Baquero, 2006) (Pillay & Banzhaf, 2010) (Mejía Caballero & Paternina Arboleda, 2010), modelos matemáticos basados en reducción polinómica (Studenovský, 2009), búsqueda tabú (Lü & Hao, 2010), sistemas expertos (Wu, 2011), entre otras técnicas (Kahar & Kendall, 2010) (Turabieh & Abdullah, 2011) (Soza, Landa Becerra, Riff, & Coello Coello, 2011) (Hao & Benlic, 2011).

Finalmente, se expresa que la presente investigación busca la aplicación de técnicas inteligentes para la solución del problema de asignación de horarios en las instituciones de educación básica y media, basado en un enfoque en ritmos cognitivos, encontrando soluciones con una efectividad superior al 93% respecto a la solución óptima.

2. Metodología

Paso 1: Identificación de restricciones. Inicialmente se identifican todas las restricciones en la programación de horarios, cada restricción es calificada de uno a cinco. Según sea fuerte (5) o débil (1). Las restricciones inviolables, las cuales producen soluciones inválidas, son calificadas con cinco.

Paso 2: Codificación del problema. El problema es codificado, por medio de un vector, como el ilustrado en la Tabla 1.

Tabla 1. Representación de la solución inicial.

-----Grupo 1-----				-----Grupo 2-----				-----Grupo 3-----	
M ₁	M ₃	M _{2...}	M _N	M ₃	M _{2..}	M _N	...	M _{1...}	M _N
(Hora1)	(Hora2)	(Hora3)		(Hora1)	(Hora2)...	(Hora3)			

En la anterior estructura cada conjunto de cuadros sombreados representa un grupo, el contenido de cada celda representa la asignatura y la posición del vector representa la hora de clase. La anterior representación es tomada del problema de asignación de pedidos en centros de trabajo en un ambiente Job Shop. En esta solución se cambian los pedidos por materias y los centros de trabajo por grupos (Castrillón, Giraldo, & Sarache, 2009).

Paso 3. Función Objetivo. Con el fin evaluar cada una de las soluciones generadas, se define la siguiente función:

$$Fitness = \text{Min} \left(\sum_j K_j \right) \begin{cases} \text{Si no viola restriccion : } K = 0, \\ \text{Si viola restriccion menor a } 5, \\ K = 10 * \text{Calificacion_restriccion}. \\ \text{Si restriccion violada vale cinco :} \\ K = \infty \end{cases}$$

1)

Paso 4: Generación y selección de la mejores soluciones. Con base en las definiciones establecidas en los pasos dos y tres y mediante los operadores genéticos de combinación (97%) y mutación (3%), se inicia un proceso de generación de las soluciones, hasta que transcurra un numero de iteraciones $N = 1000$, sin que la función objetivo haya sido mejorada. En este punto el algoritmo, por lo menos, se encuentra en un óptimo local.

Experimentación

Para la experimentación del anterior problema se toma como referencia un problema hipotético en una institución educativa, en un solo grupo. Se suponen restricciones de todas las categorías.

Tabla 2. Asignaturas de clase.

Código	Nombre	Horas /Semana
M0	Descanso	5
M1	Ingles	5
M2	Castellano	5
M3	Matemáticas	5
M4	Sociales	4
M5	C. Naturales	4
M6	Educación Física	3
M7	Religión	2
M8	Aptitudes Deportivas	1
M9	Música	1
M10	Ética	1
M11	Cívica	1
M12	Informática	1
M13	Dibujo	1
M14	Danzas	1

Tabla 3. Restricciones.

Código Materia	Nombre	Restricción cada día	Calificación
-	Horario	7:00 A.M. – 3:00 P.M.	5
Mo	Descanso	9:00-9:30 y 11:30 – 12:00	5
M1	Ingles	8:00 A.M.	3
M2	Castellano	9:30 A.M.	3
M3	Matemáticas	7:00 A.M	3
M4- I4	Otras asignaturas	Después de las 12:00 M.	1

3. Resultados

Pasos 1- 4: Como resultado de estos pasos, se obtiene la siguiente programación de horarios:

Tabla 4. Mejor solución encontrada. Fitness = 50

Código	Lunes	Martes	Miércoles	Jueves	Viernes
7:00	M3	M3	M3	M3	M3
8:00	M1	M1	M1	M1	M1
9:00	M0	M0	M0	M0	M0
9:30	M2	M2	M2	M2	M2
10:30	M5	M5	M6	M6	M6
11:30	M0	M0	M0	M0	M0
12:00	M4	M4	M4	M4	M5
1:00	M5	M7	M7	M9	M10
2:00	M11	M12	M13	M14	M8

La tabla 4 muestra la mejor solución encontrada, es de notar que para una mejor comprensión se pone la solución en forma matricial, sin embargo en la realidad esta solución está representada por un solo vector, en el cual las nueve primeras posiciones representan la columna del lunes en la tabla 4, las posiciones 10 al 18 representan las columna del día martes de la tabla anterior y así sucesivamente.

En la solución planteada en la tabla 4, la franja de las 10:30, no cumple con las restricciones impuestas, aspecto que ocasiona que el menor fitness posibles sea 50.

De otro lado, la peor solución posible (Valida) que se puede obtener, es ilustrada por medio de la tabla 5:

Tabla 5. Peor solución encontrada. Fitness = 650

Código	Lunes	Martes	Miércoles	Jueves	Viernes
7:00	M14	M13	M12	M11	M10
8:00	M6	M6	M7	M8	M9
9:00	M0	M0	M0	M0	M0
9:30	M4	M5	M5	M5	M6
10:30	M7	M5	M4	M4	M4
11:30	M0	M0	M0	M0	M0
12:00	M3	M3	M3	M3	M3
1:00	M2	M2	M2	M2	M2
2:00	M1	M1	M1	M1	M1

Dado que las soluciones generadas por el programa oscilan en un rango comprendido entre 0 (óptimo) y 650 (peor solución válida), se puede establecer el porcentaje de aproximación de la solución encontrada respecto a una solución óptima de cero (si existirá), encontrándose que la mejor solución generada, por lo menos se aproxima en un 93% respecto a una posible solución óptima.

4. Conclusiones

El empleo de las técnicas inteligentes en la programación de horarios, permite una programación más adecuada, la cual respeta las restricciones propuestas y garantiza el desarrollo de los ritmos cognitivos. Una solución que por lo menos se aproxima en un 93% respecto a una solución óptima ideal, si existiera.

En esta metodología la solución solo es penalizada, por no cumplir una restricción. En una futura línea de investigación cada solución puede ser penalizada por medio de una función de lógica difusa, teniendo una mayor penalización cuando más alejada este la asignatura del horario deseado.

Finalmente, como se demuestra en este artículo, es importante resaltar que el empleo de las técnicas inteligentes en los procesos de programación, abarca un gran número de áreas, las cuales tienen una gran utilidad para los docentes, estudiantes, profesionales y organizaciones interesadas en emplear los beneficios de la inteligencia artificial en sus campos de acción.

Bibliografía

- Brito Santana, J., Campos Rodríguez, C., García López, F. C., García Torres, M., Melián Batista, B., Moreno Pérez, J. A., y otros. (2004). *Metaheurísticas: una revisión actualizada*. La Laguna, España: Universidad de La Laguna.
- Buscher, U., & Shen, L. (2009). An integrated tabu search algorithm for the lot streaming problem in job shops. *European Journal of Operational Research*, 385-399.
- Castrillón, O. D., Giraldo, J., & Sarache, W. (2009). Job shop methodology based on an ant colony. *Dyna*, 76 (159), 177-184.
- Chao-Hsien Pan, J., & Huang, H.-C. (2009). A hybrid genetic algorithm for no-wait job shop scheduling problems. *Expert Systems with Applications*, 5800-5806.
- De Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, 151-162.
- Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. *Proceedings of the IEEE International Joint Conference on Neural Networks* (págs. 1942-1948). Perth, Australia: IEEE Press.
- Fang Ming, G., & Qiong, L. (2009). A Hybrid PSO algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Fifth International Conference on Natural Computation* (págs. 171-176). IEEE Press.

- Ge, H.-W., Sun, L., Liang, Y.-C., & Qian, F. (2008). An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling. *Systems and Humans* , 358-368.
- Granada E., M., Toro Ocampo, E. M., & Franco Baquero, J. F. (2006). Programación óptima de horarios de clase usando un algoritmo memético. *Scientia et Technica* , 255-260.
- Hao, J.-K., & Benlic, U. (2011). Lower bounds for the ITC-2007 curriculum-based course timetabling problem. *European Journal of Operational Research* , 464-472.
- Hederich Martínez, C., Camargo Uribe, Á., & Reyes Cuervo, M. E. (2004). *Ritmos cognitivos en la escuela*. Bogotá, Colombia: Universidad Pedagógica Nacional.
- Jafarpour, B., Meybodi, M. R., & Shiry, S. (2007). A hybrid method for optimization (Discrete PSO + CLA). *International Conference on Intelligent and Advanced Systems* (págs. 55-60). IEEE Press.
- Kahar, M., & Kendall, G. (2010). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research* , 557-565.
- Kieran Burke, E., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research* , 266-280.
- Lü, Z., & Hao, J.-K. (2010). Adaptive Tabu Search for course timetabling. *European Journal of Operational Research* , 235-244.
- Luna Valero, F. (2008). *Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones*. Málaga, España: Univesidad de Málaga.
- Manikas, A., & Chang, Y.-L. (2009). Multi-criteria sequence-dependent job shop scheduling using genetic algorithms. *Computer & Industrial Engineering* , 179-185.
- Mejía Caballero, J. M., & Paternina Arboleda, C. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Educación en Ingeniería* , 140-149.
- Pillay, N., & Banzhaf, W. (2010). An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing* , 457-467.
- Reeves, C. R. (1993). *Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publishing .
- Soza, C., Landa Becerra, R., Riff, M. C., & Coello Coello, C. A. (2011). Solving timetabling problems using a cultural algorithm. *Applied Soft Computing* , 337-344.
- Studenovský, J. (2009). Polynomial reduction of time space scheduling to time scheduling. *Discrete Applied Mathematics* , 1364-1378.
- Turabieh, H., & Abdullah, S. (2011). An integrated hybrid approach to the examination timetabling problem. *Omega* , 598-607.

Valouxis, C., & Housos, E. (2003). Constraint programming approach for school timetabling. *Computers & Operations Research* , 1555–1572.

Wu, C.-C. (2011). Parallelizing a CLIPS-based course timetabling expert system. *Expert Systems with Applications* , 7517-7525.

Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S., & Xiong, J. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing* , 888-896.

Yun, Y. S. (2002). Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems. *Computers & Industrial Engineering* , 623-644.

METODOLOGIA BASADA EN PARTÍCULAS INTELIGENTES PARA LA OPTIMIZACIÓN DE LA PRODUCCIÓN EN AMBIENTES JOB SHOP

Victor F. Suarez Chilma

Departamento de Ingeniería Industrial, Universidad Nacional de Colombia
Manizales, Caldas 0017, Colombia. vfsuarezc@unal.edu.co

Santiago Ruiz Herrera

Departamento de Ingeniería Industrial, Universidad Nacional de Colombia
Manizales, Caldas 0017, Colombia. sruizhe@unal.edu.co

Omar D. Castrillón Gómez

Departamento de Ingeniería Industrial, Universidad Nacional de Colombia
Manizales, Caldas 0017, Colombia. odcastrillong@unal.edu.co

RESUMEN

Muchos enfoques diferentes se han generado para la programación de la producción en ambientes Job shop. El presente trabajo tiene como objetivo el planteamiento de una metodología basada en un algoritmo evolutivo denominado *Partículas Inteligentes*, a través de la cual se busca optimizar el proceso de secuenciación de pedidos en este tipo de configuración productiva, estableciendo como propósito la reducción del tiempo total de proceso y del tiempo total de ocio. Para ello se define el problema mediante una representación matricial, la cual permite establecer un espacio D-dimensional en el que las diferentes soluciones evolucionan a través de todo el espacio dado, teniendo como referencia la mejor solución hallada, la cual es renovada progresivamente en la medida que ofrezca niveles superiores de optimización o que se cumpla con parámetros preestablecidos de proceso. Las soluciones halladas mediante esta metodología permiten obtener aproximaciones superiores al 90% respecto a la solución óptima esperada.

Palabras Clave: Partículas inteligentes, Secuenciación de pedidos, Job shop, Optimización, Tiempo de proceso (makespan), Tiempo de ocio (idle time).

1. INTRODUCCIÓN

El problema de programación de la producción en ambientes Job Shop (Job Shop Scheduling Problem - JSSP) ha sido objeto de estudios en contextos académicos e industriales durante las últimas décadas [1][2], lo cual ha conllevado al desarrollo de distintas herramientas matemáticas y computacionales con el fin de alcanzar objetivos como la reducción del tiempo de proceso (makespan), la reducción del tiempo de ocio (idle time) y

el aumento de utilización de las máquinas [3]. En este sentido, las técnicas de inteligencia artificial han cobrado gran fuerza debido a los resultados sobresalientes que ofrecen en el análisis y solución de este tipo de problemas, gracias al empleo de metaheurísticas que permiten la búsqueda de soluciones con niveles de optimización superiores a los métodos convencionales, al mismo tiempo que ofrecen un bajo costo computacional [4].

Si bien, existen diversos tipos de metaheurísticas, las metaheurísticas de búsqueda y las metaheurísticas evolutivas son las que mejor se acomodan a los problemas de optimización que presenta la secuenciación de procesos en las diversas configuraciones productivas. En el caso de los ambientes Job Shop, se han desarrollado diferentes metodologías de secuenciación de máquinas, basadas en metaheurísticas tales como: Algoritmos genéticos [5][6], lógica difusa [7], búsquedas tabú [8], colonias de hormigas [9], sistemas artificiales inmunes [10] y células de aprendizaje automático [11]. De igual forma existen modelos en los que se han desarrollado sistemas híbridos de estas metaheurísticas [12].

La presente investigación plantea el uso de una metaheurística evolutiva denominada partículas inteligentes (PSO, Particle Swarm Optimization), a través de la cual se establece una metodología que permite mejorar la secuenciación de pedidos en los diferentes centros de trabajo en un ambiente Job Shop. PSO es un técnica de búsqueda estocástica basada en poblaciones evolutivas, la cual fue desarrollada por el Dr. Eberhart y el Dr. Kennedy [13], inspirada en las observaciones del comportamiento social de las bandadas de pájaros, los bancos de peces y en la teoría de enjambres. En este tipo de búsqueda basada en poblaciones, la solución actual se

sustituye por un conjunto de soluciones que recorren conjuntamente todo el espacio de soluciones, interactuando entre ellas. Además de los movimientos aplicables a las soluciones que forman parte de este conjunto, PSO contemplan otros operadores para generar nuevas soluciones a partir de las ya existentes. Por tanto, las soluciones llamadas partículas, empiezan a volar en el espacio de búsqueda, y son guiadas por la partícula que mejor solución ha encontrado hasta el momento y que actúa como líder de la bandada. Cada partícula evoluciona teniendo en cuenta la anterior solución y a su respectivo líder. En esta evolución las partículas modifican su velocidad hacia la mejor solución de su entorno, tomando como referencia la información de su líder [4].

Dada la estructura como se concibe PSO, se pueden destacar las siguientes características [14]:

- 1) PSO es robusto, versátil y de propósito general, lo que permite aplicar una solución a problemas de distinta índole sin necesidad de grandes cambios. Para la implementación del algoritmo se requiere sólo de las especificaciones del problema y de unos pocos parámetros. Luego se hace uso de operaciones matemáticas básicas mediante un proceso iterativo para hallar la solución. De esta manera, las exigencias computacionales, tanto en memoria como en velocidad de procesamiento, no son elevadas.
- 2) PSO se puede utilizar para resolver muchos de los mismos tipos de problemas que los algoritmos genéticos (AG). Sin embargo, a diferencia de los AG, PSO es un sistema que tiene memoria, lo que permite que las partículas que vuelan, recuerde siempre las soluciones óptimas que se encuentran en el proceso, para que puedan volver a ellas. Además, PSO utiliza menos cantidad de parámetros, lo que permite encontrar más fácilmente la combinación necesaria para alcanzar una solución óptima.
- 3) PSO se integra fácilmente con muchas técnicas de búsqueda local, lo cual permite mejorar la calidad de la solución. De igual forma el algoritmo se adapta bien a la paralelización mediante la implementación de un clúster de estaciones de trabajo.

El problema general es encontrar una programación de los pedidos de tal forma que se minimice el tiempo de proceso (makespan), tiempo de ocio (idle time) y se aumente la utilización de las máquinas [15]. La formalización matemática de la anterior teoría hace factible que la misma pueda ser aplicada en los procesos de secuenciación de la producción, solucionando problemas fundamentales en esta área como: repartición de recursos, asignación de máquinas, ordenación y secuenciación de los diferentes pedidos en cada una de las distintas máquinas, incumplimiento de plazos de

entrega, inapropiada estimación de la demanda, difícil manejo de las órdenes de compra, mal control de inventarios, frecuentes acciones de empuje de trabajos, desequilibrio en la capacidad de los centro de trabajo e insatisfacción de las condiciones de calidad [16].

2. DESCRIPCIÓN DEL PROBLEMA

En un ambiente de producción Job Shop, hay N trabajos que deben ser procesados a través de M máquinas. Suponga que cada trabajo debe pasar a través de cada máquina una y solo una vez. Cada trabajo debe ser procesado en las máquinas en un orden particular y no hay restricciones de precedencia entre los diferentes operaciones de trabajo. Cada máquina puede procesar sólo un trabajo a la vez y este no puede ser interrumpido. Además, el tiempo de proceso en cada máquina es fijo y conocido.

En PSO, el espacio de solución del problema es formulado como un espacio de búsqueda. La posición de cada partícula en el espacio de búsqueda es una solución correlacionada con el problema. Las partículas cooperan para determinar la mejor solución en el espacio de búsqueda. A continuación se presentan los pasos que se deben llevar a cabo para realizar un proceso de secuenciación mediante PSO. La metodología propuesta, empieza por considerar los supuestos, propuestos en [17]:

- i. Inicialmente es necesario suponer una población de K partículas, en la cual, cada partícula $X_k = (x_{k1}, \dots, x_{kD})$ representa una solución potencial y se define como un punto en un espacio D -dimensional.
- ii. Cada partícula K sobrevuela el espacio de soluciones hacia nuevas posiciones X_k , con un vector de velocidad $V_k = (v_{k1}, \dots, v_{kD})$.
- iii. La población de K partículas debe ser inicializada con posiciones y velocidades aleatorias, X_i y V_i , respectivamente.
- iv. Se debe definir una función de evaluación, *Fitness*, la cual permita clasificar cada una de las posiciones encontradas por el algoritmo.

Al calcular la función *Fitness* se asignan las mejores posiciones históricamente visitadas por la partícula, $pbest_k$, y por toda la población, $gbest$.

- v. La velocidad de cada partícula debe ser actualizada y acotada por un valor máximo impuesto en cada dimensión V_{max} , de acuerdo con la ecuación (1):

$$V_k = wV_k + c_1r_1(pbest_k - X_k) + c_2r_2(gbest - X_k), V_k \leq V_{Dmax} \quad \forall D \quad (1)$$

Donde w es la constante de inercia, c_1 y c_2 son los coeficientes de aceleración que determinan en qué medida la partícula es influenciada en su desplazamiento por su propia memoria ($pbest_i$) y por la cooperación social ($gbest$), y r_1 y r_2 representan dos números aleatorios con distribución uniforme $U[0,1]$, cuyo objetivo es introducir el comportamiento estocástico y un tanto impredecible que adoptan ciertos organismos en su desplazamiento.

Resultados empíricos han mostrado que una constante de inercia $w=0.7298$ y coeficientes de aceleración $c1=c2=1.49618$ proveen un buen comportamiento de convergencia [18][19].

- vi. Se debe actualizar la posición de la partícula K de acuerdo con la ecuación (2):

$$X_K = X_K + V_K \cdot \Delta t \quad (2)$$

Donde el paso temporal, Δt , normalmente se considera unidad, forzando que la nueva posición X_k quede dentro de los límites del espacio de soluciones.

- vii. Se requiere evaluar de nuevo la función *Fitness* de la partícula y si es necesario se debe actualizar su memoria $pbest_k$, y/o la mejor solución de conjunto $gbest$, respectivamente, en el caso de que el resultado hallado sea mejor que el que se tiene hasta ese momento.
- viii. La velocidad de la partícula (*Ec.1*) debe ser actualizada constantemente hasta que se complete el movimiento de todas las partículas de la población o se cumpla un criterio específico de terminación. Para ello es necesario iterar el procedimiento, desde el paso v tantas veces como se halla determinado en el criterio de terminación. Cuando se complete el último ciclo se debe salvar la solución $gbest$ como solución al problema antes de detener el algoritmo.

3. METODOLOGÍA

Para la solución del JSSP, se han descrito diversas técnicas relacionadas con la inteligencia artificial. Sin embargo, en esta sección se propone una metodología basada en PSO, tal cual se indica en la introducción.

3.1. Representación del problema

Según Thompson, Muth y Winters [20], el JSSP se puede representar mediante una matriz [operación, máquina]. En esta matriz, las filas representan los pedidos (P_i) y las columnas los centros de trabajo (C_j). $Tp(i,j)$ representa el tiempo de proceso del pedio i en la máquina j . La tabla 1 nos muestra un esquema general de la representación.

Tabla 1. Representación del problema JSSP NXM

	CT 1	CT 2	...	CT M
Pedido 1	Tp(1,1)	Tp(1,2)		Tp(1,M)
Pedido 2	Tp(2,1)	Tp(2,2)		Tp(2,M)
...				
Pedido N	Tp(N,1)	Tp(N,2)		Tp(N,M)

3.2. Soluciones iniciales

Las soluciones iniciales X_i se deben generar de acuerdo al numeral iii de la sección 2. Dado que estas se establecen de forma aleatoria, la única condición que deben cumplir, es que correspondan a una secuencia válida dentro del límite del espacio de soluciones establecido.

3.3. Evaluación

Para cada una de las soluciones, establecidas en el paso anterior, se debe definir un diagrama de Gantt (para cada solución), el cual establezca el orden de los procesos en el tiempo, en cada uno de los diferentes centros de trabajo. Una vez establecido el anterior diagrama se procede a evaluar cada una de las diferentes soluciones, con el fin de calcular los tiempos totales de proceso y los tiempos totales de ocio, bajo las siguientes funciones de cálculo *Fitness* [21]:

$$Fitness_{Makespan} = \min \sum_{i=1}^N \sum_{j=1}^M P_{ij} \quad (3)$$

$$Fitness_{idle} = \min \sum_{j=1}^M f_j \quad (4)$$

El objetivo fundamental, es minimizar las dos funciones *Fitness*. Donde N representa el número de trabajos, M , representa el número de máquinas, P_{ij} es el tiempo de procesamiento del trabajo i , en la máquina j y f_j , es el tiempo total ocio de la máquina j .

Dado que el tiempo total de ocio (idle time) es una consecuencia directa del tiempo de proceso (makespan), la búsqueda sólo es guiada por la función makespan, ecuación (3). Con base en el menor tiempo makespan, es calculado el tiempo total de ocio, según la ecuación (4).

Posteriormente, el proceso continúa hasta que se cumpla una de las dos siguientes opciones:

- i. Se encuentre una solución que se aproxime en un 99% a la solución óptima estimada (véase subsección 3.4).
- ii. Se completen 5000 iteraciones (véase sección 2 -viii).

3.4. Estimación del óptimo

Con el fin de calcular la aproximación de las soluciones encontradas, respecto a la mejor solución, es necesario estimar la solución óptima. Para estimar la solución

óptima, se considera que el tiempo de proceso óptimo nunca es inferior al máximo tiempo de proceso entre: El máximo tiempo empleado por los centros de trabajo, con tiempo muerto igual a cero y el tiempo de proceso total que emplea un pedido P_i , en pasar por todos los centros de trabajo, con un tiempo muerto igual a cero.

El tiempo óptimo estimado permitirá determinar la efectividad de la metodología propuesta y establecer el porcentaje de aproximación, de cada una de las posibles soluciones encontradas, respecto al óptimo general.

4. EXPERIMENTACIÓN

Considere un proceso de tres operaciones básicas, las cuales se pueden realizar indistintamente en tres diferentes centros de trabajo, con los tiempos de proceso ilustrados en la tabla 2. En este proceso, se supone una capacidad ilimitada por cada centro de trabajo, siendo el objetivo fundamental del proceso, secuenciar tres pedidos en los diferentes centros de trabajo, con base en los tiempos de procesos establecidos.

Tabla 2. Representación del problema JSSP 3X3.

	CT 1	CT 2	CT 3	Suma
Pedido 1	1	2	3	6
Pedido 2	5	1	2	8
Pedido 3	1	3	2	6
Suma	7	6	7	20

En esta tabla, la última columna (*suma*) representa el tiempo de proceso total que emplea un pedido (P_i) en pasar por todos los centros de trabajo, con un tiempo muerto igual a cero. Igualmente, la última fila representa la sumatoria de todos los tiempos de proceso de un pedido en un centro de trabajo (C_i), con un tiempo muerto igual a cero.

Como se indicó en la subsección 3.4., el mayor valor entre estos dos valores, puede ser considerado como un óptimo estimado. En este caso es 8.

5. RESULTADOS

La solución inicial del problema puede estar representada por la siguiente la matriz X_k

$$X_k = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \tag{5}$$

Donde cada una de las filas representa el centro de trabajo en la que debe ser procesado el pedido, y cada una de las columnas la referencia del pedido. La solución anterior es válida, pero no es la óptima. Es necesario recordar que un pedido no puede ser procesado al mismo tiempo en dos centros de trabajo diferentes, por tanto, el tiempo muerto del proceso presentado en la secuencia de la ecuación (5) es muy elevado y no es aproximado al

óptimo estimado. El diagrama de Gantt de la secuencia se muestra en la figura 1.

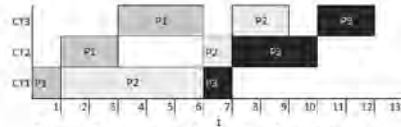


Figura 1. Diagrama de Gantt solución inicial.

Finalmente después de aplicar el procedimiento descrito en la sección 3, llegamos a la solución final, representada por la siguiente matriz:

$$X_k = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 1 & 2 \end{bmatrix} \tag{6}$$

La mejor solución encontrada, coincide en un 100% respecto a la solución óptima estimada, con un tiempo de computo estimado de 2.2 segundos El diagrama de Gantt para la solución final se encuentra en la figura 2:

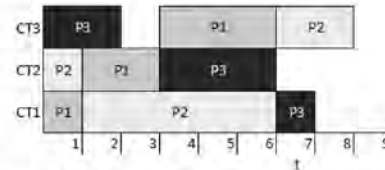


Figura 2. Diagrama de Gantt mejor solución encontrada.

6. CONCLUSIONES

El presente trabajo busca destacar la aplicación de las técnicas de inteligencia artificial como un medio a través del cual, diferentes empresas, en las que los sistemas de producción comprenden el desarrollo de un gran número de operaciones manuales, tengan la oportunidad de mejorar sus niveles de competitividad mediante herramientas que les permitan una adecuada programación de su producción.

En general, los resultados obtenidos mediante las técnicas de inteligencia artificial permiten encontrar soluciones óptimas o con niveles de aproximación superiores al 90% de la solución óptima. Sin embargo, dado que estas técnicas esta diseñadas para espacios continuos, su discretización puede generar soluciones invalidas, lo cual debe ser considerado a la hora de implementar el procedimiento, mediante el establecimiento de parámetros y condiciones de ciclo a través de las cuales se eviten las búsquedas sobre óptimos locales.

La metodología presentada logra obtener una solución que se aproxima a la solución óptima estimada en un

porcentaje superior al 99% con una robustez superior al 99.5% y un tiempo de cómputo prácticamente despreciable.

En futuras líneas de investigación, la metodología propuesta puede ser combinada con otras técnicas de inteligencia artificial como sistemas expertos, búsqueda tabú, minería de datos, etc., las cuales permitirán mejorar los resultados obtenidos.


7. REFERENCIAS

- [1] Sha, D.Y., and Hsing-Hung Lin. "A multi-objective PSO for job-shop scheduling problems." *Expert Systems with Applications*, 2010: 1065–1070.
- [2] Niu, Qun, Bin Jiao, and Xingsheng Gu. "Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time." *Applied Mathematics and Computation*, 2008: 148–158.
- [3] Jun-jie, Bai, Gong Yi-guang, and Wang Ning-sheng. "An improved PSO algorithm for flexible job shop scheduling with lot-splitting." *Intelligent Systems and Applications*, 2009: 1-5.
- [4] Brito Santana, Julio, et al. *Metaheurísticas: una revisión actualizada*. La Laguna, España: Universidad de La Laguna, 2004.
- [5] Manikas, Andrew, and Yiit-Long Chang. "Multi-criteria sequence-dependent job shop scheduling using genetic algorithms." *Computer & Industrial Engineering*, 2009: 179-185.
- [6] Chao-Hsien Pan, Jason, and Han-Chiang Huang. "A hybrid genetic algorithm for no-wait job shop scheduling problems." *Expert Systems with Applications*, 2009: 5800-5806.
- [7] Yun, Young Su. "Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems." *Computers & Industrial Engineering*, 2002: 623-644.
- [8] Buscher, Udo, and Liji Shen. "An integrated tabu search algorithm for the lot streaming problem in job shops." *European Journal of Operational Research*, 2009: 385–399.
- [9] Xing, Li-Ning, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, and Jian Xiong. "A knowledge-based ant colony optimization for flexible job shop scheduling problems." *Applied Soft Computing*, 2010: 888-896.
- [10] Ge, Hong-Wei, Liang Sun, Yan-Chun Liang, and Feng Qian. "An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling." *Systems and Humans*, 2008: 358-368.
- [11] Jafarpour, B, M R Meybodi, and S Shiry. "A hybrid method for optimization (Discrete PSO + CLA)." *International Conference on Intelligent and Advanced Systems*. IEEE Press, 2007. 55-60.
- [12] Fang Ming, Guo, and Liu Qiong. "A Hybrid PSO algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date." *Fifth International Conference on Natural Computation*. IEEE Press, 2009. 171-176.
- [13] Eberhart, Russell, and James Kennedy. "Particle swarm optimization." *Proceedings of the IEEE International Joint Conference on Neural Networks*, Perth, Australia: IEEE Press, 1995. 1942-1948.
- [14] Akjiratikarl, Chananes, Pisal Yenradee, and Paul R. Drake. "PSO-based algorithm for home care worker scheduling in the UK." *Computers & Industrial Engineering*, 2007: 559–583.
- [15] Sha, D Y, and Cheng-Yu Hsu. "A hybrid particle swarm optimization for job shop scheduling problem." *Computers & Industrial Engineering*, 2006: 791–808.
- [16] Miltenburg, John. *Manufacturing Strategy: How to formulate and implement a winning plan*. Ed. 1. Portland, Oregon: Productivity Press, 1995.
- [17] Pérez, Jesús R, and José Basterrechea. "Optimización con enjambre de Partículas aplicada a la reconstrucción del diagrama de radiación de antenas." *XX Congreso Nacional URSI*. Gandía, España, 2005.
- [18] Eberhart, R C, and Y Shi. "Comparing inertia weights and constriction factors in particle swarm." *Proceedings of the IEEE Congress on Evolutionary Computation*. San Diego, USA, 2000. 84–88.
- [19] Van Den Bergh, F, and A P Engelbrecht. "A study of particle swarm optimization particle trajectories." *Information Sciences*, 2006: 937–971.
- [20] Thompson, Gerald L, John F Muth, and Peter Winters. *Industrial scheduling*. Prentice-Hall international series in management. Englewood Cliffs, N.J: Prentice-Hall, 1963.
- [21] Castrillón, Omar D, William A Sarache, and Jaime A Giraldo. "Aplicación de un algoritmo evolutivo en la solución de problemas Job Shop - Open Shop." *Información Tecnológica (Centro de Información Tecnológica)*, 2011: Artículo en Prensa.

MINISTERIO DEL INTERIOR DIRECCION NACIONAL DE DERECHO DE AUTOR UNIDAD ADMINISTRATIVA ESPECIAL OFICINA DE REGISTRO		Libro - Tomo - Partida 13-33-265 Fecha de registro 16-May-2012
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO		
Page 1 of 2		
1. DATOS DE LAS PERSONAS		
AUTOR		
Nombres y apellidos	VICTOR FABIO SUAREZ	No de identificación CC 1053786048
Nacional de	COLOMBIA	--
Dirección	UNIVERSIDAD NACIONAL CAMPUS LA NUBIA	Ciudad MANIZALES
AUTOR		
Nombres y apellidos	ALVARO GUERRERO AGUIRRE	No de identificación CC 1053798716
Nacional de	COLOMBIA	--
Dirección	UNIVERSIDAD NACIONAL CAMPUS LA NUBIA	Ciudad MANIZALES
AUTOR		
Nombres y apellidos	OMAR DANILO CASTRILLON GOMEZ	No de identificación CC 76063287
Nacional de	COLOMBIA	--
Dirección	UNIVERSIDAD NACIONAL CAMPUS LA NUBIA	Ciudad MANIZALES
TITULAR DERECHO PATRIMONIAL		
Razón social	UNIVERSIDAD NACIONAL DE COLOMBIA - BOGOTA	No de identificación NIT 899999063
Dirección	KR 45 26 - 85 OF 515, ED. URIEL GUTIERREZ	Ciudad BOGOTA D.C.
PRODUCTOR		
Razón social	UNIVERSIDAD NACIONAL DE COLOMBIA - BOGOTA	No de identificación NIT 899999063
Dirección	KR 45 26 - 85 OF 515, ED. URIEL GUTIERREZ	Ciudad BOGOTA D.C.
2. DATOS DE LA OBRA		
Título original	SOFTWARE ASIGNACIÓN DE HORARIOS BASADA EN TÉCNICAS INTELIGENTES Y RITMOS COGNITIVOS	
Fecha de creación	2012	--
Pais de origen	COLOMBIA	
CLASE DE OBRA	INEDITA	
CARACTER DE LA OBRA	OBRA ORIGINARIA	
CARACTER DE LA OBRA	OBRA COLECTIVA	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	PROGRAMA DE COMPUTADOR	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	DESCRIPCIÓN DEL PROGRAMA	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	MATERIAL AUXILIAR	
TÍTULO ANTERIOR DE LA OBRA		

MINISTERIO DEL INTERIOR DIRECCION NACIONAL DE DERECHO DE AUTOR UNIDAD ADMINISTRATIVA ESPECIAL OFICINA DE REGISTRO		Libro - Tomo - Partida 13-33-265 Fecha de registro 16-May-2012
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO		
Page 2 of 2		
Nombres y apellidos	DAVID ALFREDO CARDENAS OROZCO	No de identificación 80729578
Nacional de	COLOMBIA	En representación de UNIVERSIDAD NACIONAL DE COLOMBIA
Dirección	CRA 45 26-85 OF 515	Teléfono 3166075 Ciudad BOGOTA D.C.
Correo electrónico	ofin_nal@unal.edu.co	Radicación de entrada 1-2012-20915
 OSCAR EDUARDO SALAZAR ROJAS JEFE DE LA OFICINA DE REGISTRO		
<small>Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).</small>		

C. Anexo: Certificación prueba piloto



INSTITUCIÓN EDUCATIVA SAN JORGE BILINGÜE
Educamos en el amor generando sabiduría y proyectándolo a la sociedad.
RESOLUCIÓN N°. 853 24 DE JUNIO DE 2008 DE LA SECRETARÍA DE EDUCACIÓN MUNICIPAL
Nit. 900.070.015 - 4

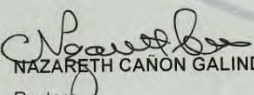
Manizales, 12 de octubre de 2012

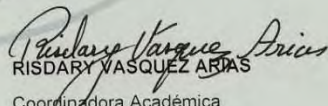
Señores
UNIVERSIDAD NACIONAL DE COLOMBIA
Sede Manizales

Asunto: Certificación del desarrollo de la fase experimental del trabajo de grado del docente Victor Fabio Suarez Chilma

Mediante la presente se certifica que la Institución Educativa San Jorge facilitó que el docente Victor Fabio Suarez Chilma realizara la fase experimental de su trabajo de grado, proceso en el cual se implementó la jornada académica para grupos de sexto a once de acuerdo a la aplicación informática desarrollada por el docente. De igual manera, se le suministró la información académica de los grupos para realizar el análisis de sus resultados durante los dos primeros periodos académicos.

Para constancia se firma en Manizales a los 12 días del mes de octubre de 2012.


NAZARETH CAÑÓN GALINDO
Rectora


RISDAY VASQUEZ ARIAS
Coordinadora Académica

Carrera 21 N° 49 - 30 Teléfono: 885 4181 Manizales

Bibliografía

1. Abdullah, S., Turabieh, H., McCollum, B., & McMullan, P. (2012). A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics* , 1-23.
2. Abdul-Rahman, S., Burke, E. K., Bargiela, A., McCollum, B., & Özcan, E. (2011). A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of Operations Research* .
3. Abramson, D. (1991). Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science* , 98-113.
4. Abramson, D., & Abela, J. (1992). A Parallel Genetic Algorithm for Solving the School Timetabling Problem. *15 Australian Computer Science Conference*, (págs. 1-9). Hobart.
5. Alba Torres, E. (1999). *Análisis y diseño de Algoritmos Genéticos Paralelos Distribuidos*. Tesis Doctoral, Universidad de Málaga, Málaga.
6. Al-Betar, M. A., & Khader, A. T. (2012). A harmony search algorithm for university course timetabling. *Annals of Operations Research* , 3-31.
7. Avella, P., D'Auria, B., Salerno, S., & Vasil'ev, I. (2007). A computational study of local search algorithms for Italian high-school timetabling. *Journal of Heuristics* , 543-556.
8. Ayob, M., Hamdan, A. R., Abdullah, S., Othman, Z., Zakree, M., Nazri, A., et al. (2011). Intelligent Examination Timetabling Software. *Procedia Social and Behavioral Sciences* , 600-608.
9. Banco Mundial. (2009). *La calidad de la educación en Colombia: Un análisis y algunas opciones para un programa de política*. Bogotá: Misión residente en Colombia.

10. Beligiannis, G., Moschopoulos, C., & Likothanassis, S. (2009). A genetic algorithm approach to school timetabling. *Journal of the Operational Research Society* , 23--42.
11. Birbas, T., Daskalaki, S., & Housos, E. (2009). School timetabling for quality student and teacher schedules. *Journal of Scheduling* , 177-197.
12. Boland, N., Hughes, B. D., Merlot, L. T., & Stuckey, P. J. (2008). New integer linear programming approaches for course timetabling. *Computers & Operations Research* , 2209-2233.
13. Brito Santana, J., Campos Rodríguez, C., García López, F. C., García Torres, M., Melián Batista, B., Moreno Pérez, J. A., y otros. (2004). *Metaheurísticas: una revisión actualizada*. La Laguna, España: Universidad de La Laguna.
14. Burke, E. K., Kendall, G., & Soubeiga, E. (2003). A Tabu-Search Hyperheuristic for Timetabling. *Journal of Heuristics* , 451-470.
15. Burke, E. K., Kendall, G., Mısırlı, M., & Özcan, E. (2010). Monte Carlo hyper-heuristics for examination timetabling. *Annals of Operations Research* .
16. Burke, E. K., Pham, N., Qu, R., & Yellen, J. (2012). Linear combinations of heuristics for examination timetabling. *Annals of Operations Research* , 89-109.
17. Burke, E., Elliman, D., & Weare, R. (1994). A Genetic Algorithm Based University Timetabling System. *East-West Conference on Computer Technologies in Education*. Crimea, Ukraine.
18. Buscher, U., & Shen, L. (2009). An integrated tabu search algorithm for the lot streaming problem in job shops. *European Journal of Operational Research* , 385-399.
19. Caballero Rosas, J. A., & Gómez Cardona, J. P. (2004). *Desarrollo de timetabling enfocado a la programación de exámenes finales*. Medellín, Colombia: Universidad EAFIT - Departamento de ingeniería de producción.
20. Čangalović, M., & Schreuder, J. A. (1991). Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research* , 248-258.

21. Carranza, H. E., Chávez, L., Fissore, M. L., & Simón, S. B. (2008). Un enfoque multiobjetivo para la asignación de canales en sistemas celulares. *Información Tecnológica* , 87-96.
22. Carrasco, M., & Pato, M. (2001). A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem. *Lecture Notes in Computer Science* , 3-17.
23. Chao-Hsien Pan, J., & Huang, H.-C. (2009). A hybrid genetic algorithm for no-wait job shop scheduling problems. *Expert Systems with Applications* , 5800-5806.
24. Chávez Bosquez, O. A., De los Santos Torres, G., & Gómez Ramos, J. L. (2005). *Búsqueda tabú aplicada a un problema NP-Completo: Generación de horarios en la DAIS*. Universidad Juárez Autónoma de Tabasco.
25. Cheong, C. Y., Tan, K. C., & Veeravalli, B. (2009). A multi-objective evolutionary algorithm for examination timetabling. *Journal of Scheduling* , 121-146.
26. Colorni, A., Dorigo, M., & Maniezzo, V. (1990). A Genetic Algorithm to Solve the Timetabling Problem. *Computational Optimization and Applications Journal* .
27. Correa Flórez, C. A., Bolaños, R. A., & Molina Cabrera, A. (2008). Algoritmo multiobjetivo NSGA-II aplicado al problema de la mochila. *Scientia Et Technica* , 206-211.
28. Dammak, A., Elloumi, A., Kamoun, H., & Ferland, J. A. (2008). Course Timetabling at a Tunisian University: A Case of Study. *Journal of Systems Science and Systems Engineering* , 334-352.
29. Daskalaki, S., & Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research* , 106-120.
30. Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research* , 117-135.
31. De Causmaecker, P., Demeester, P., & Vanden Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research* , 307-318.
32. De La Cruz Nogales, F. M. (2009). Las estrategias de aprendizaje: Técnicas para el estudio. *Revista Digita Innovación y Experiencias Educativas* .

33. De Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research* , 151-162.
34. Demeester, P., Bilgin, B., De Causmaecker, P., & Vanden Berghe, G. (2012). A hyperheuristic approach to examination timetabling problems: benchmarks and a new problem from practice. *Journal of Scheduling* , 83-103.
35. Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. *Proceedings of the IEEE International Joint Conference on Neural Networks* (págs. 1942-1948). Perth, Australia: IEEE Press.
36. Erben, W., & Keppler, J. (1996). A Genetic Algorithm Solving a Weekly Course-Timetabling Problem. *Lecture Notes in Computer Science, 1996, Volume 1153, Practice and Theory of Automated Timetabling* , 192-211.
37. Estaún Ferrer, S. (2002). Ontogénesis de los ritmos atencionales: ritmos endógenos o exógenos. *Seminario Internacional Complutense « Ritmos Psicológicos y Jornada Escolar »* (págs. 49-64). Madrid: Facultad de Psicología de la U.C.M.
38. Fang Ming, G., & Qiong, L. (2009). A Hybrid PSO algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Fifth International Conference on Natural Computation* (págs. 171-176). IEEE Press.
39. Feito Alonso, R. (2007). Tiempos escolares: El debate sobre la jornada escolar continua y partida. *Cuadernos de pedagogía* , 74-79.
40. Fernández Enguita, M. (2000). *La hora de la escuela. Análisis y valoración de los procesos, los efectos y las opciones de implantación de la jornada continua*. Salamanca: Universidad de Salamanca.
41. Flores Pichardo, M. (2011). Revisión de algoritmos genéticos aplicados al problema de la programación de cursos universitarios. *Programación Matemática y Software* , 49-65.
42. Fogel, D. B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics* , 139-144.
43. Franco Baquero, J. F., Toro Ocampo, E. M., & Gallego Rendón, R. A. (2008). Problema de asignación óptima de salones resuelto con Búsqueda Tabú. *Ingeniería & Desarrollo* , 149-175.

44. Ge, H.-W., Sun, L., Liang, Y.-C., & Qian, F. (2008). An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling. *Systems and Humans* , 358-368.
45. Glover, F., & Melián, B. (2003). Búsqueda Tabú. *Revista Iberoamericana de Inteligencia Artificial* , 29-48.
46. Gogos, C., Alefragis, P., & Housos, E. (2012). An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research* , 203-221.
47. Goldberg, D. E., & Lingle, J. R. (1985). Alleles, loci and the traveling salesman problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, (págs. 154-159).
48. Granada E., M., Toro Ocampo, E. M., & Franco Baquero, J. F. (2006). Programación óptima de horarios de clase usando un algoritmo memético. *Scientia et Technica* , 255-260.
49. Hao, J.-K., & Benlic, U. (2011). Lower bounds for the ITC-2007 curriculum-based course timetabling problem. *European Journal of Operational Research* , 464-472.
50. Hederich Martinez, C., Camargo Uribe, Á., & Reyes Cuervo, M. E. (2004). *Ritmos cognitivos en la escuela*. Bogotá, Colombia: Universidad Pedagógica Nacional.
51. Hernández, R., Miranda P, J., & Rey, P. A. (2008). Programación de horarios de clases y asignación de salas para la facultad de ingeniería de la Universidad Diego Portales mediante un enfoque de programación entera. *Revista Ingeniería de Sistemas* , 121-141.
52. Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press: Ann Arbor.
53. Ibarra Florencio, N. (2011). *Tipos de Lenguajes Formales*. México D.F.: Centro de Ciencias de la Complejidad.
54. Jafarpour, B., Meybodi, M. R., & Shiry, S. (2007). A hybrid method for optimization (Discrete PSO + CLA). *International Conference on Intelligent and Advanced Systems* (págs. 55-60). IEEE Press.
55. Kahar, M., & Kendall, G. (2010). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research* , 557-565.

56. Kieran Burke, E., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research* , 266-280.
57. Kung, H. T., Luccio, F., & Preparata, F. T. (1975). On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery* , 469-476.
58. Lewis, R., Paechter, B., & Rossi-Doria, O. (2007). Metaheuristics for University Course Timetabling. *Studies in Computational Intelligence* , 237-272.
59. Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research* , 498-516.
60. Lü, Z., & Hao, J.-K. (2010). Adaptive Tabu Search for course timetabling. *European Journal of Operational Research* , 235-244.
61. Luna Valero, F. (2008). *Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones*. Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación. Málaga: Universidad de Málaga.
62. Maciques Rodríguez, E. (2004). *Trastornos del aprendizaje. Estilos de aprendizaje y el diagnóstico psicopedagógico*. Obtenido de http://www.sld.cu/galerias/pdf/sitios/rehabilitacion-equino/trastornos_del_aprendizaje_y_estilos_de_aprendizaje_1.pdf
63. Maldonado, C. E., & Gómez Cruz, N. A. (2010). *Modelamiento y simulación de sistemas complejos*. Bogotá D.C.: Universidad del Rosario.
64. Manikas, A., & Chang, Y.-L. (2009). Multi-criteria sequence-dependent job shop scheduling using genetic algorithms. *Computer & Industrial Engineering* , 179-185.
65. Mansour, N., Isahakian, V., & Ghalayini, I. (2011). Scatter search technique for exam timetabling. *Applied Intelligence* , 299-310.
66. McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., & Qu, R. (2012). A new model for automated examination timetabling. *Annals of Operations Research* , 291-315.

67. Mejía Caballero, J. M. (2008). *Asignación de horarios de clases universitarias mediante algoritmos evolutivos*. Barranquilla: Universidad del Norte.
68. Mejía Caballero, J. M., & Paternina Arboleda, C. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Educación en Ingeniería* , 140-149.
69. MEN. (2002). *Decreto 1850*. Bogotá: Ministerio de Educación Nacional.
70. MEN. (2012). *Ministerio de Educación Nacional*. Recuperado el 11 de Febrero de 2012, de <http://menweb.mineduacion.gov.co/seguimiento/estadisticas/principal.php>
71. Mendoza Crisostomo, P., Flores Trujillo, I. A., & Morales Genis, D. (2009). Algoritmo Evolutivo para generar cargas académicas en TIC-SI. *4to Simposio Internacional en Sistemas Inteligentes y Organizaciones Inteligentes*. Xalapa, México.
72. Minh, K., Thanh, N., Trang, K., & Hue, N. (2010). Using tabu search for solving a high school timetabling problem. *Studies in Computational Intelligence* , 305-313.
73. Montoya-Torres, J. R., Aponte Penagos, A. F., & Rosas Castro, P. A. (2010). Un procedimiento de búsqueda voraz adaptativo probabilista para un problema monoproducción de localización de instalaciones no capacitado. *Ingeniería y Desarrollo* , 15-32.
74. Moujahid, A., Inza, I., & Larrañaga, P. (2008). *Algoritmos Genéticos*. Universidad del País Vasco - Euskal Herriko Unibertsitatea UPV-EHU, Departamento de Ciencias de la Computación e Inteligencia Artificial.
75. Nothegger, C., Mayer, A., Chwatal, A., & Raidl, G. R. (2012). Solving the post enrolment course timetabling problem by ant colony optimization. *Annals of Operations Research* , 325-339.
76. Pais, T. C., & Amaral, P. (2012). Managing the tabu list length using a fuzzy inference system: an application to examination timetabling. *Annals of Operations Research* , 341-363.
77. Patterson, J. H., Talbot, F. B., Slowinski, R., & Weglarz, J. (1990). Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. *European Journal of Operational Research* , 68-79.

78. Peñuela Meneses, C. A., & Granada Echeverri, M. (2007). Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no-dominado (NSGA-II). *Scientia et Technica* , 175-180.
79. Peñuela, C. A., Franco B, J. F., & Toro O, E. M. (2008). Colonia de hormigas aplicada a la programación óptima de horarios de clase. *Scientia Et Technica* , 49-54.
80. Pillay, N., & Banzhaf, W. (2010). An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing* , 457-467.
81. Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog Verlag.
82. Reeves, C. R. (1993). Heuristic Techniques for Combinatorial Problems. *Blackwell Scientific Publishing* .
83. Ruiz, S., Castrillón, O. D., & Sarache, W. A. (2012). Una metodología multiobjetivo para optimizar un ambiente job shop. *Información Tecnológica* , 35-46.
84. Sabar, N. R., Ayob, M., Qu, R., & Kendall, G. (2011). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence* .
85. Sadeh, N., Sycara, K., & Xiong, Y. (1995). Backtracking techniques for the job shop scheduling constraint satisfaction problem. *Artificial Intelligence* , 455-480.
86. Schwefel, H. P. (1965). *Kybernetische evolution als strategie der experimentellen Forschung in der Strömungstechnik*. Technical University of Berlin.
87. Schweickardt, G., & Miranda, V. (2010). Metaheurística FEPSO aplicada a problemas de Optimización Combinatoria: Balance de Fases en Sistemas de Distribución Eléctrica. *Ciencia, docencia y tecnología* , 133-163.
88. Soza, C., Landa Becerra, R., Riff, M. C., & Coello Coello, C. A. (2011). Solving timetabling problems using a cultural algorithm. *Applied Soft Computing* , 337-344.
89. Studenovský, J. (2009). Polynomial reduction of time space scheduling to time scheduling. *Discrete Applied Mathematics* , 1364-1378.

90. Tassopoulos, I. X., & Beligiannis, G. N. (2012). Using particle swarm optimization to solve effectively the school timetabling problem. *Soft Computing* .
91. Tassopoulos, I. X., & Beligiannis, G. N. (2012). Using particle swarm optimization to solve effectively the school timetabling problem. *Soft Computing* .
92. Turabieh, H., & Abdullah, S. (2011). An integrated hybrid approach to the examination timetabling problem. *Omega* , 598-607.
93. Urrutia Galicia, J. L., Alcérreca Huerta, J. C., & Ordaz Alcántara, M. A. (2008). Programación lineal con espacios covariante y contravariante. Una perspectiva física y matemática. *Ingeniería, investigación y tecnología* , 185-204.
94. Valouxis, C., & Housos, E. (2003). Constraint programming approach for school timetabling. *Computers & Operations Research* , 1555-1572.
95. Wren, A. (1996). Scheduling, Timetabling and Rostering - A Special Relationship? *Practice and theory of automated timetabling* (págs. 46-75). Edinburgh: Springer.
96. Wu, C.-C. (2011). Parallelizing a CLIPS-based course timetabling expert system. *Expert Systems with Applications* , 7517-7525.
97. Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S., & Xiong, J. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing* , 888-896.
98. Yun, Y. S. (2002). Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems. *Computers & Industrial Engineering* , 623-644.