



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO  
UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

Horarios de cursos personales para estudiantes universitarios basados en  
algoritmos genéticos y recocido simulado

TESIS

para obtener el grado de  
Maestra en Ciencias de la Computación

Presenta:

Ing. Brenda Sunuami González López

Tutor académico

Dr. René Arnulfo García Hernández

Tutores adjuntos

Dra. Yulia Nikolaevna Ledeneva

Dr. Ángel Hernández Castañeda

# Resumen

---

En este trabajo se genera la Asignación de Cargas Horarias para estudiantes (ACH), a partir de un catálogo de cursos publicado (CCP) y un enfoque evolutivo para resolverlo. ACH es el resultado de un proceso manual o automático de un sistema de programación de cursos universitarios para asignar clases a horarios y espacios, considerando restricciones y preferencias del estudiante. Sin embargo, independientemente de cómo se haya creado la ACH, cada estudiante universitario se enfrenta a limitaciones particulares para seleccionar los cursos de su horario. Debido a la complejidad de generar el horario manualmente, los estudiantes no consideraron una mejor opción. En este trabajo presentamos un método basado en un algoritmo genético y recocido simulado que considera las limitaciones y preferencias que presentan los estudiantes de la Universidad Autónoma del Estado de México para generar una ACH. De acuerdo con la experimentación con un conjunto de datos fundamental, el método propuesto no solo muestra mejores resultados cuantitativos que la generación manual del horario, sino que también muestra buenos resultados cualitativos basados en la evaluación de los estudiantes.

# Contenido

<b>DEDICATORIA</b> .....	<b>IV</b>
<b>AGRADECIMIENTOS</b> .....	<b>IV</b>
<b>PUBLICACIONES</b> .....	<b>V</b>
<b>RESUMEN</b> .....	<b>VI</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>IX</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>X</b>
<b>CAPÍTULO 1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1. PLANTEAMIENTO DEL PROBLEMA.....	2
1.2. HIPÓTESIS.....	2
1.3. OBJETIVO GENERAL.....	3
1.4. OBJETIVOS ESPECÍFICOS .....	3
1.5. DELIMITACIÓN .....	3
1.6. ESTRUCTURACIÓN DE LA TESIS.....	3
<b>CAPÍTULO 2. MARCO TEÓRICO</b> .....	<b>4</b>
2.1. ALGORITMOS GENÉTICOS (AG) .....	5
2.1.1. CROMOSOMAS O INDIVIDUOS .....	5
2.1.2. POBLACIÓN INICIAL.....	5
2.1.3. OPERADORES GENÉTICOS .....	5
2.1.3.1. OPERADOR DE SELECCIÓN .....	5
2.1.3.2. OPERADOR DE CRUZA O RECOMBINACIÓN .....	6
2.1.3.3. OPERADOR DE MUTACIÓN .....	6
2.2. OPTIMIZACIÓN MULTIOBJETIVO.....	6
2.3. MÉTODOS HEURÍSTICOS.....	7
2.3.1. ALGORITMOS EVOLUTIVOS (AE) .....	7
2.4. RECOCIDO SIMULADO.....	7
2.5. RESUMEN DEL CAPÍTULO .....	8
<b>CAPÍTULO 3. ESTADO DEL ARTE</b> .....	<b>9</b>
3.1. APLICACIÓN DE RECOCIDO SIMULADO PARA LA GENERACIÓN AUTOMÁTICA DE HORARIOS .....	10
3.2. ALGORITMOS MEMÉTICOS (CON APLICACIÓN A INSTITUCIÓN) .....	10
3.3. PROGRAMACIÓN ENTERA (CON APLICACIÓN A INSTITUCIÓN) .....	10
3.4. BÚSQUEDA TABÚ (CON APLICACIÓN A INSTITUCIÓN) .....	11
3.5. RITMOS COGNITIVOS (CON APLICACIÓN A PROFESORES) .....	12
3.6. RESUMEN DEL CAPÍTULO .....	14

<b>CAPÍTULO 4. MÉTODO PROPUESTO .....</b>	<b>15</b>
4.1. DESCRIPCIÓN DEL MÉTODO PROPUESTO .....	16
4.1.1. DESCRIPCIÓN DEL PLAN DE ESTUDIOS.....	16
4.1.2. RESTRICCIONES .....	17
4.1.3. LAS CONDICIONES PRIMARIAS .....	17
4.1.4. LAS CONDICIONES SECUNDARIAS.....	17
4.1.5. DESCRIPCIÓN FORMAL.....	17
4.1.6. FUNCIÓN DE APTITUD Y CONDICIONES DEL ESTUDIANTE .....	18
4.2. ALGORITMO GENÉTICO .....	19
4.2.1. POBLACIÓN .....	19
4.2.2. INDIVIDUO .....	19
4.2.3. FUNCIÓN DE EVALUACIÓN .....	20
4.2.4. OPERADORES EVOLUTIVOS .....	20
4.3. RECOCIDO SIMULADO.....	21
4.3.1. TEMPERATURA INICIAL .....	21
4.3.2. FUNCIÓN DE ENFRIAMIENTO .....	21
4.4. RESUMEN .....	22
<b>CAPÍTULO 5. EXPERIMENTACIÓN.....</b>	<b>23</b>
5.1. DESCRIPCIÓN DE LOS DATOS UTILIZADOS .....	24
5.1.1. RESULTADOS DE CONDICIONES DE LA FUNCIÓN DE APTITUD.....	24
5.2. EXPERIMENTOS CON ALGORITMOS GENÉTICOS .....	25
5.2.1. RESULTADOS CON ALGORITMOS GENÉTICOS .....	30
5.3. EXPERIMENTOS CON RECOCIDO SIMULADO .....	32
5.4. RESUMEN .....	37
<b>CAPÍTULO 6. CONCLUSIONES.....</b>	<b>38</b>
6.1. APORTACIONES.....	39
<b>REFERENCIAS.....</b>	<b>40</b>

# Índice de tablas

<b>TABLA 1.</b> TRABAJOS RELACIONADOS CON LA ASIGNACIÓN DE CARGAS HORARIAS. ....	13
<b>TABLA 2.</b> CONDICIONES DE LOS ESTUDIANTES PARA GENERAR CARGAS HORARIAS MANUAL Y AUTOMÁTICAMENTE PARA 25 ESTUDIANTES EVALUADOS CON LA FUNCIÓN DE APTITUD PROPUESTA .....	25
<b>TABLA 3.</b> COMPARACIÓN DE LA CARGA HORARIA MANUAL Y AUTOMÁTICAMENTE CON LA FUNCIÓN DE APTITUD PARA LOS 25 ESTUDIANTES DE LA TABLA 2. ADEMÁS, LA COMPARACIÓN MUESTRA, EN LA COLUMNA DE LA DERECHA, EL PUNTO DE VISTA DE LOS ESTUDIANTES DE LA CARGA HORARIA AUTOMÁTICA CON RESPECTO A SUS CONDICIONES, ESTÁS SE DIVIDEN EN CUATRO CATEGORÍAS: PEOR, IGUAL, EQUIVALENTE Y MEJOR. ....	31
<b>TABLA 4.</b> COMPARACIÓN DE LA CARGA HORARIA MANUAL Y AUTOMÁTICAMENTE, TOMANDO EN CUENTA LA TÉCNICA DE RECOCIDO SIMULADO CON LA FUNCIÓN DE APTITUD PARA 25 ESTUDIANTES. ADEMÁS, LA COMPARACIÓN MUESTRA EN LA COLUMNA DE LA DERECHA, EL PUNTO DE VISTA DE LOS ESTUDIANTES DE LA CARGA HORARIA AUTOMÁTICA.....	35

# Índice de figuras

<b>FIGURA 1.</b> EJEMPLO DE CRUZA O RECOMBINACIÓN EN UN ALGORITMO GENÉTICO. ....	6
<b>FIGURA 2.</b> MAPA CURRICULAR DE INGENIERÍA DE SOFTWARE.....	16
<b>FIGURA 3.</b> MÉTODO PROPUESTO. ....	20
<b>FIGURA 4.</b> CATÁLOGO DE CURSOS QUE BRINDA LA INSTITUCIÓN ACADÉMICA.....	24
<b>FIGURA 5.</b> PROBABILIDAD DE CRUZA PARA EL ESTUDIANTE 1. ....	26
<b>FIGURA 6.</b> PROBABILIDAD DE MUTACIÓN PARA EL ESTUDIANTE 1. ....	26
<b>FIGURA 7.</b> EVOLUCIÓN DEL AG PARA EL ESTUDIANTE 1.....	27
<b>FIGURA 8.</b> PROBABILIDAD DE CRUZA PARA EL ESTUDIANTE 2. ....	27
<b>FIGURA 9.</b> PROBABILIDAD DE MUTACIÓN PARA EL ESTUDIANTE 2. ....	28
<b>FIGURA 10.</b> EVOLUCIÓN DE AG PARA EL ESTUDIANTE 2. ....	28
<b>FIGURA 11.</b> PROBABILIDAD DE CRUZA PARA EL ESTUDIANTE 3. ....	29
<b>FIGURA 12.</b> PROBABILIDAD DE MUTACIÓN PARA EL ESTUDIANTE 3. ....	29
<b>FIGURA 13.</b> EVOLUCIÓN DE AG PARA EL ESTUDIANTE 3. ....	30
<b>FIGURA 14.</b> RESULTADOS CUALITATIVOS POR LOS ESTUDIANTES. ....	32
<b>FIGURA 15.</b> RECOCIDO SIMULADO PARA E1.....	33
<b>FIGURA 16.</b> DESEMPEÑO DE RECOCIDO SIMULADO ESTUDIANTE 2.....	33
<b>FIGURA 17.</b> RECOCIDO SIMULADO PARA E3.....	34
<b>FIGURA 18.</b> RESULTADOS CUALITATIVOS POR LOS ESTUDIANTES.....	36



# Capítulo 1.

## Introducción

---

La solución de problemas se ha visto beneficiada cada vez más por técnicas computacionales que le han permitido al humano, por un lado, tener mejores resultados y por el otro, realizar la misma tarea en un menor tiempo. La generación de cargas horarias no es una excepción, los estudiantes enfrentan cada período de inscripción el problema de diseñar su carga horaria. Este tipo de problema se conoce como programación horaria (en inglés, *timetabling*) (Hernández, 2008).

Los problemas de esta tarea consisten en la asignación de recursos logísticos, humanos y materiales a la programación de eventos en una calendarización que normalmente se repite por semana (González, 2018).

La mayoría de las instituciones universitarias cuentan con planes de estudio flexibles. Por lo que la programación horaria de cursos queda a cargo de los estudiantes donde el mismo debe de elegir, entre una oferta de Unidades de Aprendizaje (también conocidas como clases o materias), aquellas que va cursar durante el siguiente período (Sabatier, 2008). La flexibilidad del plan de estudios trae consigo varios beneficios porque puede generarse una carga académica adecuada a las condiciones particulares de cada estudiante. Sin embargo, el número de condiciones individuales y la gran cantidad de Unidades de Aprendizaje (UAs) ofertadas generan una gran combinación de posibilidades.

En la UAEMex (Universidad Autónoma del Estado de México), el estudiante genera su carga horaria de forma manual por lo que demora un largo período de tiempo y en varios casos se obtiene una mala carga horaria por no satisfacer las condiciones del estudiante. Una carga horaria incorrecta produce problemas académicos, alimenticios y económicos. Por ejemplo, si un estudiante selecciona todas las unidades de aprendizaje de manera continua entonces no tiene tiempo para alimentarse adecuadamente. Otro ejemplo es cuando un estudiante tiene que asistir a la universidad más días de los necesarios por lo que afecta en los recursos de tiempo y económicos empleados. Para conocer cuáles son los criterios involucrados en la generación de cargas horaria se hicieron encuestas tanto a los estudiantes como a los tutores encargados de guiar o los estudiantes en su avance en el plan de estudios.

En este trabajo se propone un método para generar automáticamente cargas horarias bajo los criterios que comúnmente utilizan los estudiantes y los tutores para la generación de cargas horarias mediante un algoritmo genético y recocido simulado, herramientas de las ciencias computacionales.

### *1.1. Planteamiento del problema*

Dado que existen muchas posibilidades de horarios y condiciones que el estudiante debe considerar en su carga horaria y muchas de estas condiciones son comunes en los estudiantes, entonces la asignación de cargas horarias se trata de un problema de optimización, el cual debe ser valorado de acuerdo con las condiciones que se satisfagan al estudiante, por lo cual surge la siguiente pregunta de investigación, Dadas las condiciones de cada estudiante, ¿Se podrán generar cargas horarias automáticamente optimizadas para estudiante aplicando algoritmos genéticos o recocido simulado?

### *1.2. Hipótesis*

Si se modelan las condiciones de los estudiantes agrupados en criterios normalizados, entonces sería posible formular una función que evalúe una carga horaria de acuerdo a las condiciones del estudiante, lo que sugeriría el uso de un algoritmo de optimización como lo es el algoritmo genético o el recocido simulado para obtener la mejor carga horaria.

### *1.3. Objetivo general*

Implementar un algoritmo genético y recocido simulado (de manera independiente uno del otro) para asignar horarios de cursos personales para estudiantes universitarios.

### *1.4. Objetivos específicos*

- Estudiar y modelar las condiciones involucradas en la generación de cargas horarias de estudiantes.
- Implementar la codificación del algoritmo genético y recocido simulado, así como la base de datos que contiene las cargas horarias de los estudiantes, los horarios brindados por la institución y las condiciones del estudiante.

### *1.5. Delimitación*

En el presente trabajo se implementan dos algoritmos de optimización un algoritmo genético y uno de recocido simulado. El método propuesto será probado únicamente con datos propios de la Unidad Académica Profesional Tlanguistenco.

### *1.6. Estructuración de la tesis*

La estructura de la tesis será de la siguiente manera:

Capítulo I. Introducción: En este capítulo se representa una introducción al problema Horarios de cursos personales para estudiantes universitarios basados en algoritmos genéticos y recocido simulado.

Se presenta la hipótesis que es la encargada de clarificar el planteamiento del problema a tratar y el objetivo general para identificar el propósito principal de la investigación sobre la asignación de cargas horarias personalizadas para estudiantes probada con dos técnicas propias de las ciencias computacionales.

Capítulo II. Marco teórico: Se describen los conceptos empleados en esta investigación de asignación de cargas horarias personalizadas para estudiantes.

Capítulo III. Estado del arte: Se describen las investigaciones realizadas que están relacionadas con la asignación de cargas horarias personalizadas para estudiantes.

Capítulo IV. Método propuesto: Se indica el método a implementar, el cual es mediante un algoritmo genético y también se trabajó con recocido simulado.

Capítulo V. Experimentación: Se realiza la experimentación que corrobora lo establecido en la hipótesis anteriormente mencionada.

Capítulo VI. Conclusiones: Se muestran los objetivos alcanzados, así como los posibles trabajos futuros encontrados dentro de esta investigación.



## CAPÍTULO 2.

# Marco Teórico

---

En este capítulo se describen los conceptos y definiciones necesarias para la comprensión de este trabajo de investigación.

El procesamiento del lenguaje natural, es un área de estudio de la Inteligencia artificial, donde se pretende que las computadoras sean capaces de entender, el procesamiento del lenguaje natural trata de emplear matemáticas, estadística, computación y la lingüística(Ledeneva & Hernández, 2017).

## 2.1. Algoritmos genéticos (AG)

De acuerdo con John Holland, 1975 los AG están basados en el proceso genético de los organismos vivos. Un AG es un método que tiene como función optimizar algún problema, en donde la finalidad es obtener un conjunto de soluciones óptimas, el obtener el mejor individuo es parte específica del algoritmo, en particular de la selección de individuos, un genético tiene una población que consta de un determinado número de individuos donde se toman los mejores y entonces se crean nuevos, realizando así una generación nueva.

Los AG son métodos que hacen la búsqueda y optimización de problemas, son utilizados para poder buscar la optimización de algún problema, este autor dice que los algoritmos genéticos están basados en la reproducción sexual (Gestal, 2010). Los principios básicos de los AG fueron establecidos por Holland (1975) y se encuentran descritos en varios textos – Goldberg (1989), Davis (1991), Michalewicz (1992), Reeves (1993)

La eficiencia de los AG define que se trata de una técnica robusta y pueden tratar una gran variedad de problemas computacionales provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades o casi imposibles de tratar. Sin embargo, no se garantiza que el AG encuentre la solución óptima del problema, existe evidencia de que se encuentran soluciones buenas, en un tiempo competitivo con el resto de los algoritmos de optimización combinatoria.

### 2.1.1. Cromosomas o individuos

Se le llama cromosoma o individuo a la representación de solución a un problema, el cual está compuesto de uno o más genes, el individuo está conformado de información del problema a tratar. Los individuos pueden ser binarios (representados con 0 y 1) o no binarios (puede emplearse con letras o números diferentes a 0 y 1).

### 2.1.2. Población inicial

Se le llama así al conjunto de soluciones en donde los algoritmos evolutivos evolucionan, estos individuos no son estáticos, la evolución surge al pasar de generación en generación, aunque por ende entre mayor número otorga una mejor solución al problema.

### 2.1.3. Operadores genéticos

Holland, 1975, menciona que existen dos tipos de operadores genéticos: los primarios y los secundarios. Los primarios son aquellos operadores de los procesos evolutivos, dentro de estos operadores se encuentra selección y cruce, los secundarios son operadores opcionales en el proceso de la evolución. Existen diferentes técnicas para la selección de individuos para la reproducción y la reinserción. A continuación, se muestran dos técnicas:

#### 2.1.3.1. Operador de selección

Existen diferentes técnicas para la selección de individuos para la reproducción y la reinserción. A continuación, se muestran dos de las técnicas existentes:

- Selección por ruleta: Descrita por Hassoun(Solar Rodríguez & Pablo, 2015) como “la versión estocástica de la supervivencia del más apto”, el Muestreo Estocástico con Remplazo, o selección por ruleta, es la técnica de selección más usada. Esta técnica

consiste, en asignar un segmento de la ruleta a los individuos en base a la aptitud de éstos y la aptitud total de la población actual, y girar la ruleta tantas veces como selecciones se requieran.

- Selección por torneo: Seleccionar  $k$  miembros al azar y escoger el mejor (Gestal, 2010). Se elige un tamaño de torneo (valor para:  $k$ ), se crea números aleatorios, se comparan los valores y se selecciona el mejor, este proceso se repite hasta que se llene la población, y se muestra cual es el valor ganador.

### 2.1.3.2. Operador de cruce o recombinación

Como se muestra en la Figura 1, la cruce o recombinación combinan dos padres para crear nuevas soluciones (descendientes), que son posiblemente mejores. Existen muchas formas de conseguir esto, y un rendimiento competente del algoritmo depende de un mecanismo de recombinación bien diseñado, idealmente, la solución descendiente que se obtiene de la recombinación no será idéntica a ninguno de los padres, sino que contendrá información combinada de los dos (Soria, Pandolf, 2013).

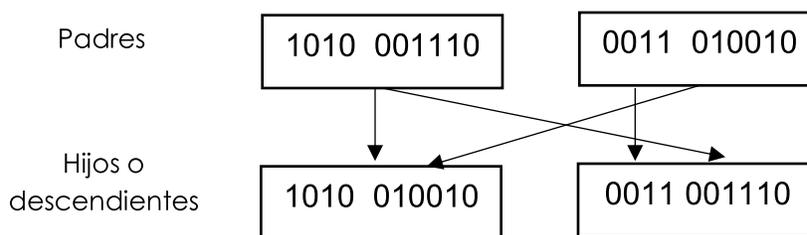


Figura 1. Ejemplo de cruce o recombinación en un algoritmo genético.

### 2.1.3.3. Operador de mutación

El operador de mutación brinda aleatoriedad a los individuos de una población. Al igual que con el operador de cruce, este operador cuenta con probabilidad de mutación (Gestal, 2010).

## 2.2. Optimización multiobjetivo

Casi todos los problemas reales involucran la optimización de varios objetivos que a menudo están en conflicto y compiten entre sí. En lugar de un óptimo simple, existen un conjunto de alternativas conocidas como soluciones óptimas de Pareto. Estas soluciones son óptimas en el amplio sentido de que ninguna otra solución es superior a éstas cuando todos los objetivos son considerados. Los problemas de optimización multiobjetivo son muy comunes en el área de la ingeniería y las técnicas para resolverlos muy diferentes a las empleadas en la optimización simple debido a que en la optimización multiobjetivo se tratan dos o más objetivos y en la optimización simple se trabaja sobre un solo objetivo (Sardi, 2011).

Calderón, 2016, menciona que la optimización multiobjetivo es un proceso complejo, debido a que regularmente las funciones objetivo que definen los problemas no están bien condicionadas o no cumplen con los requisitos mínimos para garantizar la convergencia

de algoritmos clásicos, como convexidad, continuidad y diferenciabilidad. Por lo que menciona que es por ello por lo que la literatura se enfoca en el estudio de técnicas de optimización para problemas definidos por funciones con características particulares, por ejemplo, que el costo de su evaluación sea elevado, no convexas o no diferenciables.

### 2.3. Métodos heurísticos

La palabra heurística procede del griego εὕρισκειν, que significa <<encontrar, descubrir>> (Ricci, 2013). Un método heurístico es un procedimiento para resolver problemas de optimización en donde la estructura del problema se utiliza de forma inteligente para obtener una buena solución (Zanakis & Evans, 1981; Cunqueiro, 2007).

Silver, 1980 menciona la siguiente clasificación de métodos de resolución mediante heurísticos:

- Métodos constructivos: Se caracterizan por construir una solución definiendo diferentes partes de ella en sucesivos pasos.
- Métodos de descomposición: Dividen el problema en varios más pequeños y la solución se obtiene a partir de la solución de cada uno de estos.
- Métodos de reducción: Tratan de identificar alguna característica de la solución que permita simplificar el tratamiento del problema.
- Métodos de manipulación del modelo: Obtienen una solución del problema original a partir de otra de otro problema simplificado (con menos restricciones, linealizando el problema, etc.)
- Métodos de búsqueda por entornos, en las que se parte de una solución inicial a la que se realizan modificaciones para obtener una solución final.

#### 2.3.1. Algoritmos evolutivos (AE)

De acuerdo con (Andaluz, 2004) el término AE, es empleado para describir la solución de problemas de optimización o búsquedas trabajan con individuos, en donde presentan posibles soluciones a un problema.

Un algoritmo evolutivo puede ser implementado de forma independiente al problema, lo cual hace que estos algoritmos sean robustos y flexibles, por ser para cualquier problema, sin embargo, no están especializados en ningún problema en concreto (Holland, 1975).

### 2.4. Recocido simulado

La metaheurística recocido simulado adopta su nombre gracias a la analogía con el proceso físico de recocido con sólidos. La principal característica de la metaheurística de recocido simulado es que proporciona una alternativa eficiente para escapar de óptimos locales al permitir movimientos “no tan buenos” respecto a la función objetivo (*hill-climbing moves*) se encuentra un óptimo global.

A medida que la temperatura se reduce a cero, los movimientos “no tan buenos” se producen con menos frecuencia convergiendo a óptimos globales. El procedimiento de recocido simulado comienza desde una solución aleatoria inicial. En cada iteración, una nueva solución es tomada del vecindario predefinido de la actual solución. El valor de la función objetivo de la nueva solución es comparado con la actual para determinar si un mejoramiento ha sido alcanzado(Escobar, 2012).

(Hoyos, 2007)menciona que el recocido simulado es una metodología de optimización basado en procesos termodinámicos (recocido). La idea es que a partir de puntos existentes se crean vecinos (versiones perturbadas de los puntos), a los cuales se prueba su función de costo o *fitness* y si esta es de mejor calidad que la del punto existente, el vecino entra a reemplazar al punto existente. Por medio del enfriamiento se puede lograr que vecinos de mala calidad que se pueden presentar en alto porcentaje en la parte inicial del proceso, se vayan eliminando y se reduzca la probabilidad de su aparición.

Los algoritmos de recocido se ejecutan donde su ciclo finaliza cuando se cumple alguna meta deseada, esto no se puede aplicar a los sistemas de tiempo real, donde el tiempo limitado impide que la estrategia de control estén indefinidos(Cruz Trejos, 2006).

De acuerdo con el autor Hernández, 2011, se muestra a continuación el algoritmo de recocido simulado:

1. Repetir
2. Repetir (para cierto  $c$ )
3. Seleccionar aleatoriamente  $x'$  en vecindad  $V(x)$ .
4. Sea  $\Delta = f(x') - f(x)$
5. Si  $\Delta < 0$  entonces  $x = x'$
6. En otro caso generar aleatoriamente  $I \in U(0,1)$
7. Si  $I < \exp(\Delta/c)$  entonces  $x = x'$
8. Hasta llegar a equilibrio.
9. Actualizar temperatura  $c$ .
10. Hasta condición de parada.

### 2.5. Resumen del capítulo

En este capítulo se han mencionado los conceptos básicos que son empleados en esta investigación. Se comenzó hablando de algoritmos genéticos como parte del procesamiento del lenguaje natural y recocido simulado, que corresponde a la investigación del presente trabajo.



## CAPÍTULO 3.

# Estado del Arte

---

En este capítulo se presentan los trabajos relacionados con asignación de cargas horarias personalizadas para estudiantes, algoritmos genéticos y recocido simulado, mismos que son empleados como punto de referencia para el presente trabajo.

### *3.1. Aplicación de recocido simulado para la generación automática de horarios*

En el trabajo titulado "Educational Timetabling" (Segura, 2005), resuelto con Recocido Simulado y modelado bajo una arquitectura Web", en este trabajo se implementó una mejora al algoritmo RS con el fin de disminuir el tiempo de ejecución, realizando la sintonización de los parámetros de manera analítica. Las restricciones de este problema se mencionan a continuación:

- Ningún estudiante asiste a más de un evento a la vez.
- El salón es lo suficientemente grande para todos los estudiantes asistentes y satisface todas las características requeridas por el evento.
- Solo hay un evento en cada salón, por cada período de tiempo.
- Un estudiante no debe tener un evento programado en el último período del día.
- Un estudiante no debe tener más de dos clases consecutivas.
- Un estudiante no debe tener una única clase en un día.

Los resultados obtenidos se comparan con los publicados por el PATAT (*Practice and Theory of Automated Timetabling*), una organización internacional formada por un grupo de investigadores reconocidos.

### *3.2. Algoritmos meméticos (con aplicación a Institución)*

En el trabajo titulado "Programación óptima de horarios de clase usando un algoritmo memético" (Granada, Toro, 2006), el principal objetivo es elegir un salón en un lapso para cada clase de forma que maximicen las preferencias de los estudiantes sin crear problemáticas en la programación de estudiantes o salones. La técnica que se emplea en este problema se basa en la modificación y adecuación del algoritmo genético por Chu-Beasley. Dentro de las restricciones para la solución a este trabajo son:

- Un estudiante no tenga clase en el último bloque del día.
- Un estudiante no reciba 2 eventos consecutivos.
- Un estudiante no reciba solo una clase al día.
- Los estudiantes no pueden tomar más de una unidad de aprendizaje al mismo tiempo.
- La capacidad del salón debe ser limitada.
- Solo una unidad de aprendizaje es asignada en un salón.

El método heurístico propuesto proporciona resultados interesantes en la solución del problema de optimización de horarios en cuanto a la calidad de la respuesta y velocidad de convergencia a la asignación de horarios de clase con un algoritmo memético.

### *3.3. Programación entera (con aplicación a Institución)*

De acuerdo con el autor Hernández, 2008, en su trabajo titulado "Programación de horarios de clases y asignación de salas en la Facultad de Ingeniería de la Universidad Diego Portales" (Hernández, 2008), la problemática es que cada semestre en la Universidad Diego Portales se tiene en promedio 150 cursos los cuales tienen un número variable de secciones.

Los cursos tienen dos tipos de clases: cátedra y auxiliares. La generación de carga horaria se trata de un problema de minimización. Dentro de las restricciones tomadas en cuenta para este trabajo fueron:

- Cada curso debe ser asignado a una sala de clase con capacidad de alumnos para dicho curso
- En una sala de clase, en un mismo día y bloque horario, se puede realizar lo más una clase.
- Un profesor no puede dictar más de una clase a la vez.
- Se deben respetar los horarios disponibles de los profesores.
- Las clases auxiliares deben realizarse de preferencia los miércoles en cualquier bloque horario. De no ser posible esta asignación, se pueden realizar en cualquier día y bloque horario.
- Se debe evitar, en lo posible, asignar cursos al Auditorio.

La herramienta asegura que los horarios generados cumplen con las restricciones impuestas y entrega la flexibilidad de poder realizar asignaciones horarias parciales, se asignan cursos a salas con la capacidad disponible por cada sala, se evitan los traslapes de clases en las salas. Se evita asignar cursos en el auditorio.

### *3.4. Búsqueda tabú (con aplicación a Institución)*

El algoritmo tabú es implementado en el trabajo de Martínez, 2012, en su investigación que lleva el nombre de "Algoritmo basado en búsqueda tabú para el problema de asignación de Horarios de Clases" (Martínez, 2012), consiste en asignar lo mejor posible las clases para los estudiantes en bloques de tiempo semanales y en determinadas aulas, bajo ciertas restricciones. El problema de asignación de horarios, de acuerdo con el autor puede verse como optimización combinatoria, ya que se requiere hacer la mejor asignación entre todas las posibles asignaciones de horarios. Las restricciones empleadas en este trabajo son:

- Dos secciones no deben estar asignadas al mismo tiempo en la misma aula.
- Dos aulas no deben estar asignadas al mismo tiempo con la misma sección.
- Dos secciones de una sección compuesta no deben estar asignadas al mismo tiempo.
- Dos secciones con el mismo profesor no deben estar asignadas al mismo tiempo.
- Una sección no debe estar asignada en horas en las cuales su profesor no está disponible.
- Un aula no debe ser asignada en horas en las cuales no está disponible.
- La cantidad de estudiantes de una sección debe ser menor o igual a la capacidad de cualquier aula asignada a esa sección.
- Dos secciones cuyas asignaturas son de años consecutivos no deben estar asignadas al mismo tiempo.
- Los bloques de las secciones de una sección compuesta deben estar distribuidos lo mejor posible sobre los días disponibles.

- Los bloques de las secciones de una sección compuesta deben estar asignados en un turno del día: mañana o tarde.
- No se deben asignar secciones a la hora de almuerzo.

Con la implementación de este algoritmo se encuentran mejores soluciones que las obtenidas con el método manual, en un tiempo de cómputo razonable.

### *3.5. Ritmos cognitivos (con aplicación a Profesores)*

En el trabajo titulado “*Programación de horarios escolares, mediante el algoritmo genético NSGA-II*” (Suárez, 2013), se trata como multiobjetivo con elitismo, se plantea la solución del problema de programación óptima de horarios escolares de una escuela pública colombiana. Para este trabajo tomaron en cuenta las siguientes restricciones:

- Asignaturas donde los estudiantes tienen una mejor disposición para recibir su contenido en la primera y segunda hora de jornada.
- Asignaturas donde los estudiantes tienen una mejor disposición para recibir su contenido en la tercera y cuarta hora de jornada.
- En un mismo día no se deben dictar más de dos horas de la misma materia.
- Un profesor no puede dictar 2 materias a la misma hora del día.
- No se pueden programar dos clases en el mismo salón, a una misma hora del día.

El algoritmo permitió 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.

En la Tabla 1 se muestran algunas de las herramientas que ayudan a resolver el problema de asignación de cargas horarias para estudiantes mediante técnicas computacionales, en donde se aprecia que la aplicación es para instituciones y profesores, pero no se encontraron trabajos que tomen en cuenta las condiciones de los estudiantes.

**Tabla 1.** Trabajos relacionados con la asignación de cargas horarias.

<b>Investigación</b>	<b>Técnica empleada</b>	<b>Aplicación</b>
Educational Timetabling resuelto con Recocido Simulado y modelado bajo una arquitectura Web. Generalización y aplicación práctica: casos PATAT y UADY (González, 2005)	Recocido simulado	Institución
Programación óptima de horarios de clase usando un algoritmo memético (Granada, 2006).	Algoritmo memético	Institución
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 (Hernández, 2008).	Programación entera	Institución
Algoritmo basado en búsqueda tabú para el problema de asignación de Horarios de Clases (Martínez, 2012).	Búsqueda Tabú	Institución
Programación de horarios escolares basados en Ritmos Cognitivos usando un Algoritmo Genético con ordenamiento no dominante, NSGA-II (Suárez, 2013).	Ritmos Cognitivos	Profesores

### *3.6. Resumen del capítulo*

En este capítulo se mencionan los trabajos relacionados al problema de horarios de cursos personales para estudiantes universitarios basados en algoritmos genéticos y recocido simulado. En este capítulo se muestra que la mayoría de los trabajos hacen referencia a generar cargas horarias para instituciones y docentes. Sin embargo, no se emplea para la generación de cargas horarias para estudiantes a través de condiciones que satisfagan al estudiante de manera académica y personal.

Se mencionan trabajos como lo es el trabajo de Segura, 2005, que trata sobre el *timetabling* resuelto con recocido simulado, con aplicación en Institución(Segura, 2005).

Así como el trabajo de Granada en 2006, en el cual trabajó la programación óptima de horarios de clase usando un algoritmo memético(Granada, 2006).

El trabajo dentro del área investigada realizado por Suárez, 2013(Suárez, 2013), titulado "*Programación de horarios escolares basados en Ritmos Cognitivos usando un Algoritmo Genético con ordenamiento no dominante, NSGA-II*"



## CAPÍTULO 4.

# Método Propuesto

---

Si se emplean algoritmos genéticos y recocido simulado para la generación de horarios de cursos personales para estudiantes universitarios basados en algoritmos genéticos y recocido simulado, entonces se satisfacen las necesidades del estudiante.

### 4.1. Descripción del método propuesto

Se proponen dos métodos de optimización para poder generar diferentes cargas horarias que satisfagan las condiciones del estudiante, estos métodos son propuestos debido a que son heurísticas que se encargan de la búsqueda y optimización.

El problema de la presente tesis tiene varias condiciones, por lo tanto, se plantea como un problema de optimización multiobjetivo, es por esto que se propone utilizar un algoritmo genético y recocido simulado para poder valorar los resultados con estas dos técnicas computacionales.

Datos de entrada:

- Los horarios con las unidades de aprendizaje
- Unidades de aprendizaje a cursar
- Unidades de aprendizaje que puede y debe cursar
- Condiciones del estudiante

#### 4.1.1. Descripción del plan de estudios

La carrera de ingeniería de software cuenta con 55 cursos distribuidos en nueve semestres, pero el número de semestres es flexible. Los cursos establecidos sugeridos por semestre se muestran en la Figura 2.

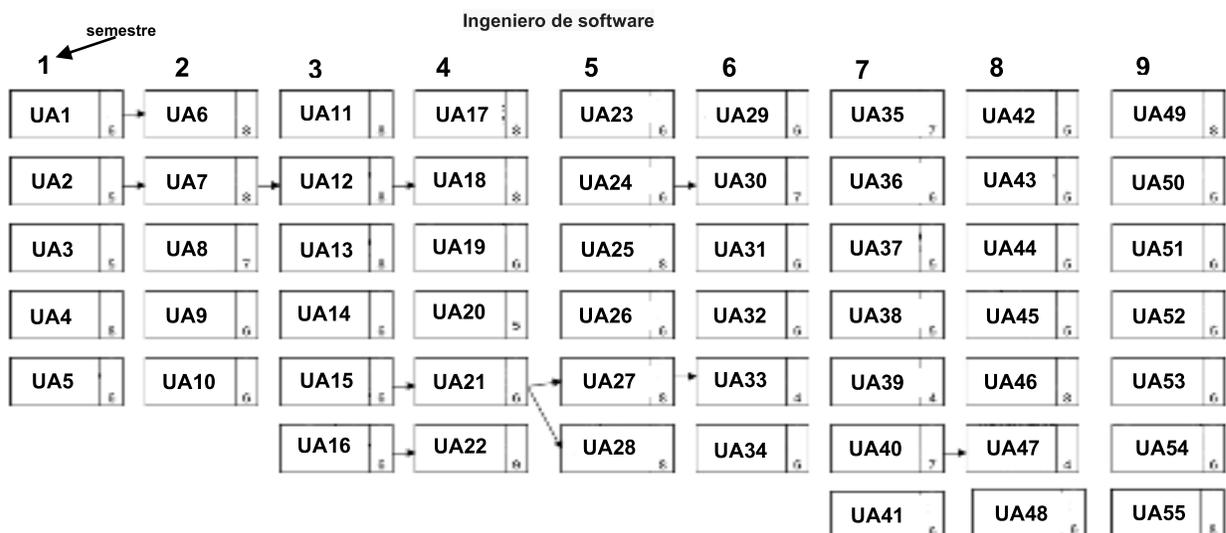


Figura 2. Mapa curricular de ingeniería de software.

### *4.1.2. Restricciones*

Para el presente trabajo se toman en cuenta condiciones del estudiante y de la Institución a la que pertenece, esto con la finalidad de tomar en cuenta lo que un estudiante necesita para la asignación de horarios personales. Gracias a una encuesta realizada se pudieron detectar y representar condiciones que son útiles para la conformación de la función de aptitud del algoritmo genético, mismas que son divididas en condiciones primarias y descripción formal respectivamente, a continuación, se definen cada una de ellas en lenguaje natural.

### *4.1.3. Las condiciones primarias*

- Dos unidades de aprendizaje no pueden ser seleccionadas en el mismo día, ni con el mismo horario.
- La institución brinda los horarios de clases.
- El estudiante debe elegir cuantas horas semanales de clase puede asistir a la Institución

### *4.1.4. Las condiciones secundarias*

- Dos secciones cuyas asignaturas son de semestres consecutivos no deben ser asignadas al mismo tiempo.
- No se pueden asignar unidades de aprendizaje seriadas (es decir si el estudiante no a cursado la primera unidad de aprendizaje de la línea de seriación no puede cursar la siguiente sin haber aprobado la primera).
- Los bloques deben ser asignados de acuerdo con el horario establecido por la institución.

### *4.1.5. Descripción formal*

La función de aptitud busca minimizar los errores de acuerdo con el horario propuesto por la Institución. El error calculado se determina por la suma de los errores en cada uno de los criterios. Enseguida se muestran las definiciones de los criterios empleados para este trabajo.

I. Condiciones de Tiempo por Semana (CTS): La condición de tiempo disponible por semana varía de acuerdo con cada estudiante, por ejemplo: existen estudiantes que trabajan o tienen labores externas a la institución, por lo cual se implementa esta condición.

II. Condición de Horas Traslapadas (CHT): Un estudiante no puede tomar dos unidades de aprendizaje al mismo tiempo (horas traslapadas). Un claro ejemplo de esto es un estudiante que selecciona dos unidades de aprendizaje en el mismo día y la misma hora, lo que provoca un serio problema, porque no puede estar en ninguna de las dos clases al mismo tiempo, lo que repercute en el rendimiento escolar del estudiante.

III. Condición de Unidades de Aprendizaje que Puede y Debe Cursar el estudiante (CUAPDC): Se toma en cuenta el avance curricular de cada estudiante para poderle sugerir posibles cargas horarias a seleccionar.

IV. Condición de Sugerencia de Cargas Horarias en Tiempo Real (CSCHTR): El tiempo real es un factor de suma importancia, ya que el estudiante en el momento que desee una sugerencia de carga horaria, podrá obtenerla (Fernández, 2005). Esta condición cobra relevancia minutos antes de la inscripción del estudiante debido a que las unidades de aprendizaje disponibles cambian el horario, cancelación o apertura adicional, etc.

V. Condición de Créditos Máximos (CCMAX) y Mínimos para cada semestre (CCMIN): El estudiante tiene un mínimo y máximo de créditos que cumplir, la mayoría de las instituciones tienen un mínimo y máximo de créditos por semestre, es decir el horario propuesto por el estudiante no puede excederse del límite de créditos, así como tampoco le pueden faltar créditos.

VII. Unidades de aprendizaje que se pueden adelantar (CUAA): Unidades de aprendizaje que el estudiante puede adelantar, dependiendo el curso que se encuentre, si es alumno regular y el tutor avala el que el estudiante pueda adelantar unidades de aprendizaje.

#### *4.1.6. Función de aptitud y condiciones del estudiante*

La función de aptitud está integrada de la siguiente manera:

$$F(t) = \alpha CTS(t) + \beta CHT(t) + \gamma CUAPDC(t) + \psi CCMAX(t) + \phi CCMIN(t) + \Omega CUAA(t) .$$

Las condiciones son:

##### **I. CTS**

El primer error calcula el Número de Horas del individuo (NH) que se exceden de acuerdo con las Horas Máximas Disponibles por Semana (HMS) que tiene el estudiante:

$$CTS(indi) = NH(indi) = \frac{\begin{cases} NH(indi) > HMS & NH(indi) - HMS \\ \text{En otro caso} & 0 \end{cases}}{HMS}$$

##### **II. CHT**

Para la condición de Horas Traslapadas se modela el error como el Número de las Horas Traslapadas propuestas por el individuo. Este error se realiza con HMS (Horas Máximas Disponibles por Semana), es decir:

$$CHT = \frac{NHT}{HMS}$$

##### **III. CUAPDC**

Para la condición de Unidades de Aprendizaje que Puede y Debe Cursar el estudiante, evalúa como error la asignación de unidades de aprendizaje de semestres más avanzados, propuesto por el individuo.

Se deben seleccionar UA de los primeros semestres, es decir:

$$\sum_{i=1}^n \text{Semestre } indi(UAi)$$

Para normalizar este valor se calcula como peso menor la sumatoria de las  $n$  unidades de aprendizaje de menores semestres que debe tomar y como peso mayor la sumatoria de las unidades de aprendizaje de mayores semestres que puede tomar

$$CUAPDC = \frac{\sum_{i=1}^n \text{semestre indi } (UAi) - \sum_{i=1}^n \text{semestre } (LUAP(i))}{\sum_{i=1}^n \text{semestre}(LUAP(i))}$$

#### IV. CCMAX, CCMIN

El error de esta condición evalúa qué tan alejado está el número de créditos propuesto por el individuo, de los rangos permitidos por CCMAX y CCMIN.

$$CCMIN, CCMAX = \begin{cases} NC(indi) > CCMAX & \frac{NC - CCMAX}{CCMAX} \\ NC(indi) < CCMIN & \frac{CCMIN - NC}{CCMIN} \\ CCMIN \leq NC(indi) \leq CCMAX & 0 \end{cases}$$

#### V. CUAA

Para la condición de unidades de aprendizaje que el estudiante puede adelantar, dónde se toman en cuenta los Créditos de las Unidades de Aprendizaje a Asignar (CREDUA), entre el número de Créditos Máximos del Curso(CCMAX):

$$CUAA = \frac{CREDUA}{CCMAX}$$

### 4.2. Algoritmo genético

Los algoritmos genéticos pretenden resolver problemas mediante la optimización. De acuerdo con el autor (Gestal, 2010), los algoritmos genéticos son métodos los cuales pueden adaptarse a la búsqueda y optimización de problemas, basados en el proceso genético de los organismos vivos.

#### 4.2.1. Población

La población se encuentra compuesta por un conjunto de individuos que en este caso son los correos electrónicos empleados.

#### 4.2.2. Individuo

Cada individuo de la población representa una solución válida del problema. La información que contiene el individuo es almacenada en su cromosoma y cada una de las unidades que compone el cromosoma se llama gen, que normalmente es en unidades binarias.

### 4.2.3. Función de evaluación

Hace referencia al cálculo numérico o puntuación que se asocia a cada individuo e indica su calidad. Por ejemplo, en este trabajo se emplea la precisión que se consigue al clasificar un conjunto de datos.

### 4.2.4. Operadores evolutivos

Son los operadores de selección, cruce y mutación y sirven para mezclar los cromosomas de dos individuos distintos (padres), produciendo normalmente otros dos individuos distintos (hijos) y la mutación altera los genes de un cromosoma.

En este caso se ocupa: selección por torneo, en donde indica que se elige  $k$  aleatoriamente de esto se selecciona el más alto y entonces es el valor que pasa a la siguiente generación, se emplea el operador de cruce en  $n$  puntos, este se obtiene seleccionando la cantidad de  $n$  veces para los cruces que se hagan, se cruzan en un punto dado aleatoriamente, y la mutación es por inversión binaria, por ser un algoritmo con codificación binaria, una vez obtenido el gen a mutar se reemplazará el gen mutado por el valor contrario.

Y también se toma en cuenta el criterio de parada, mismo que puede ser detenido al alcanzar el número de generaciones deseado o tiempo de ejecución específico.

En la Figura 3 se muestra la estructura del algoritmo genético para poder generar cargas horarias para estudiantes.

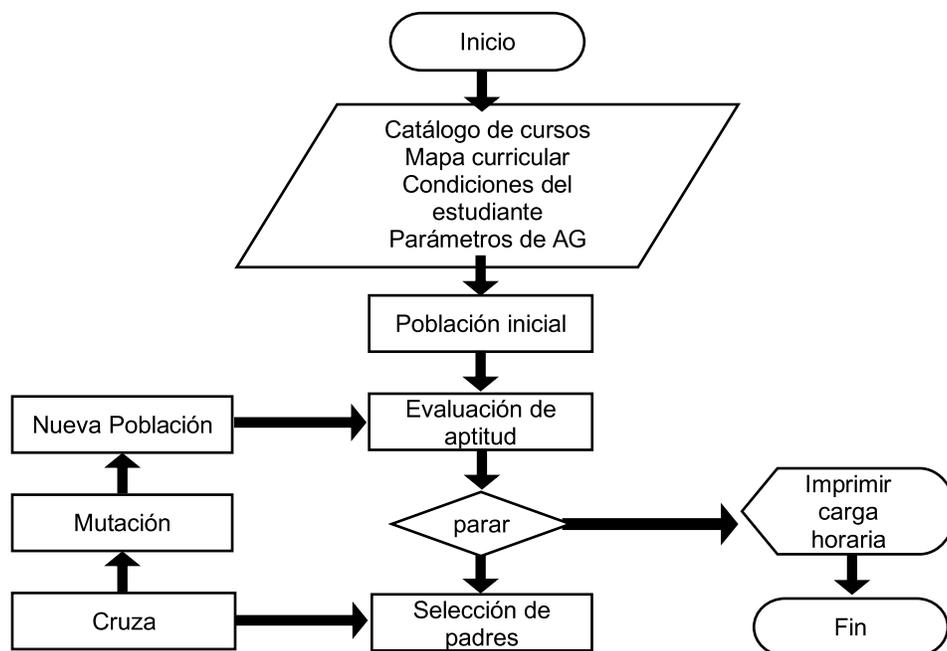


Figura 3. Método propuesto.

### 4.3. Recocido simulado

La solución al problema se planteó como un vector de  $n$  dimensiones, una por cada carga horaria a generar. Cada una de estas localidades contiene un valor entero que representa el tipo de horario programado a determinado estudiante. También existe para cada localidad del vector, un arreglo de horarios posibles a asignar al estudiante por lo que se tiene delimitados los valores que puede tomar.

Los parámetros característicos de este algoritmo son la temperatura inicial y la función de enfriamiento, estos determinan la probabilidad de salto en el espacio de búsqueda en un momento dado como se muestra enseguida:

El recocido simulado se trata de un problema multiobjetivo, por lo cual se evalúan las siguientes condiciones que fueron previamente normalizadas en un rango de 0 a 1, como se menciona anteriormente.

Enseguida se muestra la función de evaluación para recocido simulado:

$$P(T) = e^{\frac{FE(va) - FE(vn)}{T}}$$

Dónde:  $FE$  es la función de evaluación,  $va$  es el estado actual y  $vn$  es el nuevo estado con un costo mayor.  $T$  es la temperatura que está en función del tiempo.

#### 4.3.1. Temperatura inicial

El parámetro temperatura es determinante para saber qué tan probable es aceptar una solución peor a la que ya se tiene, ésta debe ser alta al inicio del algoritmo con la finalidad de escapar a mínimos locales, pero se debe ir reduciendo después de algunas iteraciones. En este caso se encontrará el valor de ésta haciendo correr el algoritmo con  $P(T) = 0$  (suponiendo  $T = 0$ ) y analizando el comportamiento de éste se propondrá un valor de  $T$  que haga que la probabilidad sea un poco menor a la probabilidad de mejorar la solución.

#### 4.3.2. Función de enfriamiento

Se usará la función de enfriamiento geométrico, ésta es la más usada en la práctica, se define como:  $T_{n+1} = T_n \alpha$ , donde  $\alpha$  es un valor cercano a 1 y se calculará de acuerdo con la velocidad con que converja la solución al probar el algoritmo con  $P(T)$  igual a cero (probabilidad de aceptación a un estado peor).

### *4.4. Resumen*

En este capítulo se describió el método propuesto que pretende realizar la asignación de cargas horarias para estudiantes con un algoritmo genético y recocido simulado.

Se describen las fases de las que está compuesto el método propuesto, desde la definición general de las técnicas a implementar como los pasos que se siguieron para el algoritmo genético (tomando en cuenta los parámetros de cruce y mutación), las condiciones del estudiante, así como los pasos del método propuesto para recocido simulado (temperatura y la función de enfriamiento), estas técnicas (algoritmos genéticos y recocido simulado) son empleadas para la asignación "Horarios de cursos personales para estudiantes universitarios basados en algoritmos genéticos y recocido simulado", ambas heurísticas elegidas para la búsqueda y optimización.

Parte del método propuesto para cumplir el siguiente capítulo (Experimentos) con la hipótesis realizada en el Capítulo 1.



# CAPÍTULO 5.

## Experimentación

---

En este capítulo se describe la experimentación realizada bajo el método propuesto descrito en el capítulo anterior, así como los resultados generados por los mismos.

Con la finalidad de comprobar la hipótesis propuesta se presenta la aplicación del método en la asignación de cargas horarias para estudiantes mediante algoritmos genéticos y recocido simulado.

### 5.1. Descripción de los datos utilizados

En la Figura 4 se muestran los horarios brindados por la institución, los horarios son realizados para estudiantes con una situación académica regular. Estos forman parte de la descripción de los datos utilizados, en conjunto con las condiciones del estudiante para realizar la asignación de carga horario.

Hora/día	L	M	M	J	V	S
7:00 - 8:00						
8:00 - 9:00		UA51			UA50	
9:00 - 10:00						
10:00 - 11:00	UA52					UA54
11:00 - 12:00			UA49	UA53	UA55	
12:00 - 13:00						
13:00 - 14:00						

Hora/día	L	M	M	J	V	S
7:00 - 8:00				UA47		
8:00 - 9:00	UA45	UA46				
9:00 - 10:00				UA44	UA48	
10:00 - 11:00	UA42					
11:00 - 12:00						
12:00 - 13:00		UA43				
13:00 - 14:00						

Hora/día	L	M	M	J	V	S
7:00 - 8:00						
8:00 - 9:00	UA50		UA52		UA55	
9:00 - 10:00						
10:00 - 11:00	UA49					
11:00 - 12:00		UA51		UA53		
12:00 - 13:00			UA54			
13:00 - 14:00						

Hora/día	L	M	M	J	V	S
7:00 - 8:00	UA44					
8:00 - 9:00		UA45				UA42
9:00 - 10:00						
10:00 - 11:00	UA46					
11:00 - 12:00		UA48				
12:00 - 13:00		UA47				UA43
13:00 - 14:00						

Figura 4. Catálogo de cursos que brinda la institución académica.

#### 5.1.1. Resultados de condiciones de la función de aptitud

Para la recolección de datos sobre ACH de los estudiantes, en primera instancia se aplicó un examen para identificar las condiciones que los estudiantes toman en cuenta para realizar ACH; como segunda instancia se solicitó considerar el catálogo de cursos publicado y como tercera instancia realizar la ACH.

La Tabla 2 se muestran las condiciones de los estudiantes y la evaluación de ACH por la función de aptitud, en dónde se muestra la evaluación de la cara horaria generada manualmente, para poder identificar si la generada automáticamente es más satisfactoria para las condiciones del estudiante.

**Tabla 2.** Condiciones de los estudiantes para generar cargas horarias manuales para 25 estudiantes evaluados con la función de aptitud propuesta, mismos que son obtenidos de la UAPT, dónde se solicitó a 25 estudiantes de la licenciatura, brindar sus condiciones al momento de realizar cargas horarias, por lo cual es un primer experimento de asignación de cargas horarias personales.

Estudiantes	Condiciones del estudiante						Función de aptitud
	CTS	CHT	CUAPDC	CCMAX	CCMIN	UAA	Carga horaria generada manualmente
E1	20	0	6	42	17	0	0
E2	15	0	5	42	17	1	0
E3	14	1	6	35	17	0	0
E4	15	2	5	41	17	1	0
E5	16	1	6	42	17	1	0
E6	18	0	6	36	17	0	0.07
E7	20	0	6	42	17	0	0.08
E8	17	1	6	41	17	0	0.02
E9	16	1	6	38	17	0	0.08
E10	20	0	6	42	17	0	0.02
E11	16	1	6	42	17	0	0.08
E12	14	0	6	35	17	0	0.06
E13	20	0	6	34	17	1	0.06
E14	25	0	5	42	17	0	0.08
E15	24	1	5	37	17	0	0.07
E16	22	0	6	42	17	0	0.8
E17	17	1	6	36	17	0	0.02
E18	14	0	5	38	17	1	0.08
E19	20	0	6	42	17	0	0.03
E20	15	0	6	42	17	1	0.06
E21	18	1	6	42	17	0	0.08
E22	20	0	6	38	17	0	0.07
E23	17	1	5	38	17	0	0.08
E24	18	1	5	42	17	0	0.08
E25	20	0	6	38	17	0	0.07

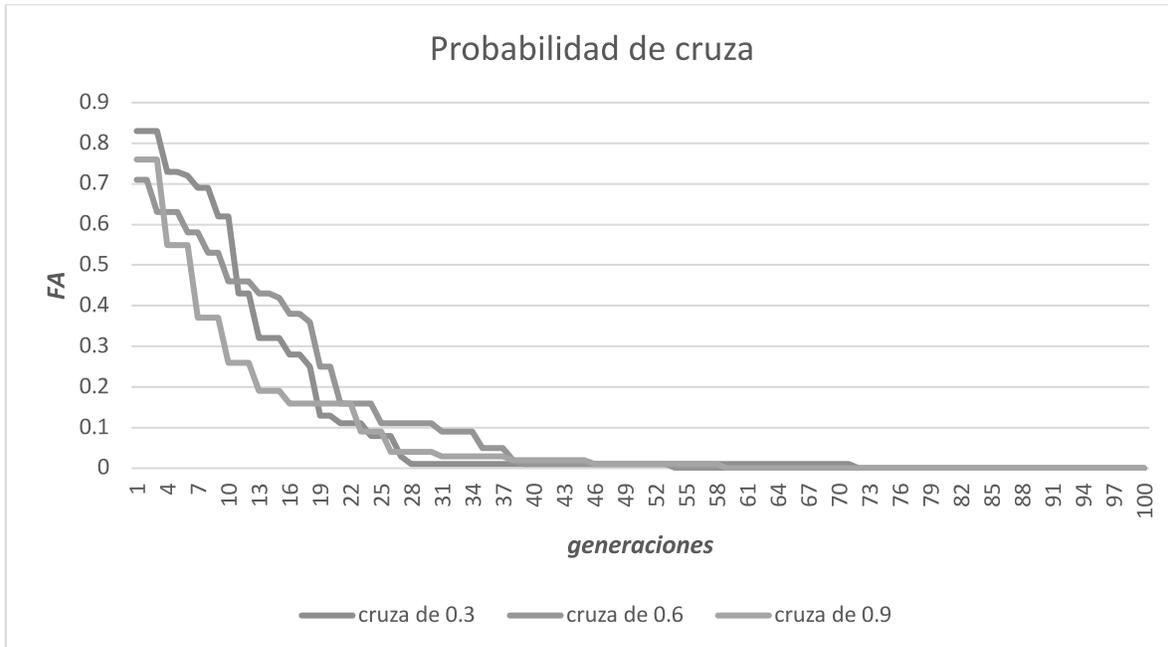
## 5.2. Experimentos con Algoritmos genéticos

Los siguientes experimentos tienen como objetivo mostrar el desempeño evolutivo del algoritmo genético propuesto, para obtener un buen desempeño con un conjunto de parámetros predefinidos. En todos los experimentos se utilizan los siguientes parámetros AG:

- Selección de torneo con  $k=2$ , cruce en 2 puntos, tamaño de la población de 100 individuos y el número máximo de generación es de 100.

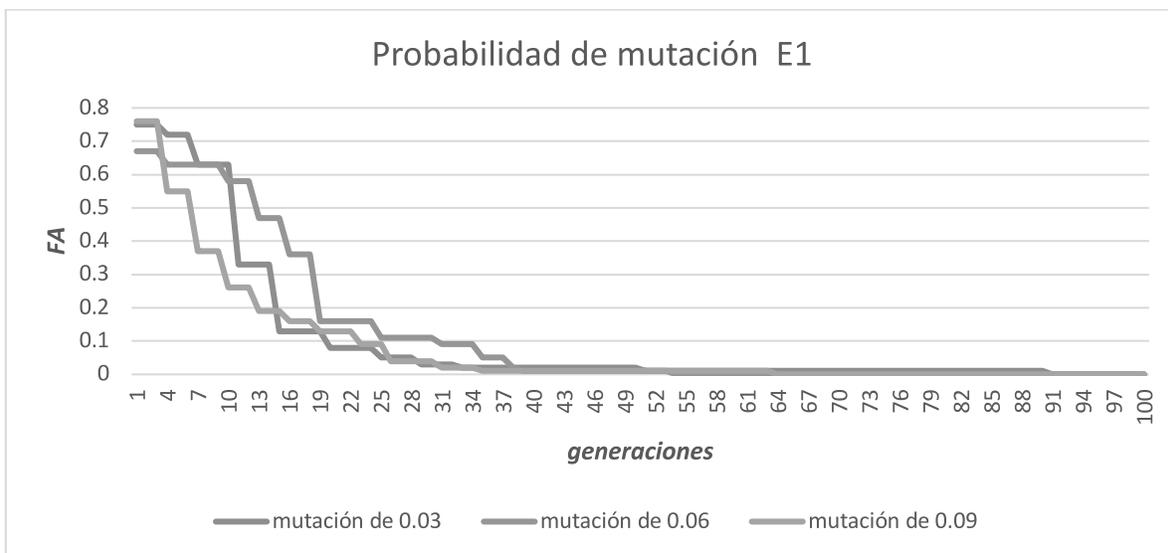
Para los siguientes experimentos, las preferencias y condiciones de los estudiantes E1, E2 y E3. En todos los experimentos, las variables  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\varepsilon$ ,  $\lambda$ ,  $\psi$ ,  $\theta$  y  $\Omega$  reciben el mismo peso (0.125).

La Figura 5 se muestra la aptitud obtenida con respecto al paso de generaciones aplicando las siguientes probabilidades de mutación: 0.3, 0.6 y 0.9, para el Estudiante 1.



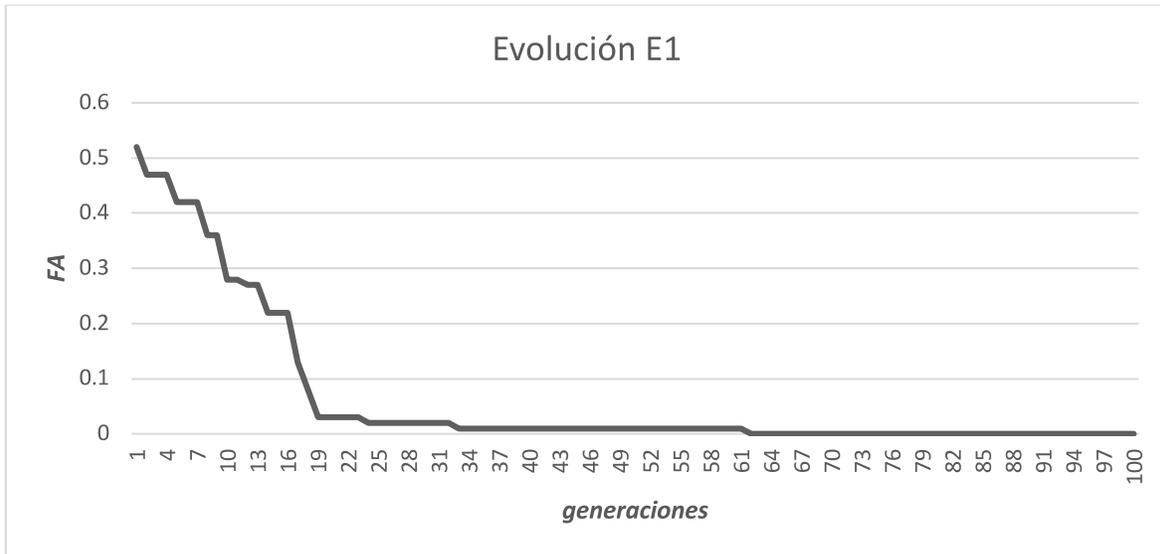
**Figura 5.** Probabilidad de cruce para el Estudiante 1.

El segundo parámetro es para determinar la probabilidad de mutación. Como se puede observar en todos los casos, el método propuesto puede generar una carga horaria adecuada para el estudiante E1. Sin embargo, la probabilidad de mutación de 0.03 parece más apropiada ya que tiene una mejor convergencia.



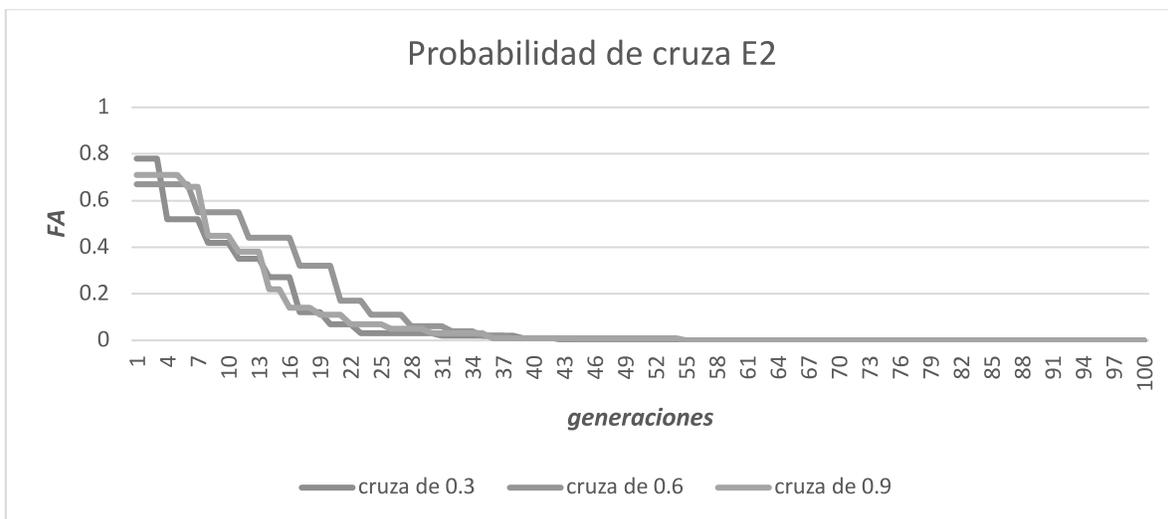
**Figura 6.** Probabilidad de mutación para el Estudiante 1.

La Figura 7 se muestra el rendimiento del AG propuesto, utilizando la probabilidad de cruce de 0.3 y la probabilidad de mutación de 0,03. Para comprobar que el comportamiento no depende de las condiciones del alumno Estudiante 1, se repitieron los mismos parámetros del esquema en los alumnos Estudiante 2 y Estudiante 3.



**Figura 7.** Evolución del AG para el Estudiante 1.

En la Figura 8 la probabilidad de cruce oscila entre 0,3 y 0,9, en los tres casos. El método propuesto puede generar una carga horaria adecuada para el estudiante E2; pero la probabilidad de cruce de 0.3 parece adecuada para obtener un mejor resultado.



**Figura 8.** Probabilidad de cruce para el Estudiante 2.

En la Figura 9 la probabilidad de mutación que se probó es 0.03, 0.06 y 0.09. Como se puede observar, en todos los casos, el método propuesto puede generar una carga horaria adecuada para el estudiante E2. sin embargo, la probabilidad de mutación de 0.03 parece más apropiada.

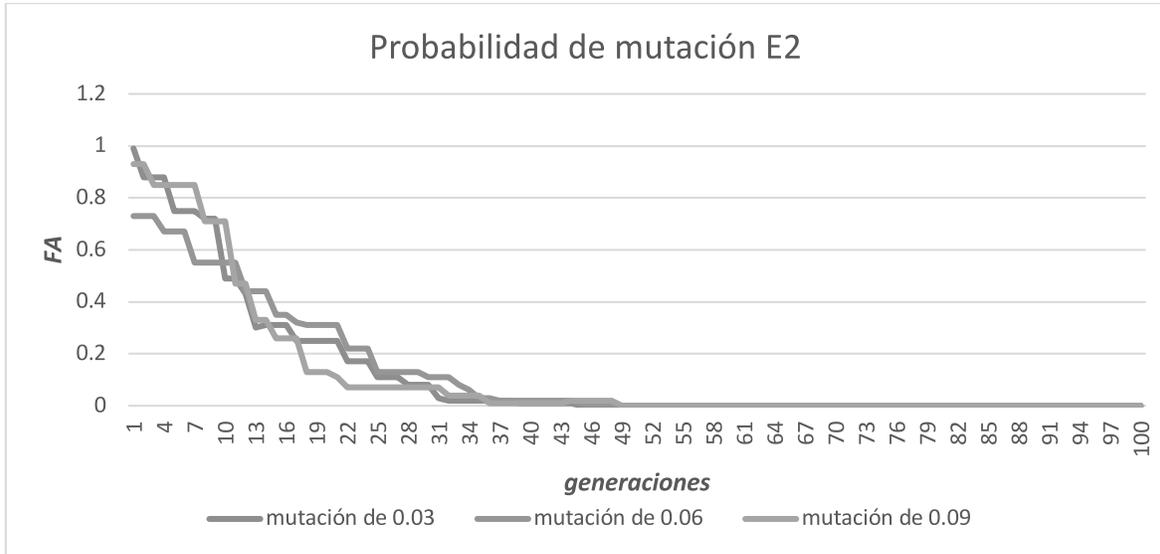


Figura 9. Probabilidad de mutación para el Estudiante 2.

La Figura 10 muestra el rendimiento del AG propuesto utilizando la probabilidad de cruce de 0,3 y la probabilidad de mutación de 0,03. Para observar si este comportamiento depende de las condiciones del alumno E2, se repitieron los mismos parámetros del esquema en el alumno E3 en dónde ponemos a prueba los parámetros con diferentes condiciones y situaciones de los estudiantes. Los gráficos 11 y 12 muestran el desempeño del AG propuesto utilizando la probabilidad de cruce de 0.3 y la probabilidad de mutación de 0.03.

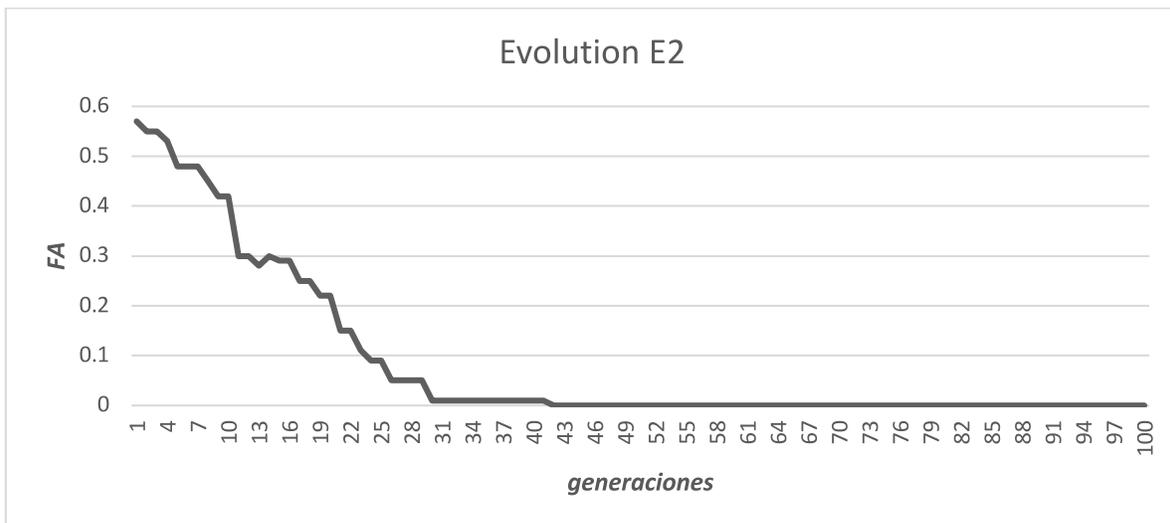


Figura 10. Evolución de AG para el Estudiante 2.

La Figura 11 muestra probabilidad de cruce de 0,3, 0.6 Y 0.9. Como se puede observar, en todos los casos, el método propuesto puede generar una carga horaria adecuada para el estudiante E3. Sin embargo, la probabilidad de cruce de 0.09 parece mejor debido a la convergencia que se alcanza con menos generaciones a diferencia de los demás parámetros.

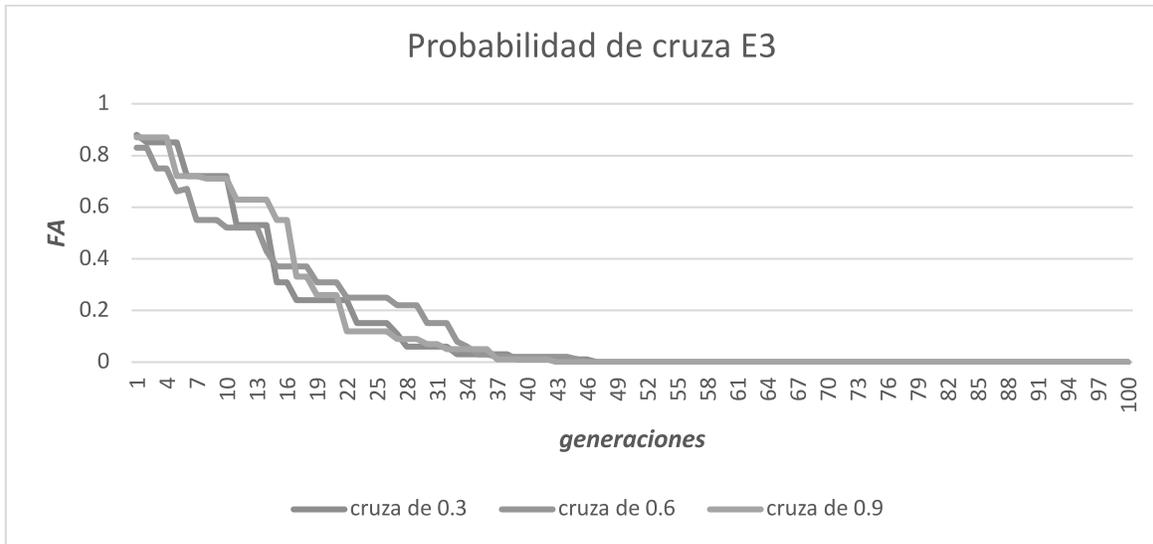


Figura 11. Probabilidad de cruce para el Estudiante 3.

La Figura 12 muestra la probabilidad de mutación de 0.03, 0.06 Y 0.09. Como se puede observar, en todos los casos, el método propuesto puede generar una carga horaria adecuada para el estudiante E3. Sin embargo, la probabilidad de mutación de 0.09 es mejor.

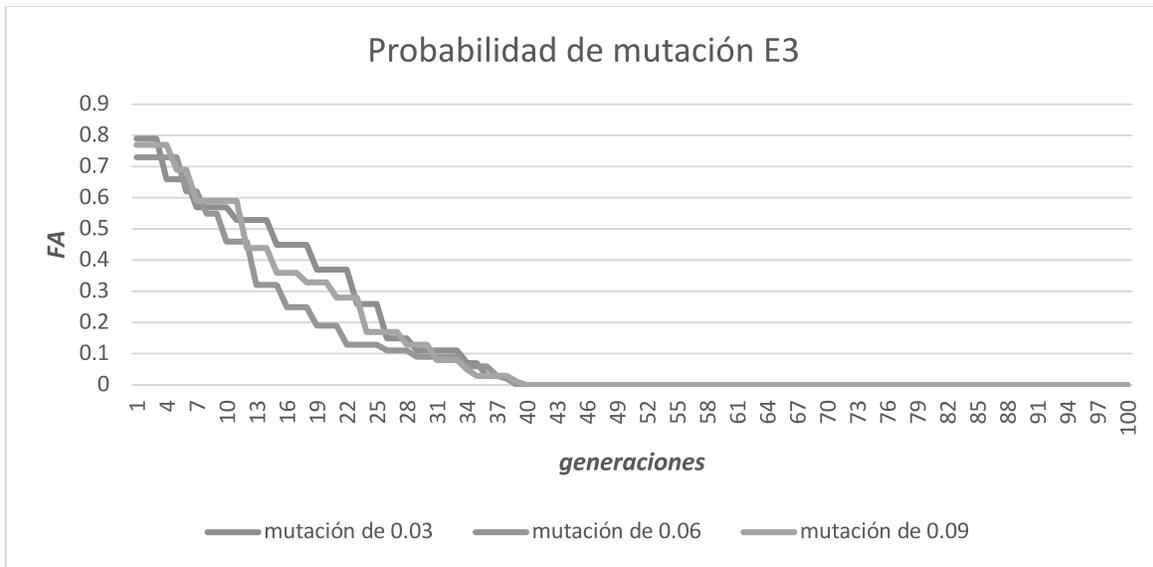
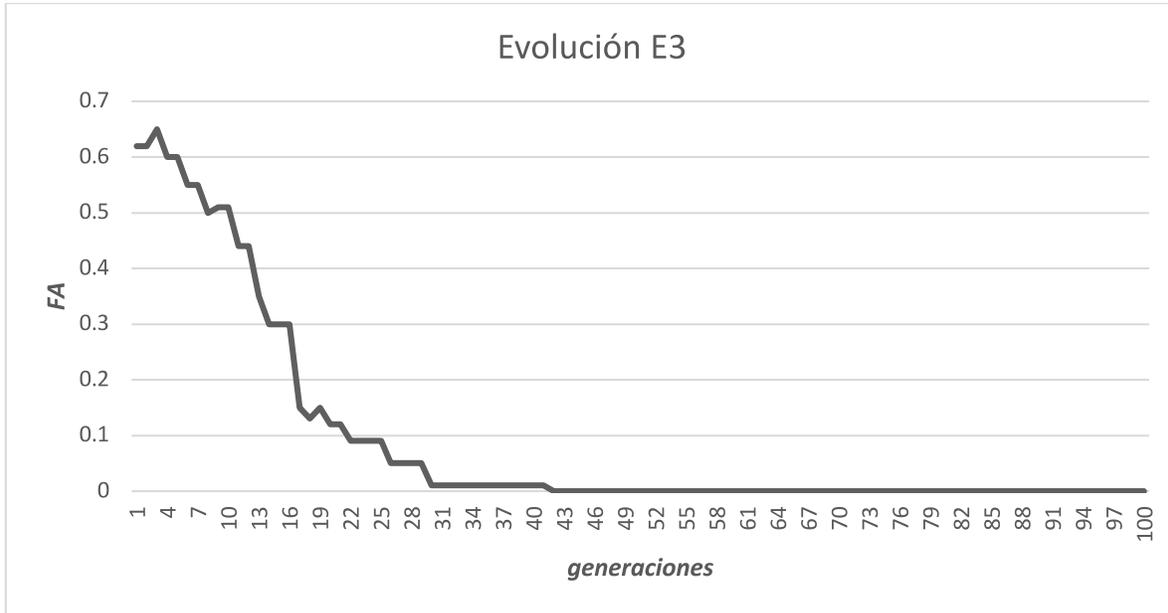


Figura 12. Probabilidad de mutación para el Estudiante 3.

La Figura 13 muestra el desempeño del AG propuesto, usando la probabilidad de cruce de 0.3 y la probabilidad de mutación de 0.03 para el estudiante E3. Con la experimentación de esta sección, la probabilidad de cruce de 0.3 y la probabilidad de mutación de 0.03 se utilizaron para todos los experimentos de la siguiente sección.



**Figura 13.** Evolución de AG para el Estudiante 3.

### 5.2.1. Resultados con Algoritmos genéticos

Utilizando las restricciones de horario y preferencias de los estudiantes de la Tabla 2, el método genético propuesto con los parámetros previamente ajustados obtuvo los resultados de aptitud mostrados en la columna central de asignación de cargas horarias automático de la Tabla 3. En la Tabla 3 y de acuerdo con estos resultados, nuestro método propuesto obtuvo resultados iguales o mejores que los actuales hechos a mano. Estos resultados apoyan nuestra hipótesis sobre un enfoque evolutivo, el algoritmo genético es suficiente para encontrar una buena carga horaria basada en las preferencias de los estudiantes y las limitaciones del plan de estudios.

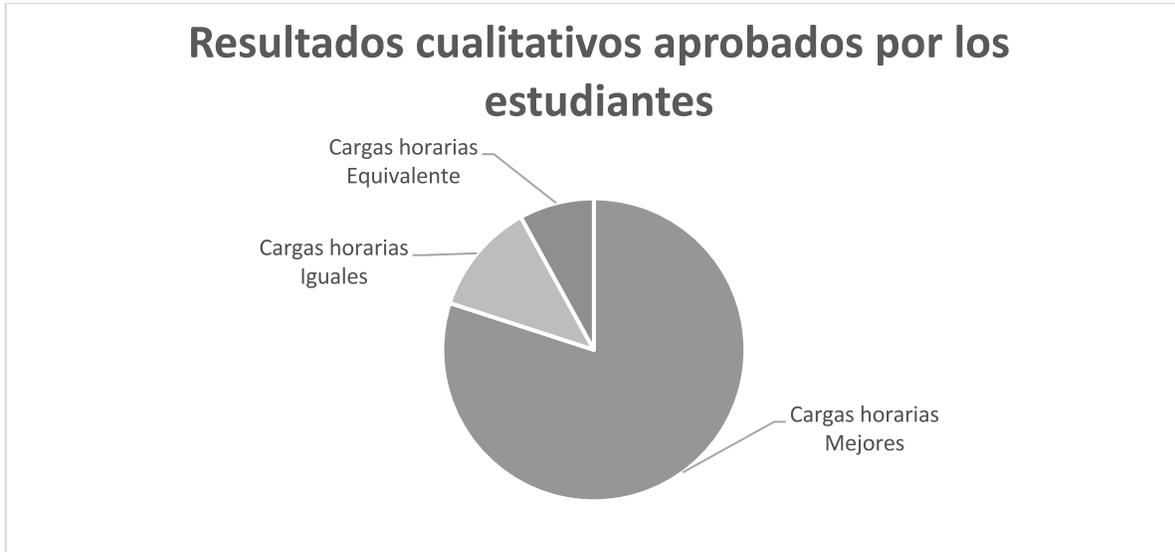
## Experimentación

Los valores cuantitativos de la Tabla 3 no muestran la valoración del punto de vista del alumno sobre la carga horaria automática. En la última columna de la Tabla 3, el alumno calificó la carga horaria automática con respecto a la hecha manualmente dentro de cuatro categorías: peor, igual, equivalente y mejor. Si bien, la evaluación cualitativa está implícita en la evaluación cuantitativa, la evaluación cualitativa muestra que para los estudiantes E4 y E5 (con la misma evaluación de aptitud) la carga horaria automática fue equivalente; y para los estudiantes E6, E7, E8, E10, E13, E14 y E17 la diferencia entre la evaluación de aptitud es demasiado cercana (menor o igual a 0.02) los estudiantes notan la diferencia ya que podría etiquetarse como equivalente.

**Tabla 3.** Comparación de la carga horaria manual y automáticamente con la función de aptitud para los 25 estudiantes de la Tabla 2. Además, la comparación muestra, en la columna de la derecha, el punto de vista de los estudiantes de la carga horaria automática con respecto a sus condiciones, éstas se dividen en cuatro categorías: peor, igual, equivalente y mejor.

Estudiante	Función de evaluación		Evaluación Cualitativa
	Carga horaria manual	Carga horaria Automática	
E1	0	0	Igual
E2	0	0	Igual
E3	0	0	Igual
E4	0	0	Equivalente
E5	0	0	Equivalente
E6	0.07	0.05	Mejor
E7	0.08	0.06	Mejor
E8	0.02	0	Mejor
E9	0.08	0.04	Mejor
E10	0.02	0.01	Mejor
E11	0.08	0.03	Mejor
E12	0.06	0.02	Mejor
E13	0.06	0.04	Mejor
E14	0.08	0.06	Mejor
E15	0.07	0	Mejor
E16	0.8	0	Mejor
E17	0.02	0	Mejor
E18	0.08	0	Mejor
E19	0.03	0	Mejor
E20	0.06	0	Mejor
E21	0.08	0	Mejor
E22	0.07	0	Mejor
E23	0.08	0	Mejor
E24	0.08	0	Mejor
E25	0.07	0	Mejor

La Figura 14 muestra la evaluación cualitativa aprobada por los estudiantes de la carga horaria generada por el método propuesto. Las cargas horarias automáticas fueron las mismas que las realizadas por el alumno en el 12% de los resultados. Las cargas horarias automáticas fueron equivalentes a las realizadas por el alumno y son una alternativa factible en el 8% de los resultados. Se crearon mejores cargas horarias que las generadas manualmente por el alumno en el 80% de los resultados. Cabe señalar que en ningún caso las cargas horarias fueron peores que las generadas manualmente o con superposición de horarios. Estos resultados aprueban nuestra hipótesis sobre la función de aptitud que se formula en relación con el punto de vista de los estudiantes.



**Figura 14.** Resultados cualitativos por los estudiantes.

### 5.3. Experimentos con Recocido Simulado

Se realizó más experimentación con el algoritmo de recocido simulado para este problema. Seleccionamos la heurística meta-SAA porque podría ser una alternativa factible para la implementación en los dispositivos móviles de los estudiantes. En este caso, los parámetros iniciales para el recocido simulado son: temperatura inicial de 2, temperatura mínima de 1/10000 y número máximo de iteraciones de 100, con la función de enfriamiento geométrico.

En las Figura 15, 16 y 17. Se muestra el desempeño del RS para los estudiantes E1, E2 y E3. En los tres casos, el RS funcionó bien con la ventaja de utilizar una única solución, por lo que el algoritmo RS requirió menos evaluaciones que el algoritmo genético.

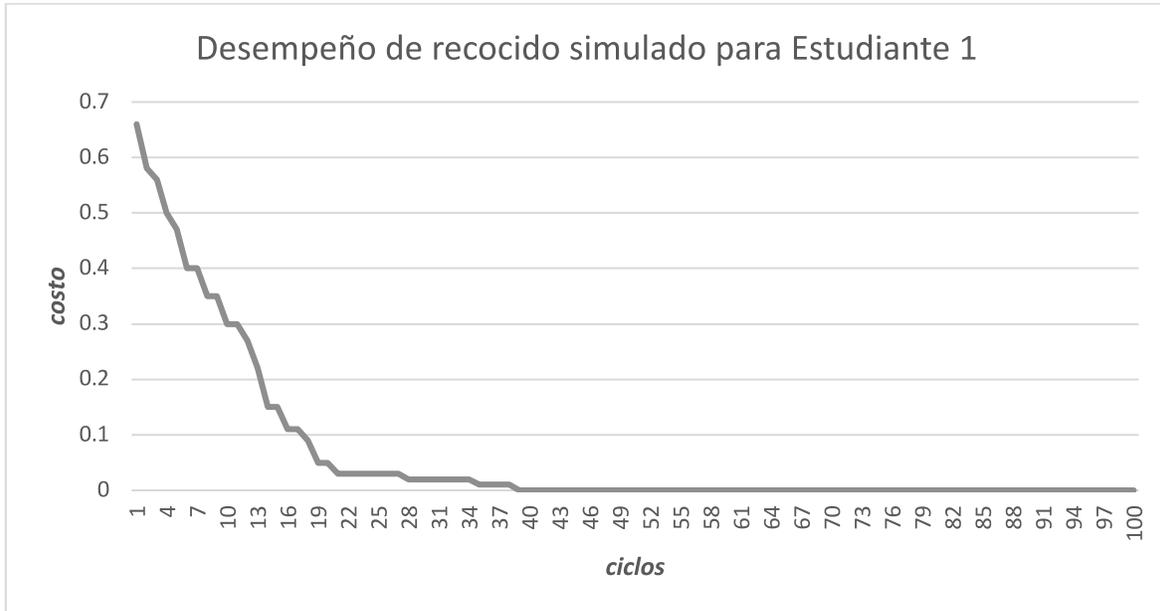


Figura 15. Recocido simulado para E1

En la Figura 16 se muestra el desempeño del RS, probado con una temperatura inicial de 2 y temperatura mínima de 1/10000, con 100 iteraciones, usando función de enfriamiento geométrico para E2.

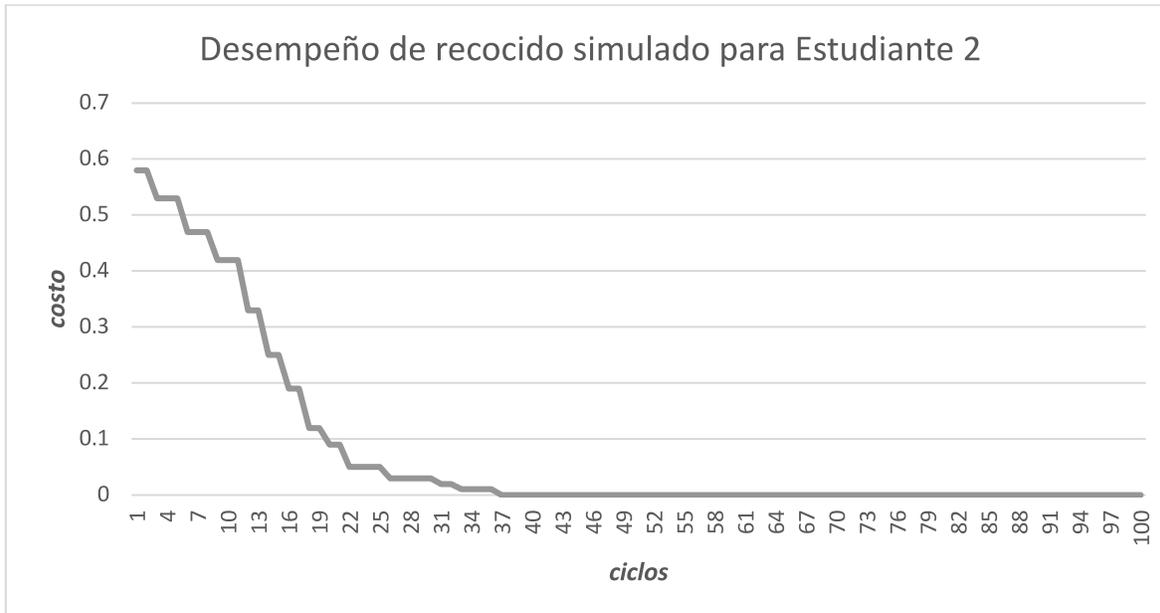
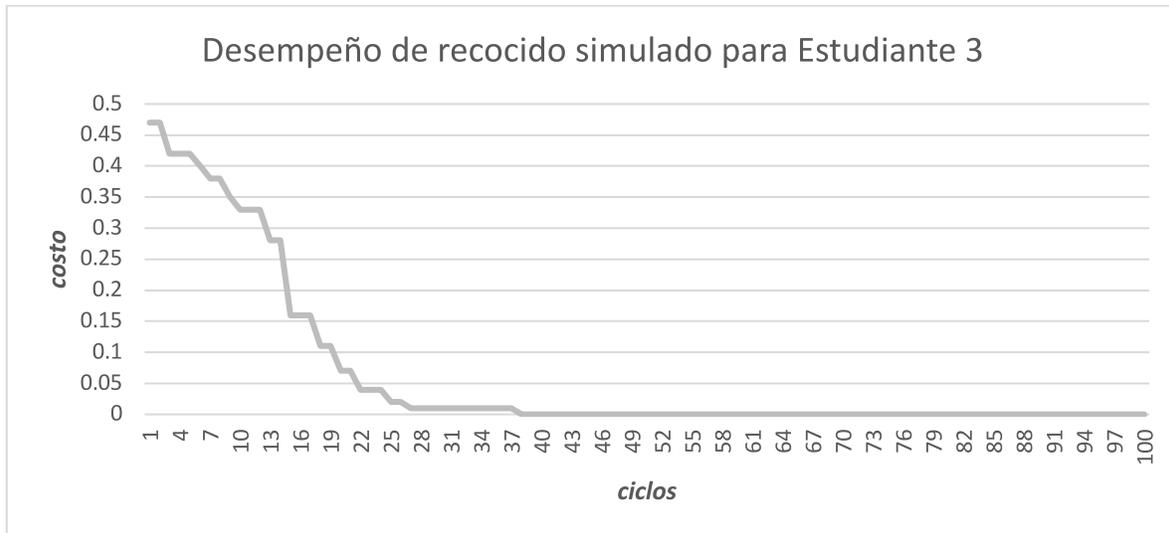


Figura 16. Desempeño de recocido simulado estudiante 2.

## Experimentación

En la Figura 17 se muestra el desempeño del RS, probado con una temperatura inicial de 2 y temperatura mínima de  $1/10000$ , con 100 iteraciones; usando función de enfriamiento geométrico para E3.



**Figura 17.** Recocido simulado para E3.

## Experimentación

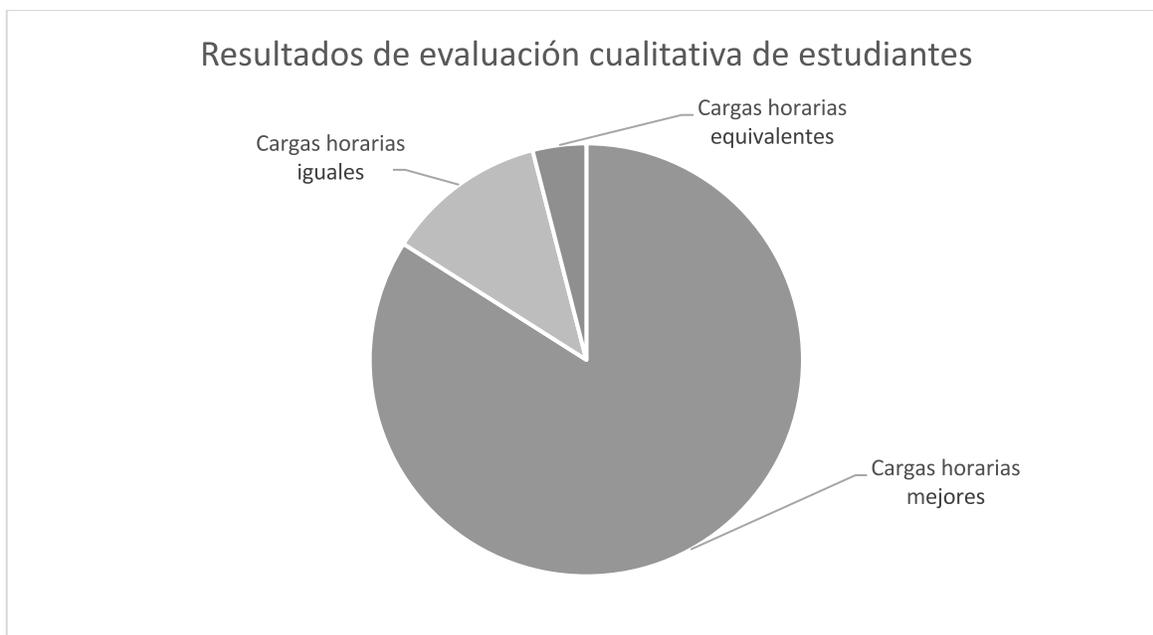
---

En la Tabla 4 se muestra la comparación de ACH generadas automáticamente y las generadas manualmente.

**Tabla 4.** Comparación de la carga horaria manual y automáticamente, tomando en cuenta la técnica de Recocido simulado con la función de aptitud para 25 estudiantes. Además, la comparación muestra en la columna de la derecha, el punto de vista de los estudiantes de la carga horaria automática

Estudiante	Función de evaluación		Evaluación
	Carga horaria manual	Cualitativa	
E1	0	0	Igual
E2	0	0	Igual
E3	0	0	Igual
E4	0	0	Equivalente
E5	0	0	Mejor
E6	0.07	0.03	Mejor
E7	0.08	0.02	Mejor
E8	0.02	0	Mejor
E9	0.08	0.01	Mejor
E10	0.02	0	Mejor
E11	0.08	0.03	Mejor
E12	0.06	0	Mejor
E13	0.06	0	Mejor
E14	0.08	0	Mejor
E15	0.07	0	Mejor
E16	0.8	0	Mejor
E17	0.02	0	Mejor
E18	0.08	0	Mejor
E19	0.03	0	Mejor
E20	0.06	0	Mejor
E21	0.08	0	Mejor
E22	0.07	0	Mejor
E23	0.08	0	Mejor
E24	0.08	0	Mejor
E25	0.07	0	Mejor

La Figura 18 se muestra la evaluación cualitativa aprobada por los estudiantes de la carga horaria generada por el método propuesto. Las cargas horarias automáticas fueron las mismas que las realizadas por el alumno en el 12% de los resultados. Las cargas horarias automáticas fueron equivalentes a las realizadas por el alumno y son una alternativa factible en el 4% de los resultados. Se crearon mejores cargas horarias que las generadas manualmente por el alumno en el 84% de los resultados. Cabe señalar que en ningún caso las cargas horarias fueron peores que las generadas manualmente o con superposición de horarios. Estos resultados validan nuestra hipótesis sobre la función de aptitud que se formula en relación con el punto de vista de los estudiantes.



**Figura 18.** Resultados cualitativos por los estudiantes.

### *5.4. Resumen*

En este capítulo fue descrita la tarea asignación de cargas horarias para estudiantes mediante dos técnicas computacionales, por ejemplo: un algoritmo genético y recocido simulado que fue seleccionado para aplicar el método propuesto en este trabajo de investigación.

El método propuesto indica los pasos a seguir para poder realizar una experimentación con mayores probabilidades para corroborar la hipótesis, y los experimentos son muestra de la funcionalidad del método propuesto.

Los resultados obtenidos indican que el método propuesto en este trabajo es efectivo para aplicarlo en la asignación de cargas horarias para estudiantes, siendo casos reales.



## CAPÍTULO 6.

# Conclusiones

---

Cabe mencionar que el desempeño educativo de una institución depende del desempeño educativo del alumno, el cual a su vez depende de la construcción de un buen horario personal, lo cual es una tarea que requiere mucho tiempo. La primera contribución de este trabajo es abordar un nuevo problema de horarios educativos que consiste en generar un horario de cursos universitarios personalizados para los estudiantes a partir de un catálogo de cursos publicado.

El segundo aporte es que se presenta la formulación de la función de aptitud que considera las restricciones y preferencias que presentan los estudiantes de la UAEMEX en relación con el punto de vista del estudiante. La tercera contribución se basa en el uso del conocido método de optimización de base genética para generar un cronograma personal de los cursos universitarios. Con la formulación de nuestra función de aptitud y el uso de un algoritmo evolutivo conocido, el método propuesto se puede adaptar fácilmente a otras instituciones de educación superior.

Se implementó el algoritmo Recocido Simulado con el fin de verificar la funcionalidad de las condiciones establecidas con base en las necesidades del estudiante, realizando la configuración de los parámetros para la obtención de resultados satisfactorios para el estudiante

La experimentación con el conjunto de datos y el método propuesto muestra buenos resultados cualitativos basados en la evaluación de los estudiantes. En todos los casos, el método propuesto e implementado con las dos técnicas computacionales presenta una carga horaria de alta calidad que, en comparación con los estudiantes, genera las mismas, equivalentes o mejores cargas horarias, pero nunca peores.

La creación de un conjunto de datos de prueba es una tarea que requiere mucho tiempo, por lo que en el futuro se espera automatizar esta tarea para realizar una mayor cantidad de pruebas.

### *6.1. Aportaciones*

La experimentación adicional con el algoritmo de recocido simulado sugiere que podría ser una alternativa viable para ser implementada en dispositivos móviles. En la experimentación, el recocido simulado mostró resultados mejores que el algoritmo genético, de acuerdo con la evaluación cualitativa del estudiante.

Al completar la investigación se pudo apreciar la variedad de enfoques que se pueden utilizar para resolver problemas de manera automática. En el futuro será útil realizar comparaciones con otros algoritmos metaheurísticos incrementando condiciones que sean útiles para el estudiante, así como condiciones para tomar en cuenta condiciones de los profesores e instituciones.



## Referencias

---

- Abramson, D., & Abela, J. (1991). A parallel genetic algorithm for solving the school timetabling problems. *Proceedings of the First International Workshop on Parallel Processing for AI*, 6–11.
- Alvarado-moya, P. (2016). *Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte Multiobjective optimization with expensive functions. Survey on the state of the art*. 16–24.
- Botello, S. (n.d.). *Optimización multiobjetivo de estructuras usando EDAs e información de vecindades*.
- Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), 266–280. [https://doi.org/10.1016/S0377-2217\(02\)00069-3](https://doi.org/10.1016/S0377-2217(02)00069-3)
- Carter, M. W. (2001). A comprehensive course timetabling and student scheduling system at the University of Waterloo. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2079 LNCS, 64–82. [https://doi.org/10.1007/3-540-44629-x\\_5](https://doi.org/10.1007/3-540-44629-x_5)

- Chávez, E. (2012). Revista Mexicana de Trastornos Alimentarios Mexican Journal of Eating Disorders. In Revista Mexicana de Trastornos Alimentarios/Mexican Journal of Eating Disorders (Vol. 3).
- Chávez, J. (2014). Algoritmo de recocido simulado aplicado al problema de secuenciamiento regular. *Avances Investigación En Ingeniería*, 11(1), 9. <https://doi.org/10.18041/1794-4953/avances.1.295>
- Chen, X., Ong, Y.-S., Lim, M.-H., & Tan, K. C. (2011). A multi-Facet survey on memetic computation. *IEEE transactions on evolutionary computation*, 15(5), 591. <https://doi.org/10.1109/TEVC.2011.2132725>
- Cotta, C., & Fernández, A. J. (2007). Memetic algorithms in planning, scheduling, and timetabling. *Studies in Computational Intelligence*, 49, 1–30. [https://doi.org/10.1007/978-3-540-48584-1\\_1](https://doi.org/10.1007/978-3-540-48584-1_1)
- Cruz Trejos, E. (2006). Negociación de portafolios de acciones usando la metaheurística recocido simulado. *Scientia et Technica*, 1(30), 301–306. <https://doi.org/10.22517/23447214.6553>
- Cubillos, G. (2013). Timetabling School problem and genetic algorithms. *Vínculos*, 10(2), 259–276.
- Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1), 117–135. [https://doi.org/10.1016/S0377-2217\(03\)00103-6](https://doi.org/10.1016/S0377-2217(03)00103-6)
- Escobar, W. (2012). Negociación de portafolios de acciones usando la metaheurística Recocido simulado (Vol. 11, Issue 21).
- Ferland, J. A., & Lavoie, A. (1992). Exchanges procedures for timetabling problems. *Discrete Applied Mathematics*, 35(3), 237–253. [https://doi.org/10.1016/0166-218X\(92\)90247-8](https://doi.org/10.1016/0166-218X(92)90247-8)
- Frühwirth, T., & Abdennadher, S. (2003). *University Course Timetabling*. 117–122. [https://doi.org/10.1007/978-3-662-05138-2\\_17](https://doi.org/10.1007/978-3-662-05138-2_17)
- Gayou, U. (2014). Información artículo Eating disorders identification of risk cases among students from Querétaro Identificación de casos de riesgo de trastornos alimentarios entre estudiantes de Querétaro. In Revista Mexicana de Trastornos Alimentarios Mexican Journal of Eating Disorders Journal homepage (Vol. 5). <http://journals.iztacala.unam.mx/>
- Gómez, J. (2020). Application of Genetic Algorithms Technique in the Generation of Academic Schedules. *KnE Engineering*, 2020, 150–165. <https://doi.org/10.18502/keg.v5i1.5927>
- Granada, T. (2006). Programación óptima de horarios de clase usando un algoritmo memético. *Scientia et Technica*, 1(30), 255–260. <https://doi.org/10.22517/23447214.6531>

- Henderson, D., Jacobson, S. H., & Johnson, A. W. (2006). The Theory and Practice of Simulated Annealing. In *Handbook of Metaheuristics* (Issue April). [https://doi.org/10.1007/0-306-48056-5\\_10](https://doi.org/10.1007/0-306-48056-5_10)
- Hernández, O. (2011). Simulated annealing algorithm to solve the economic lot scheduling problem and the basic cycle approach. *Ingeniare*, 19(3), 473–485. <https://doi.org/10.4067/s0718-33052011000300015>
- Hernández, R. (2008). *Programación de horarios de clases y asignación de salas en la Facultad de Ingeniería de la Universidad Diego Portales*.
- Hertz, A. (1992). Finding a feasible course schedule using Tabu search. *Discrete Applied Mathematics*, 35(3), 255–270. [https://doi.org/10.1016/0166-218X\(92\)90248-9](https://doi.org/10.1016/0166-218X(92)90248-9)
- Hoyos, J. G. (2007). Control en línea con algoritmos genéticos y recocido simulado. *Scientia Et Technica*, 13(035), 113–116. <https://doi.org/10.22517/23447214.5377>
- Jansen, E. (2016). Practices in timetabling in higher education institutions: A systematic review. *PATAT 2016 - Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling*, 295–316.
- Lü, Z., & Hao, J. K. (2010). Adaptive Tabu Search for course timetabling. *European Journal of Operational Research*, 200(1), 235–244. <https://doi.org/10.1016/j.ejor.2008.12.007>
- Martínez, A. A. (2012). Algoritmo basado en tabú search para el problema de asignación de horarios de clases. *Universidad de Carabobo*, 11. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjAgJTuhq7bAhX5GzQIHbQYAbkQFgguMAE&url=http%3A%2F%2Faraute.facyt.uc.edu.ve%2Fsites%2Fdefault%2Ffiles%2Farticulos%2FAlgoritmo-basado-en-Tabu-Search-para-el-problema-d>
- Myszkowski, P. B., & Skowroński, M. E. (2015). Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. *Soft Computing*, 19(12), 3599–3619. <https://doi.org/10.1007/s00500-014-1455-x>
- Petrovic, S., & Burke, E. (2004). University timetabling. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, December, 45-1-45–24.
- Santa Chávez, J. J., Peñuela Meneses, C. A., & Granada Echeverry, M. (2014). Algoritmo de recocido simulado aplicado al problema de secuenciamiento regular. *Avances Investigación En Ingeniería*, 11(1), 9. <https://doi.org/10.18041/1794-4953/avances.1.295>
- Santa Chávez, J. J., Peñuela Meneses, C. A., & Granada Echeverry, M. (2014). Algoritmo de recocido simulado aplicado al problema de secuenciamiento regular. *Avances Investigación En Ingeniería*, 11(1), 9. <https://doi.org/10.18041/1794-4953/avances.1.295>
- Segura, C. M. G. (2005). Educational Timetabling resuelto con Recocido Simulado y modelado bajo una arquitectura Web. Generalización y aplicación práctica: casos PATAT y UADY. 139.

## Referencias

---

- Soria, Pandolf, V. (2013). Algoritmos genéticos celulares con operadores de recombinación aplicados a problemas de optimización discretos. 1–21.
- Suárez, G. y C. (2013). Programación de Horarios Escolares basados en Ritmos Cognitivos usando un Algoritmo Genético de Clasificación Scheduling of School Hours based on Cognitive Rhythms using a Non-dominated Sorting Genetic Algorithm, NSGA-II. 24(1), 103–114. <https://doi.org/10.4067/S0718-07642013000100012>
- Vericat, F., Stoico, C. O., Carlevaro, C. M., & Renzi, D. G. (2011). Genetic algorithm for the pair distribution function of the electron gas. In Springer. <https://link.springer.com/article/10.1007/s12539-011-0108-3>
- Weare, R., Burke, E., & Elliman, D. (2017). A hybrid genetic algorithm for highly constrained timetabling problems. School of computer science and information technology university of nottingham computer science technical report no. Nottcs-tr-1995-8 a hybrid genetic algorithm for highly constrain. January 1995.
- Willmer Escobar, J., & Linfati, R. (n.d.). Un algoritmo metaheurístico basado en recocido simulado con espacio de búsqueda granular para el problema de localización y ruteo con restricciones de capacidad \* (vol. 11, issue 21).