



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

**“ASIGNACIÓN PERSONALIZADA DE CARGAS
HORARIAS PARA ESTUDIANTES MEDIANTE UN
ALGORITMO GENÉTICO”**

**PARA OBTENER EL TÍTULO DE
INGENIERO EN SOFTWARE**

PRESENTA

BRENDA SUNUAMI GONZÁLEZ LÓPEZ

DIRECTOR DE TESIS

DR. RENÉ ARNULFO GARCÍA HERNÁNDEZ

TIANGUISTENCO, MÉX. JULIO 2018



Universidad Autónoma del Estado de México
Unidad Académica Profesional Tianguistenco

El comité revisor designado por el Departamento Académico de la Unidad Académica Profesional Tianguistenco de la Universidad Autónoma del Estado de México, aprobó la tesis: **ASIGNACIÓN PERSONALIZADA DE CARGAS HORARIAS PARA ESTUDIANTES MEDIANTE UN ALGORITMO GENÉTICO** y autorizó la impresión de la misma del C. **BRENDA SUNUAMI GONZÁLEZ LÓPEZ** el día **02 DE JULIO DE 2018**.

ATENTAMENTE
PATRIA, CIENCIA Y TRABAJO

"2018, Año del 190 Aniversario de la Universidad Autónoma del Estado de México"

Revisor
Mtra. Griselda Areli Matias
Mendoza

Revisor
Dra. Yulia Nikolaevna Ledeneva

Asesor
Dr. René Arnulfo García
Hernández

Dra. en Admon. Adriana Fonseca Munguía
Jefa del Departamento Académico de
la UAP Tianguistenco
Vo.Bo.



Declaración de originalidad del trabajo escrito

Mediante esta carta hago constar que el trabajo de tesis presentado en este documento es original porque cita debidamente los contenidos utilizados como soporte a la investigación presentada, por lo que exoneró a la Universidad Autónoma del Estado de México de cualquier problema de derechos de propiedad intelectual.

Brenda Sunuami González López

Dedicatoria

Quiero dedicar este trabajo a mi familia que en todo momento ha estado a mi lado para apoyarme, mis padres el mayor soporte y los mejores cimientos de la familia, papá por siempre preocuparte por mí, y buscar mi bienestar, así como el de mis hermanos, a mi mamá por ser mi mejor amiga y confidente, y a mis hermanos por serlo, Luis: gracias por enseñarme que el tener más edad no significa tener un mayor conocimiento, Iris: gracias por ser mi cómplice en mis locuras y por aconsejarme.

"El tiempo que dedicaste a tu rosa, es lo que hace que ella sea tan importante para ti". Antoine de Saint-Exupéry.

Agradecimiento

Gracias a Dios por no dejarme nunca y guiar mi camino, por cada oportunidad brindada.

A la Universidad Autónoma del Estado de México, a la Unidad Académica Profesional Tianguistenco, por brindarme la oportunidad de poder estudiar y superarme, de poder crear y experimentar.

A mi familia que siempre está conmigo.

A mi asesor el Doctor René Arnulfo García Hernández, por el apoyo brindado y motivación durante mi último semestre de la licenciatura y en el desarrollo de mi tesis.

A mis revisores por brindarme su apoyo.

A mis amigos y compañeros, que me apoyaron en esta travesía.

Resumen

Actualmente los estudiantes de la Unidad Académica Profesional Tlanguistenco tienen problema cada semestre con la asignación de su carga horaria.

La asignación de carga horaria implica definir periodos de tiempo a determinadas actividades dependiendo de las condiciones del estudiante.

En este trabajo se implementa un algoritmo genético con optimización multiobjetivo, con el cual se realizaron pruebas con datos de algunos estudiantes de la Ingeniería de Software, para poder asignarles su carga horaria, tomando en cuenta las condiciones del estudiante. Demostrando que los algoritmos genéticos pueden brindar solución a este tipo de problemas de una manera eficiente.

PALABRAS CLAVE: Problemas de cargas horarias, algoritmos genéticos, optimización multiobjetivo.

CONTENIDO

páginas

DEDICATORIA	III
AGRADECIMIENTO	IV
RESUMEN	V
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE GRÁFICAS	VIII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1. INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA	6
1.2 MOTIVACIÓN	6
1.3 DELIMITACIONES.....	7
1.4 HIPÓTESIS	7
1.5 OBJETIVO GENERAL	7
1.5.1 OBJETIVOS ESPECÍFICOS.....	7
1.6 METODOLOGÍA EMPLEADA	8
1.7 METODOLOGÍA DE DESARROLLO DEL SISTEMA.....	8
1.8 ORGANIZACIÓN DE LA TESIS.....	9
CAPÍTULO 2. MARCO TEÓRICO.....	8
2.1 ALGORITMOS EVOLUTIVOS.....	9
2.2 ALGORITMOS GENÉTICOS	10
2.2.1 OPERADORES GENÉTICOS.....	11
2.2.2 OPERADOR DE SELECCIÓN	11
2.2.3 OPERADOR DE CRUZA	11
2.2.4 OPERADOR DE MUTACIÓN	11
2.3 DIAGRAMA DE UN ALGORITMO GENÉTICO.....	12
2.4 CROMOSOMAS O INDIVIDUOS.....	13
2.5 POBLACIÓN INICIAL.....	13
2.6 BÚSQUEDA TABÚ	14
2.7 ALGORITMO MEMÉTICO	14
2.8 OPTIMIZACIÓN MULTIOBJETIVO	14
2.9 MÉTODOS HEURÍSTICOS	14
2.10 RESUMEN DEL CAPÍTULO	15
CAPÍTULO 3. ESTADO DEL ARTE	28
3.1 TRABAJOS RELACIONADOS A LA GENERACIÓN DE CARGAS HORARIAS	29
3.1.1 PROGRAMACIÓN ENTERA.....	29
3.1.2 ALGORITMO DE BÚSQUEDA TABÚ	30
3.1.3 ALGORITMO MEMÉTICO	31
3.1.4 RITMOS COGNITIVOS	32
3.2 TRABAJOS RELACIONADOS CON OPTIMIZACIÓN MULTIOBJETIVO.....	33

3.3 RESUMEN DEL CAPÍTULO	34
CAPÍTULO 4. MÉTODO PROPUESTO	32
4.1 DESCRIPCIÓN DEL MÉTODO PROPUESTO.....	33
4.2 CODIFICACIÓN DEL INDIVIDUO	37
4.3 FUNCIÓN DE APTITUD	37
4.4 SELECCIÓN POR TORNEO.....	39
4.5 OPERADOR DE CRUZA EN n PUNTOS.....	39
4.6 OPERADOR DE MUTACIÓN POR INVERSIÓN BINARIA.....	39
4.7 CRITERIO DE PARADA	39
4.8 RESUMEN DEL CAPÍTULO	39
CAPÍTULO 5. EXPERIMENTACIÓN	52
5.1 EXPERIMENTO 1	53
5.2 AJUSTE DE PARÁMETROS	55
5.3 EXPERIMENTO 2	62
5.4 EXPERIMENTO 3	64
5.5 RESUMEN DEL CAPÍTULO	65
CAPÍTULO 6. CONCLUSIONES	79
6.1 APORTACIONES	80
6.2 TRABAJO FUTURO	80
APARTADO DE ANEXOS	82
ANEXO A	82
ANEXO B	85
ANEXO C.....	86
BIBLIOGRAFÍA	96

Índice de Figuras

FIGURA 1. METODOLOGÍA ESPIRAL	8
FIGURA 2. MODELO GENERAL DE ALGORITMO GENÉTICO.	12
FIGURA 3. INDIVIDUO BINARIO.....	13
FIGURA 4. INDIVIDUO NO BINARIO.....	13
FIGURA 5. SE MUESTRA EN LA IMAGEN DE LA IZQUIERDA UN EJEMPLO DE MINIMIZACIÓN, Y DE LADO DERECHO SE MUESTRA LA MAXIMIZACIÓN.....	34
FIGURA 6. SE MUESTRA LA CARGA ACADÉMICA CORRESPONDIENTE A LICENCIATURA DE INGENIERÍA DE SOFTWARE.....	35
FIGURA 7. MODELO DEL MÉTODO PROPUESTO.....	36

Índice de Gráficas

GRÁFICA 1. POBLACIÓN DE 100 INDIVIDUOS, 30 GENERACIONES, PROBABILIDAD DE CRUZA 100%, PROBABILIDAD DE MUTACIÓN 0%	54
GRÁFICA 2 TENIENDO UNA POBLACIÓN DE 100 INDIVIDUOS, 30 GENERACIONES, SE ANALIZAN DISTINTOS PORCENTAJES DE PROBABILIDAD DE CRUZA.	55
GRÁFICA 3. TENIENDO UNA POBLACIÓN DE 100 INDIVIDUOS, 30 GENERACIONES, SE ANALIZAN DISTINTOS PORCENTAJES DE PROBABILIDAD DE MUTACIÓN.	56

Índice de Tablas

TABLA 1. HORARIO PROPUESTO	3
TABLA 2. HORARIO SELECCIONADO DE ACUERDO CON LAS CONDICIONES DEL ESTUDIANTE.	3
TABLA 3. CONDICIONES DE LA FUNCIÓN DE APTITUD PARA QUE EL ESTUDIANTE PUEDA SELECCIONAR SU CARGA HORARIA.....	4
TABLA 4. TRABAJOS RELACIONADOS A LA GENERACIÓN DE CARGAS HORARIAS.	33
TABLA 5. SE MUESTRA UN EJEMPLO DEL HORARIO QUE DIO A CONOCER LA UAPT PARA LA INGENIERÍA DE SOFTWARE EN EL PERÍODO 2018A.	34
TABLA 6. SE MUESTRAN LAS CONDICIONES QUE CONFORMAN LA FA.	34
TABLA 7. EJEMPLO DE LISTA DE UNIDADES DE APRENDIZAJE QUE PUEDE TOMAR EL ESTUDIANTE (LUAP).....	35
TABLA 8. CONDICIONES DEL ESTUDIANTE 1 (E1).	53
TABLA 9. CARGA HORARIA SELECCIONADA POR EL ESTUDIANTE1 (E1), EN DÓNDE SE TRASLAPAN UNIDADES DE APRENDIZAJE.	57
TABLA 10. SE MUESTRA UN HORARIO DEL CUAL TOMA LAS UNIDADES DE APRENDIZAJE: ADMINISTRACIÓN Y ORGANIZACIÓN DE PROYECTOS DE SOFTWARE, CALIDAD DE SOFTWARE Y SEGURIDAD INFORMÁTICA.	59
TABLA 11. SE MUESTRA UN HORARIO DEL CUAL TOMA LAS UNIDADES DE APRENDIZAJE: DATAWAREHOUSE, BASE DE DATOS ORIENTADA A OBJETOS.....	59
TABLA 12. SE MUESTRA UN HORARIO DEL CUAL TOMA LAS UNIDADES DE APRENDIZAJE: METODOLOGÍA DE LA INVESTIGACIÓN, CÓMPUTO EVOLUTIVO.	59
TABLA 13. SE MUESTRAN LAS UNIDADES DE APRENDIZAJE QUE PUEDE TOMAR EL ESTUDIANTE 1 (E1).....	60
TABLA 14. CARGA HORARIA SELECCIONADA POR EL ALGORITMO GENÉTICO PARA EL ESTUDIANTE1 (E1).	61
TABLA 15. CONDICIONES DEL ESTUDIANTE 2 (E2).	62
TABLA 16. CARGA HORARIA SELECCIONADA POR EL ESTUDIANTE 2 (E2).	62
TABLA 17. CARGA HORARIA ASIGNADA POR EL GENÉTICO PARA EL ESTUDIANTE 2 (E2).	63
TABLA 18. CONDICIONES DEL ESTUDIANTE 3 (E3).	64
TABLA 19. CARGA HORARIA SELECCIONADA POR EL ESTUDIANTE 3 (E3).	64
TABLA 20. CARGA HORARIA SELECCIONADA POR EL GENÉTICO PARA EL ESTUDIANTE 3 (E3).	65



CAPÍTULO 1.

Introducción

La solución de problemas en el mundo real se ha visto beneficiada cada vez más por técnicas computacionales, permitiendo tener mejores resultados, y más rápidos.

Los problemas de generación de horarios y asignación de recursos en instituciones educativas han sido ampliamente estudiados en la literatura. Estos problemas pueden ser clasificados de acuerdo con el tipo de institución educativa y por el tipo de eventos a programar, clases o evaluaciones. En el caso de las universidades, debe existir cierta flexibilidad en los horarios y en la selección de los cursos que toma cada estudiante.

La creación de cargas horarias en una institución educativa es un procedimiento el cual consta de seleccionar unidades de aprendizaje de acuerdo con los horarios ofertados por la institución. Este problema surge debido a que en todos los períodos académicos los horarios ofertados cambian. En la creación de cargas horarias se debe desarrollar la planificación de los recursos escolares (situación académica, unidades de aprendizaje, horarios) que cumplan con todas las condiciones impuestas por el estudiante para que obtenga una solución aceptable por las personas involucradas.

La Universidad Autónoma del Estado de México (UAEM) cuenta con diferentes Facultades y Unidades Académicas, una de ellas es la Unidad Académica Profesional Tianguistenco (UAPT). La UAPT cuenta con un plan curricular flexible, el cual puede ser cursado como mejor le convenga al estudiante, considerando sus propias condiciones.

Para poder cursar unidades de aprendizaje el estudiante o tutorado debe contar con aprobación del tutor. El tutorado es aquella persona que recibe orientación en su desempeño académico, desde el ingreso a una institución educativa, hasta llevarlo a su titulación). Cabe resaltar que el tutor es el encargado de verificar el desarrollo académico del estudiante (UAEM, 2009). Algunas de las competencias que debe desarrollar el tutor son:

- Una buena relación con sus tutorados.
- Motivación para ayudar a los estudiantes en su formación académica y personal.

La UAPT cuenta con 1,000 estudiantes aproximadamente, en las cuatro carreras ofertadas por la institución. Cada uno de estos estudiantes se enfrenta al problema de la creación de carga horaria para cada semestre, la cual cabe señalar se realiza de manera manual.

El proceso para la creación de la carga horaria es el siguiente: Primero, la UAPT da a conocer el horario de clases, el cual muestra las unidades de aprendizaje correspondientes al semestre actual y los horarios asignados. A partir de estos horarios, los estudiantes eligen su carga horaria de acuerdo con sus condiciones académicas, de tiempo y de economía.

Si el estudiante es regular, las unidades de aprendizaje se seleccionan de acuerdo con su semestre correspondiente. De lo contrario, el estudiante tiene que buscar, en los horarios publicados, si se ofertan las unidades de aprendizaje faltantes a asignar. Realizando la selección de unidades de aprendizaje que se adecuen a la disponibilidad de horario.

El problema de asignar las cargas horarias se caracteriza por ser un problema de optimización, debido a que se pueden generar una gran cantidad de posibles cargas académicas para un estudiante.

Por ejemplo, si el estudiante tiene una situación académica irregular, deben seleccionarse unidades de aprendizaje de diferentes semestres, como se muestra en el horario propuesto de la tabla 1.1 y con la condición:

- Número de créditos Mínimo de 25 y número de créditos Máximo de 30. Se deben seleccionar las siguientes unidades de aprendizaje: Arquitectura de Computadoras, Seguridad Informática, Calidad de Software, Metodología de la Investigación. Como se muestra en la tabla 1 para la Ingeniería de Software.

Tabla 1. Horario propuesto

Horarios	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	Seguridad Informática 8	Calidad de Software 8	Arquitectura de Computadoras 4	Arquitectura de Computadoras 4	Cómputo Evolutivo 9	Metodología de la Investigación 9
8:00 - 9:00						
9:00 - 10:00						
10:00 - 11:00			Base de Datos 6			
11:00 - 12:00	Álgebra 1					
12:00 - 13:00						
13:00 - 14 :00						

Tabla 2. Horario seleccionado de acuerdo con las condiciones del estudiante.

Horarios	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	Seguridad Informática 8	Calidad de Software 8	Arquitectura de Computadoras 4	Seguridad Informática 8		Metodología de la Investigación 9
8:00 - 9:00						
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14 :00						

La carga horaria para el estudiante, de acuerdo con sus condiciones indica que las Unidades de Aprendizaje seleccionadas son las siguientes:

Seguridad Informática: 6° semestre

Calidad de Software: 8° semestre

Metodología de la Investigación: 8° semestre

Arquitectura de Computadoras: 8° semestre

El proceso de asignación de cargas horarias en estudiantes no es algo fácil, ya que el estudiante busca personalizar la carga horaria tomando en cuenta no solo la condición del número de créditos mínimos y máximos, sino que existen en total 5 condiciones las cuales se muestran en la tabla 3.

Tabla 3. Condiciones de la Función de Aptitud para que el estudiante pueda seleccionar su carga horaria

Condición	Abreviatura
<i>Condiciones de Tiempo por Semana</i>	(CTS)
<i>Condición de Horas Traslapadas</i>	(CHT)
<i>Condición de Unidades de Aprendizaje que Puede y Debe Cursar el estudiante</i>	(CUAPDC)
<i>Condición de Sugerencia de Cargas Horarias en Tiempo Real</i>	(CSCHTR):
<i>Condición de Créditos Máximos y Mínimos para cada semestre</i>	(CCMAX) y (CCMIN)

1. 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 Universidad, por lo cual se implementa esta condición.
2. 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.
3. 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.
4. 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 en horario, cancelación o apertura adicional, etc.
5. Condición de Créditos Máximos (CCMAX) y Mínimos para cada semestre (CCMIN): El estudiante tiene un mínimo y máximo de créditos que cumplir, ya que forma parte del reglamento de la UAPT, es decir el horario propuesto por el estudiante no puede excederse del límite de créditos, así como tampoco le pueden faltar créditos.

En este trabajo se presenta un método que soluciona el problema de Asignación Personalizada de Cargas Horarias para estudiantes en la UAPT y se dará solución al mismo mediante la implementación de un algoritmo genético, debido a que es la forma de solucionar problemas complejos buscando obtener la mejor posible solución (Holland, 1992).

En relación con la problemática de la Asignación personalizada de cargas horarias para estudiantes, se pretende combatir algunos factores que son provocados por una mala organización en las cargas horarias, por ejemplo:

- Deserción Escolar: Los estudiantes deciden abandonar sus estudios ya que el dejar de tomar unidades de aprendizaje les provoca rezago en su desarrollo académico (Goicovic Donoso, 2002).

- Mala Alimentación: El estudiante elige sus unidades de aprendizaje en horarios continuos y no tiene oportunidad de comer adecuadamente, lo cual provoca problemas alimenticios (Levy & Amaya, 2015).
- Problemas Económicos: El estudiante elige a diferentes horas sus unidades de aprendizaje y hay una gran diferencia de tiempos, por lo cual se ve obligado a comprar comida en la Universidad, o regresar a su casa, implicando gastos extras a su bolsillo (Ruiz, Garcia, & Perez, 2014).

La organización de cargas horarias ha sido una gran problemática en las instituciones educativas, siendo la causa de múltiples investigaciones. Entre estas investigaciones se encuentran diferentes técnicas de optimización computacionales, las cuales han sido implementadas mediante Cómputo Evolutivo.

El cómputo evolutivo es proceso basado en la evolución, con posibles soluciones a un problema, las cuales se modifican o evolucionan obteniendo cada vez mejores resultados (Eiben & Smith, 2004). Por ejemplo: camino de hormigas (Flores Pichardo, 2001), algoritmos genéticos (Laguna & Vega, 2009) y algoritmos meméticos (Cotta & Cotta, 2016).

Cabe resaltar que no se han encontrado investigaciones y sistemas de cargas horarias que haga referencia a las condiciones de estudiante, lo cual es un problema, debido a que el estudiante en ocasiones no coincide con los horarios sugeridos por la institución. Las investigaciones y sistemas encontrados en el estado del arte están dirigidos a profesores e instituciones.

1.1 Planteamiento del problema

Dado que existen muchas posibilidades de horarios para cada conjunto de condiciones personalizadas del estudiante, 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 adapten al estudiante, ¿Cómo encontrar la mejor carga horaria que satisfaga las condiciones del estudiante en tiempo real?

1.2 Motivación

La motivación académica comienza debido a que el problema de asignación personalizada de cargas horarias para estudiantes puede solucionarse con diferentes técnicas, entre ellas: algoritmos evolutivos. En el estado del arte se han encontrado investigaciones que comprueban que los algoritmos evolutivos son funcionales para la

solución a estos problemas. La razón para utilizar algoritmos genéticos es que se cuenta con conocimientos previos sobre esta técnica, la cual es utilizada para la búsqueda y optimización de problemas (Laguna & Vega, 2009).

Los algoritmos genéticos se han implementado para la generación de horarios a Instituciones y profesores. Por otro lado, el interés sobre el tema surge de la necesidad que tienen los estudiantes de no poder asignar una carga horaria que se adecuarán a las condiciones.

1.3 Delimitaciones

Por el momento se realiza la modelación del algoritmo genético que asigna de cargas horarias personalizadas para un estudiante, pero no se realiza la implementación del sistema. Las pruebas si se realizan con datos reales de estudiantes.

1.4 Hipótesis

Dado que existen muchas posibilidades de horarios para cada conjunto de condiciones personalizadas del estudiante, entonces la asignación de cargas horarias es un problema de optimización, el cual debe ser valorado de acuerdo con las condiciones que se adapten al estudiante.

Si se utiliza un algoritmo de cómputo evolutivo, entonces se podrán asignar cargas horarias personalizadas de acuerdo con las condiciones del estudiante.

1.5 Objetivo general

Generar cargas horarias personalizadas para los estudiantes de la UAPT de acuerdo con las condiciones académicas anteriormente mencionadas.

1.5.1 Objetivos específicos

- Asignar de manera personalizada una carga horaria al estudiante.
- Minimizar el tiempo de elaboración de la carga horaria.
- Realizar la comparación de la carga horaria sugerida por el genético con la carga horaria realizada con el estudiante.

1.6 Metodología empleada

La metodología en espiral describe el ciclo de vida de un software por medio de espirales, los cuales son repetidos hasta que se el producto es terminado. Una característica clave del desarrollo en espiral es la minimización de riesgos en el desarrollo de software.

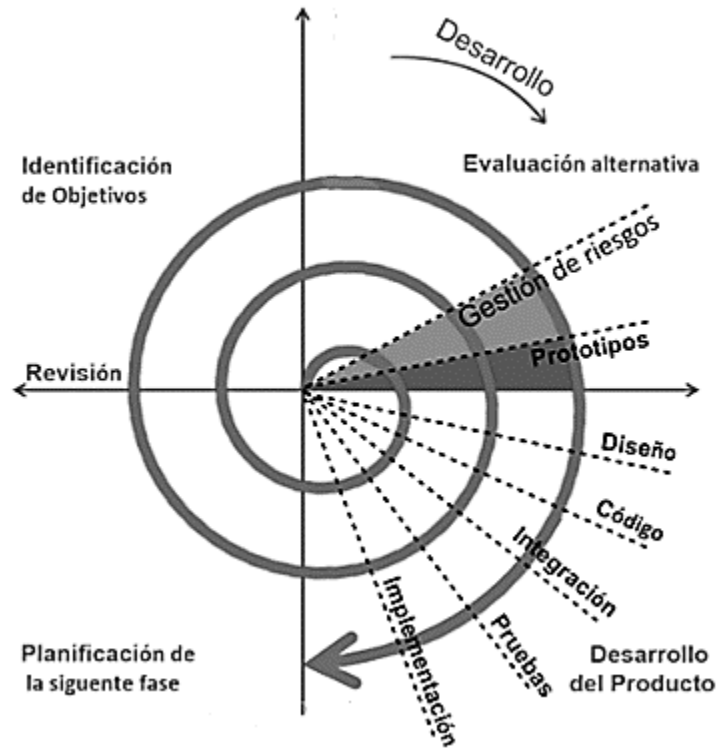


Figura 1. Metodología espiral.

El modelo empieza determinando los objetivos y las limitaciones del software al inicio de cada repetición. En la siguiente etapa se crean los modelos de prototipo del software, esto incluye el análisis de riesgos. Luego el ciclo de vida del software se usa para desarrollar el producto. En la cuarta etapa es donde se prepara la siguiente repetición.

1.7 Metodología de desarrollo del sistema

La metodología que se va a desarrollar se basa en un algoritmo genético que integra la definición de cargas horarias. Para esto se definen combinaciones de cargas horarias dependiendo de las condiciones del estudiante.

1.8 Organización de la tesis

La estructura de la tesis será de la siguiente manera:

Capítulo I. Introducción: En este capítulo se presenta una introducción al problema de asignación de cargas horarias para estudiantes, tomando en cuenta las condiciones del estudiante, así como también se establecen las delimitaciones para especificar lo que se realiza en esta investigación.

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 personalizada de cargas horarias para estudiantes.

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

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

Capítulo IV. Método Propuesto: Este capítulo indica que se propone implementar un algoritmo genético, para la optimización a esta problemática de cargas horarias.

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

Capítulo VI. Conclusiones: Se pretende realizar el modelo de un algoritmo genético que asigne cargas horarias personalizadas para estudiantes



CAPÍTULO 2.

Marco Teórico

Como se mencionó anteriormente, el problema de asignación de cargas horarias en estudiantes se trata como problema de optimización, donde es necesario realizar combinaciones, para poder asignar de manera personalizada cargas horarias a estudiantes.

Existen diferentes tipos de algoritmos que pueden solucionar este problema, por ejemplo: camino de hormigas (Algarín, 2010), algoritmos meméticos (Ocampo & Baquero, 2006), búsqueda tabú (Milla, 2012), entre otros.

2.1 Algoritmos evolutivos

Si algo funciona bien, ¿Por qué no imitarlo? (Gestalt, 2010), esta pregunta hace referencia directamente a los orígenes de la computación evolutiva.

El término algoritmos evolutivos, es empleado para describir la solución de problemas de optimización o búsquedas (Andaluz, 2004).

Los algoritmos evolutivos trabajan con una población de individuos, en donde presentan posibles soluciones a un problema (esta población se somete a ciertas transformaciones y después a un proceso de selección, que favorece a los mejores).

Los algoritmos evolutivos combinan la búsqueda aleatoria, por transformaciones de la población, con una búsqueda dirigida dada por la selección.

Principales componentes:

- Población de individuos: son una representación (no necesariamente directa) de posibles soluciones.
- Procedimiento de selección: Está basado en la aptitud de los individuos para resolver el problema.
- Procedimiento de transformación para construir nuevos individuos a partir de los anteriores.

La idea principal de los algoritmos evolutivos es mantener un conjunto de individuos que representan una posible solución del problema, estos individuos se mezclan y compiten entre sí, siguiendo el principio de selección natural, por el cual sólo los mejor adaptados sobreviven al paso del tiempo.

Un algoritmo evolutivo puede ser implementado de forma independiente al problema, lo cual hace que estos algoritmos sean robustos, por ser útil para cualquier problema, pero a la vez débiles, pues no están especializados en ningún problema en concreto (Holland, 1975).

2.2 Algoritmos Genéticos

Los algoritmos genéticos están basados en el proceso genético de los organismos vivos (Darwin, 1859) porque simulan el proceso de la evolución natural (Kuri, 2000). Se dice que un algoritmo genético es una forma de optimizar algún problema. El algoritmo genético cuenta con una población la cual se divide en individuos de donde se seleccionan los mejores individuos para crear nuevos individuos, realizando así una generación nueva (Pérez, 2002).

Los algoritmos genéticos trabajan con una población de individuos, en donde cada uno de ellos representa una posible solución al problema que se desea resolver (Ricci, 2013).

Una generación se obtiene por medio de los operadores de reproducción, para el cual existen 2 tipos:

- Cruce: Esta trata de una reproducción de tipo sexual. Se genera una descendencia a partir del mismo número de individuos de la generación anterior.
- Copia: Se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación.

En un algoritmo genético se pueden observar los siguientes módulos:

- Módulo evolutivo (interpreta la información de un cromosoma), así mismo mide la calidad del cromosoma.
- Módulo poblacional (técnica de selección y reemplazo).
- Módulo reproductivo (contiene los operadores genéticos).

Los algoritmos genéticos pretenden realizar la búsqueda de los mejores resultados optimizando todas las posibles combinaciones. Generalmente son métodos adaptativos usados en problemas de búsqueda y optimización de parámetros, basado en la reproducción sexual y en el principio de supervivencia del más apto (Fogel, 2000).

Los algoritmos genéticos son métodos los cuales pueden adaptarse a la búsqueda y optimización de problemas (Vásquez, 2012), el autor dice que los algoritmos genéticos están basados en la reproducción sexual.

2.2.1 Operadores genéticos

De acuerdo con el autor Koza (Koza, 1992), 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.

2.2.2 Operador de selección

Este operador es el encargado de elegir los individuos que pueden reproducirse (Gestal, 2010). El algoritmo genético se basa en que la naturaleza da una mayor oportunidad de reproducción a los individuos más aptos, las técnicas más comunes para identificar individuos son:

- Selección por ruleta: Los mejores individuos tienen más posibilidades de reproducirse (Blickle and Thiele, 1995).
- 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.

2.2.3 Operador de cruce

Este operador consiste en elegir a dos individuos de manera aleatoria con una determinada probabilidad de cruce, y mezclarlos para poder tener un nuevo individuo cuyos individuos son combinaciones de sus padres (Villagra, 2013).

2.2.4 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, y es el operador encargado de aumentar o reducir el espacio de búsqueda en un algoritmo genético (Gestal, 2010).

2.3 Diagrama de un algoritmo genético

Tomando como referencia el diagrama de la figura 1 propuesto se pueden mostrar cada uno de los procesos que se realizan en el algoritmo genético para la asignación de cargas horarias personalizadas para estudiantes.

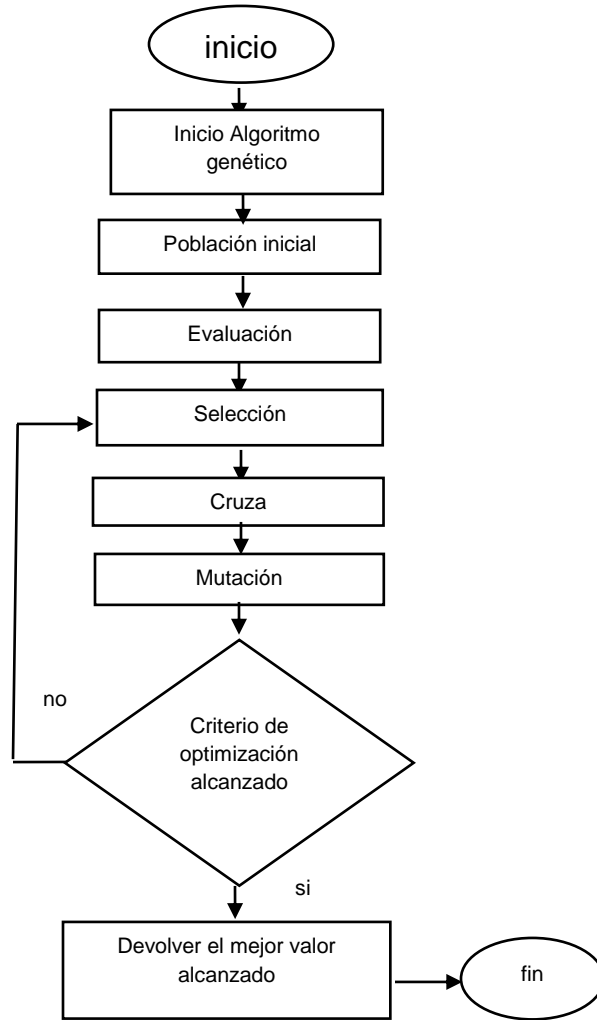


Figura 2. Modelo general de algoritmo genético.

2.4 Cromosomas o individuos

Es la representación de una solución al problema, la cual puede estar compuesta de uno o más genes, como se muestra en las figuras 2 y 3. El individuo se conforma de toda la información relevante del problema. Los individuos deben ocupar el menor espacio posible y deben ser fáciles de preservar durante la ejecución del algoritmo genético (Ahumada, 2015).

Los individuos son binarios siempre y cuando estén representados con 0 y 1.

Parámetro 1	Parámetro 2	Parámetro 3
0111010	001	11

Figura 3. Individuo binario

También existen individuos no binarios.

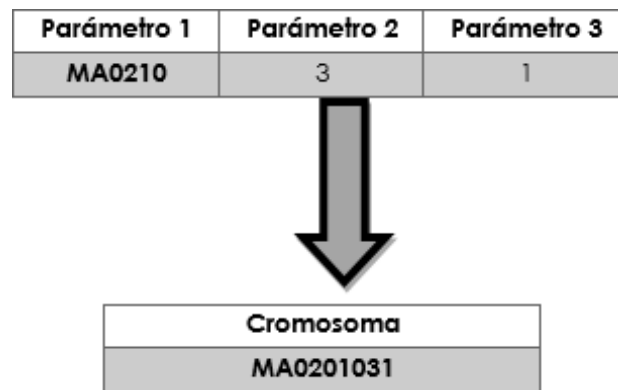


Figura 4. Individuo no binario

2.5 Población inicial

La población es un conjunto de soluciones en un ambiente particular, es la parte en donde evolucionan los algoritmos evolutivos, ya que los individuos son solo elementos estáticos que no cambian o se adaptan. La evolución se da al pasar de generación en generación, aunque desde luego un número mayor otorga una mejor solución de un problema (Reyes, 2011).

Los individuos más aptos son seleccionados como candidatos a reproducirse, es decir dan origen a nuevos individuos, o una solución al problema (Goldberg 1989). La población inicial puede ser generada de manera aleatoria (Sprockel 2013), es importante que la primera población tenga un tamaño lo suficientemente grande para poder garantizar una diversidad de soluciones (Ochoa, 2017).

2.6 Búsqueda tabú

La búsqueda tabú se basa en la explotación de diversas estrategias especiales de resolución de problemas, basadas en procedimientos de aprendizaje. Lo que pretende la búsqueda tabú es derivar y explotar una colección de estrategias inteligentes para la resolución de problemas (Glover, Melián, & Tabú, 2003) .

2.7 Algoritmo memético

Los algoritmos Meméticos son técnicas de optimización que combinan conceptos tomados de otras metaheurísticas, tales como la búsqueda basada en poblaciones (como los algoritmos evolutivos), y la mejora local (Moscato, de, & 2003, n.d.). Un algoritmo memético es un paradigma de optimización basado en la explotación de conocimiento acerca del problema que se desea resolver (Cotta & Cotta, 2016).

2.8 Optimización multiobjetivo

La toma de decisiones puede verse como un problema de optimización en el cual generalmente están involucrados dos o más objetivos. La optimización multiobjetivo o multicriterio se caracteriza por no tener una única solución, sino un conjunto de soluciones válidas (On, Pulido, Maestro, & Artificial, 2001).

2.9 Métodos heurísticos

La palabra heurística procede del griego εὐρίσκειν, 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 (Cunquero, 2007).

Los métodos heurísticos son un procedimiento simple, basado en el sentido común, que ofrece una buena solución a problemas difíciles, de un modo fácil y rápido (Zanakis & Evans, 1981).

Algunos autores proponen la siguiente clasificación de métodos de resolución mediante heurísticos (Silver et al., 1980):

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.

2.10 Resumen del capítulo

En este capítulo se han mencionado los conceptos básicos que son utilizados en este trabajo. Se comienza hablando de lo que son los algoritmos genéticos, los cuales están basados en la búsqueda y optimización de problemas.

Para la realización de un algoritmo genético se requiere básicamente de una población inicial que está conformada por un conjunto de soluciones. Es importante considerar que la primera población debe tener un tamaño lo suficientemente grande para poder garantizar una diversidad de soluciones.

Un algoritmo genético contiene los siguientes operadores genéticos:

Operador de selección: Este operador es el encargado de elegir qué individuos son los que pueden reproducirse. Los dos operadores más utilizados son: selección por ruleta y selección por torneo. En la selección por ruleta los mejores individuos tienen más posibilidades de reproducirse y en la selección por torneo se seleccionan k individuos al azar y se escoge el mejor, este proceso se repite hasta que genere una población.

Operador de cruce: Este operador consiste en elegir a dos individuos de manera aleatoria con una determinada probabilidad de cruce, después se mezclan para poder tener un nuevo individuo, de igual manera cuenta con diferentes tipos de cruces.

Operador de mutación: La mutación brinda aleatoriedad a los individuos de una población, al igual que con el operador de cruce, este operador cuenta con probabilidad de mutación. Este operador es el encargado de aumentar o reducir el espacio de búsqueda en un algoritmo genético. Existen diferentes tipos de mutación, entre ellos está la mutación por inversión binaria, en el cual el gen a mutar se reemplazará, por el valor contrario.

En cambio, los algoritmos genéticos son una rama de los algoritmos evolutivos que son empleados para describir la solución de problemas de optimización o búsqueda. Los algoritmos evolutivos trabajan con una población de individuos, los cuales representan posibles soluciones a un problema. Los algoritmos evolutivos combinan la búsqueda aleatoria, por transformaciones de la población, con una búsqueda dirigida dada por la selección.

Dentro de las técnicas para la optimización de problemas esta la búsqueda tabú. La búsqueda tabú está basada en la explotación de diversas estrategias especiales de resolución de problemas y en procedimientos de aprendizaje.

Los algoritmos meméticos son un paradigma de optimización basado en la explotación de conocimiento acerca del problema que se desea resolver.

La optimización multiobjetivo es implementada en esta investigación, ya que se trata de diversos objetivos a cumplir, Ésta es caracterizada por no tener una única solución, sino un conjunto de soluciones válidas.

Un método heurístico es un procedimiento para resolver problemas de optimización para obtener una buena solución. Algunos autores proponen la siguiente clasificación de métodos de resolución.

Los métodos constructivos se caracterizan por construir una solución de definiendo diferentes partes de ella en sucesivos pasos.

En los métodos de descomposición, el problema se divide en pasos más pequeños y la solución se obtiene a partir de la solución de cada uno de estos.

Los métodos de manipulación del modelo obtienen una solución del problema original a partir de otro problema simplificado.



CAPÍTULO 3.

Estado del Arte

En este capítulo se presentan los trabajos relacionados con la asignación de cargas horarias las cuales han sido una gran problemática en las instituciones educativas. La asignación de cargas horarias en la Unidad Académica Profesional Tianguistenco se lleva a cabo manualmente.

Los problemas de asignación de cargas horarias han sido probados con diferentes técnicas de optimización basadas en el cómputo evolutivo. Recordando el cómputo evolutivo consiste en un proceso basado en la evolución, con posibles soluciones a un problema, las cuales se modifican o evolucionan obteniendo cada vez mejores resultados (Holland, 1975). Por ejemplo: camino de hormigas (Algarín, 2010), algoritmos genéticos (Broca, 2016) y algoritmos meméticos. (E, OCAMPO, & BAQUERO, 2006).

Existen desarrollos en el mercado que dan solución al problema de asignación de horarios para instituciones e incluso para profesores, pero no existe alguno que solucione el problema en cuanto a la asignación de cargas horarias para estudiantes.

3.1 Trabajos relacionados a la generación de cargas horarias

Existen diferentes enfoques relacionados con la generación de cargas horarias. Cada uno de ellos utiliza un procedimiento o modelo diferente, donde generalmente se toman en cuenta las condiciones que establece la institución y/o los profesores.

En el trabajo de Caballero y Arboleda (Caballero & Arboleda, 2010) se propone el uso de algoritmos evolutivos, para poder llevar acabo la programación de horarios en universidades.

Los autores Caballero, Arboleda consideran en su problema a: profesores, aulas, grupos, unidades de aprendizaje, días, o períodos disponibles, para organizar y distribuir los horarios de clases utilizando la disponibilidad horaria de cada profesor.

El autor hace referencia a que este problema se presenta cada semestre, teniendo que realizar manualmente la generación de carga horaria, considerando las restricciones obligatorias y deseables.

3.1.1 Programación entera

Existe una investigación de 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. 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. Al inicio, la programación horaria era generada por un equipo conformado por 3 profesionales, los cuales demoraban en promedio un mes para obtener la programación final.

La generación de carga horaria se trata de un problema de minimización. Dentro de las restricciones tomadas en cuenta para este trabajo son:

Restricciones duras:

- 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.
- No deben existir topes de horarios entre cursos de un mismo semestre.

Restricciones suaves:

- 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.

Resultados alcanzados:

La herramienta asegura que los horarios generados cumplen con las restricciones impuestas y entrega la flexibilidad de poder realizar asignaciones horarias parciales.

3.1.2 Algoritmo de búsqueda tabú

Como se menciona anteriormente, existen diferentes técnicas para enfrentar la generación de cargas horaria. El algoritmo tabú es implementado en el trabajo de Martínez (Martínez, 2012) en su investigación que lleva el nombre de Algoritmo basado en tabú search para el problema de asignación de Horarios de Clases.

Este trabajo consiste en asignar lo mejor posible las clases para los estudiantes en bloques de tiempo semanales y en determinadas aulas, bajo ciertas restricciones. Se trabaja con la utilización de métodos heurísticos y metaheurísticas para tratar de encontrar buenas soluciones en tiempos computacionales razonables.

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. Toma como restricciones lo siguiente:

Restricciones duras:

- 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.

Restricciones blandas:

- 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.

Resultados alcanzados:

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

3.1.3 Algoritmo memético

En el trabajo de Granada (Granada, Toro, & Franco, 2006) el principal objetivo es elegir un salón y un lapso de tiempo 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.

En este trabajo se implementa un algoritmo memético, para llevar a cabo la solución a este problema. Se propuso la modificación de un algoritmo planteado por Chu-Beasley, el cual consiste en mantener constante el tamaño de la población de manera que solo se vayan reemplazando los valores.

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.

Resultados alcanzados:

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.

3.1.4 Ritmos cognitivos

En este trabajo se modela la programación de horarios escolares, mediante el algoritmo genético NSGA-II, el cual es un algoritmo multiobjetivo con elitismo, el cual ha sido adaptado para la solución del este problema. Sus restricciones son:

- 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.

Resultados alcanzados:

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.

En la tabla 4 se muestran trabajos que realizan la asignación de horarios, para profesores e instituciones, por lo que se demuestra que en el estado del arte no se encuentran trabajos enfocados a la generación de cargas horarias para estudiantes.

Tabla 4. Trabajos relacionados a la generación de cargas horarias.

Trabajo	Investigación	Técnica empleada	Aplicación
(Hernández et al., 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	Programación Entera	Institución
(Martínez, 2012)	Algoritmo basado en tabú search para el problema de asignación de Horarios de Clases.	Tabú search	Institución
(Granada et al., 2006)	Programación óptima de horarios de clase usando un algoritmo memético	Algoritmo Memético	Institución
(Suárez, 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	Ritmos Cognitivos	Profesores

3.2 Trabajos relacionados con optimización multiobjetivo

En el trabajo de (Peñuela Meneses & Granada Echeverri, 2007) pretenden explicar el funcionamiento básico de uno de los más conocidos algoritmos multiobjetivo propuesto en la literatura especializada, con el fin de presentar una herramienta matemática que pueda ser usada para resolver problemas complejos de optimización multiobjetivo.

La principal diferencia con un modelo mono-objetivo radica en la presencia de un grupo de funciones que deben ser optimizadas simultáneamente. Para este trabajo se utiliza el proceso de selección (por torneo), cruzamiento y mutación, de igual manera se utiliza un operador de apilamiento el cual permite cuantificar el espacio alrededor de una alternativa que no se encuentra ocupada por ninguna otra solución.

Cuando se trata con problemas de optimización mono-objetivo una alternativa de solución se considera mejor que otra si produce una solución objetivo de menor valor, esto para el caso de minimización. Sin embargo, en los problemas multiobjetivo este criterio debe ser revaluado, ya que se considera al mismo tiempo funciones de minimizar y funciones de maximizar, un ejemplo de minimización y maximización se muestra en la figura 4.

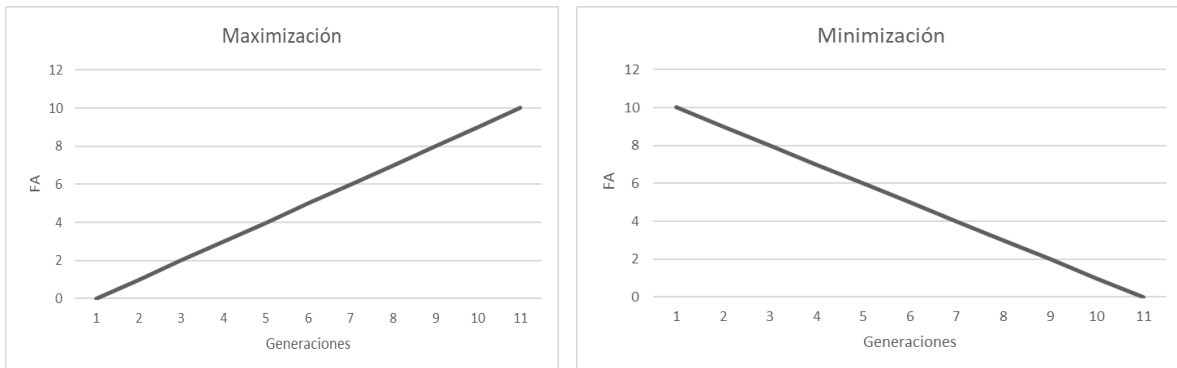


Figura 5. Se muestra en la imagen de la izquierda un ejemplo de minimización, y de lado derecho se muestra la maximización.

3.3 Resumen del capítulo

En este capítulo se mencionaron los trabajos relacionados al problema de la asignación de cargas horarias, la cual tiene una suma importancia para poder organizar los tiempos de la institución, así como de los profesores, de acuerdo con el estado del arte.

De igual manera se menciona el trabajo realizado por Hernández, quién realiza la 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.

Otra técnica mencionada es Algoritmo Tabú; el cual es implementado en el trabajo de Martínez, su investigación lleva por nombre Algoritmo basado en Tabú search para el problema de asignación de horarios de clases.

Se hace mención del trabajo de Granada que realiza su trabajo con algoritmos meméticos, para el cual se utiliza la investigación de Chu-Beasley, con la finalidad de encontrar la mejor manera de programar una secuencia de periodos, los cuales mantengan un orden para la organización de horarios.

En el trabajo de Suárez se implementan Ritmos Cognitivos. Con esta investigación se pretende realizar buena propuesta de horario escolar, es enfocada en una escuela pública colombiana, en dónde se requiere la asignación de salones y docentes. Para este trabajo se modela la programación de horarios escolares, mediante un algoritmo genético NSGA-II, el cual es un algoritmo multiobjetivo con elitismo.

Se hace mención de los trabajos relacionados con la optimización multiobjetivo, se le llama multiobjetivo o multicriterio debido a que se tiene más de un objetivo para cumplir. En el trabajo de Peñuela se presenta una herramienta matemática que pueda ser usada para resolver problemas complejos.



CAPÍTULO 4.

Método Propuesto

Recordando el problema: ¿Cómo encontrar la mejor carga horaria que satisfaga las condiciones del estudiante en tiempo real? En el presente capítulo se describe el método propuesto para obtener un modelo que permita describir la importancia que tiene la asignación de cargas horarias para estudiantes. La idea principal es obtener un modelo que sugiera una carga horaria.

4.1 Descripción del método propuesto

Se propone un método de optimización porque existe la posibilidad de poder generar diferentes cargas horarias que se satisfagan las condiciones del estudiante. El estado del arte muestra los métodos que se han utilizado para resolver la optimización cercana a este problema, empleando Algoritmos Meméticos (Cotta & Cotta, 2016), Algoritmos Tabú (Martínez, 2012), Programación entera (Hernández, Miranda P, & Rey, 2008), Programación Multiobjetivo- Multicriterio (Peñuela Meneses & Granada Echeverri, 2007). Sin embargo, los trabajos no están enfocados en el estudiante, si no en la institución o en los profesores.

El problema de la presente tesis tiene varias condiciones, por lo tanto, se plantea como un problema de optimización multiobjetivo, por lo cual se propone utilizar en específico un algoritmo genético.

Donde los datos de entrada son los horarios con las unidades de aprendizaje que ofrece la Universidad. Como se muestra en las condiciones establecidas por el estudiante mencionadas en la tabla 1.3.

De acuerdo con el avance que el estudiante tiene en cuanto a su desarrollo académico, y tomando en cuenta los horarios ofertados por la UAPT, se conforma una Lista de Unidades de Aprendizaje que se Pueden tomar (LUAP) de tamaño n .

Enseguida se mencionan los requerimientos que deben ser considerados para la configuración del algoritmo genético, para llevar acabo la asignación de cargas horarias para estudiantes.

Datos de Entrada

Se muestran ejemplos de entrada, ejemplo referente a los horarios por la UAPT en la tabla 5, referente a las condiciones en la tabla 6, referente a la carga académica se muestra en la figura 5 y referente a la Lista de Unidades de Aprendizaje que Puede tomar el estudiante en la tabla 7.

Tabla 5. Se muestra un ejemplo del horario que dio a conocer la UAPT para la Ingeniería de software en el período 2018A.

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00-8:00	Técnicas de procesamiento de imágenes	Datawarehouse	Base de datos orientada a objetos			Metodología de la investigación
8:00-9:00						
9:00-10:00						
10:00-11:00			Administración y organización de proyectos de sw			
11:00-12:00						
12:00-13:00						
13:00-14:00						

Tabla 6. Se muestran las condiciones que conforman la FA.

Condición	Abreviatura
<i>Condiciones de Tiempo por Semana</i>	(CTS)
<i>Condición de Horas Traslapadas</i>	(CHT)
<i>Condición de Unidades de Aprendizaje que Puede y Debe cursar el estudiante</i>	(CUAPDC)
<i>Condición de Sugerencia de Cargas Horarias en Tiempo Real</i>	(CSCHTR):
<i>Condición de Créditos Máximos y Mínimos para cada semestre</i>	(CCMAX) y (CCMIN)

En la figura 7 se muestra el modelo implementado en este trabajo, para la asignación de cargas horarias para estudiantes.

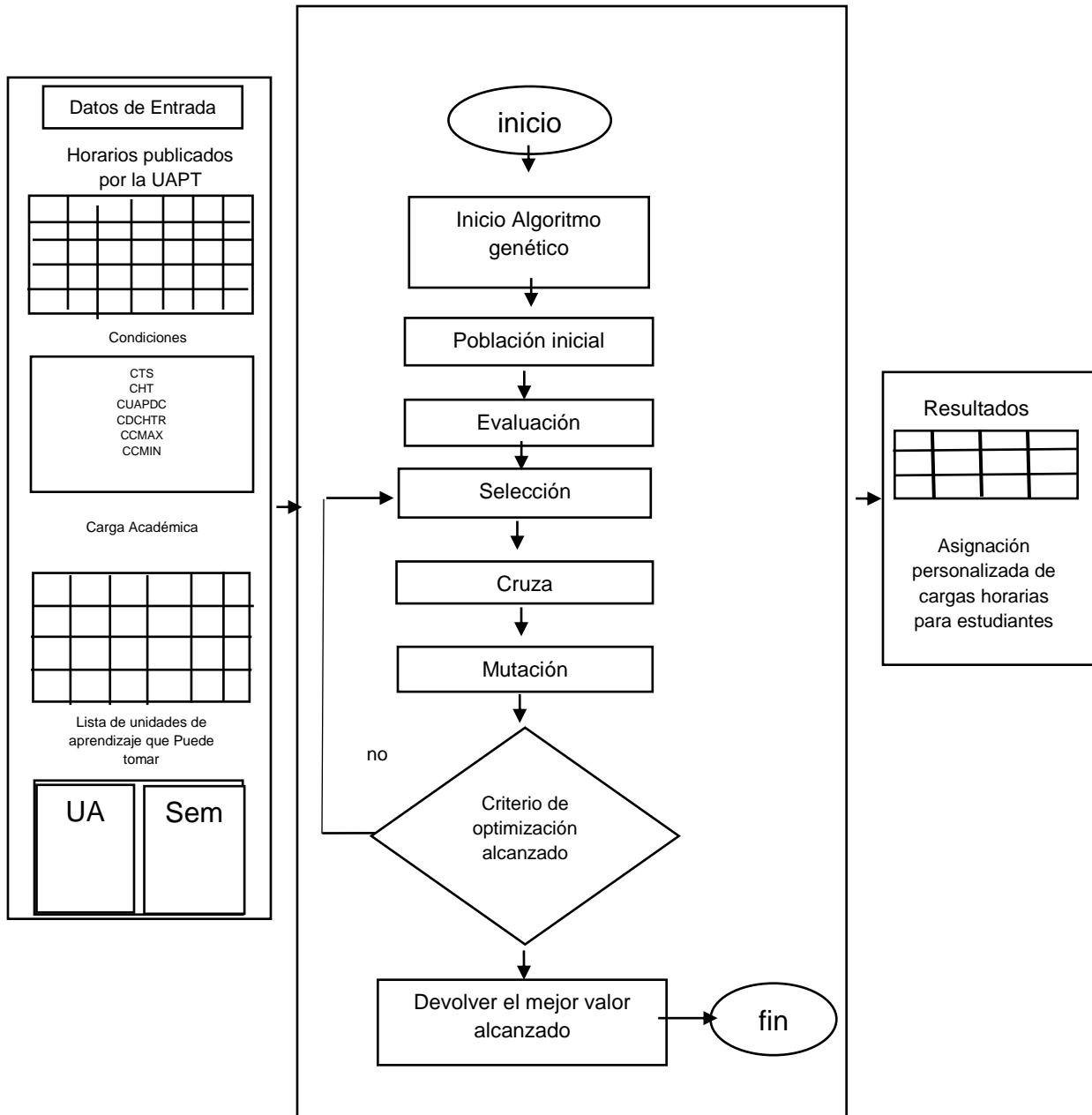


Figura 7. Modelo del método propuesto.

4.2 Codificación del individuo

La longitud del individuo depende de las n posibles unidades de aprendizaje que el estudiante puede tomar. El cual se va a determinar de acuerdo con el horario establecido por la UAPT, condiciones y la carga académica del estudiante. Donde cada gen es una unidad de aprendizaje, el cual es representado con una codificación binaria (0,1).

Dónde: 1 significa seleccionar la unidad de aprendizaje y 0 significa no seleccionar la unidad de aprendizaje

4.3 Función de aptitud

La función de aptitud busca minimizar los errores de acuerdo con el horario propuesto por el individuo.

El error calculado se determina por la suma de los errores en cada uno de los 5 criterios.

La función de aptitud se divide en condiciones, las cuales son normalizadas en un rango de 0 a 1 para poder llevar a la minimización del error.

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:

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

CHT

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

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

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 n unidades de aprendizaje de mayores semestres que puede tomar

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

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.

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

La Función de Aptitud (FA) del algoritmo genético es la siguiente:

$$FA = \alpha CTS + \beta CHT + \gamma CUAPDC + \lambda CTR + \psi CCMIN + \phi CCMAX$$

Las variables $\alpha, \beta, \gamma, \varepsilon, \lambda, \psi, \theta$, que contiene cada condición, tiene como función: dar mayor o menor peso a las condiciones, en conjunto la suma de las variables debe dar un resultado igual a 1.

4.4 Selección por torneo

El operador de selección por torneo, indica que se elige k aleatoriamente, de esto se selecciona el más alto y entonces es el valor que pasa a la siguiente generación

4.5 Operador de cruza en n puntos

El cruce en n puntos aleatorios se obtiene seleccionando la cantidad de n veces para los cruces que se hagan, se cruza en punto dado aleatoriamente.

4.6 Operador de mutación por inversión binaria

Por ser un algoritmo genético con codificación binaria, una vez obtenido el gen a mutar, se reemplazará el gen mutado, por el valor contrario.

4.7 Criterio de parada

El algoritmo puede ser detenido al alcanzar el número de generaciones alcanzadas o tiempo de ejecución específico.

4.8 Resumen del capítulo

En este capítulo se muestra el método propuesto para poder obtener el modelo que permita la asignación personalizada de cargas horarias para estudiantes.

Se describe cada una de las fases que componen el algoritmo genético, así como los operadores que son implementados en el algoritmo.

En el siguiente capítulo se muestran los experimentos realizados.



CAPÍTULO 5.

Experimentación

Recordando la hipótesis, dado que existen muchas posibilidades de horarios para cada conjunto de condiciones personalizadas del estudiante, 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 adapten al estudiante.

Si se utiliza un algoritmo de cómputo evolutivo entonces se podrán asignar cargas horarias personalizadas de acuerdo con las condiciones del estudiante.

Para este trabajo se implementan datos reales con estudiantes de la UAPT, de la ingeniería de software.

5.1 Experimento 1

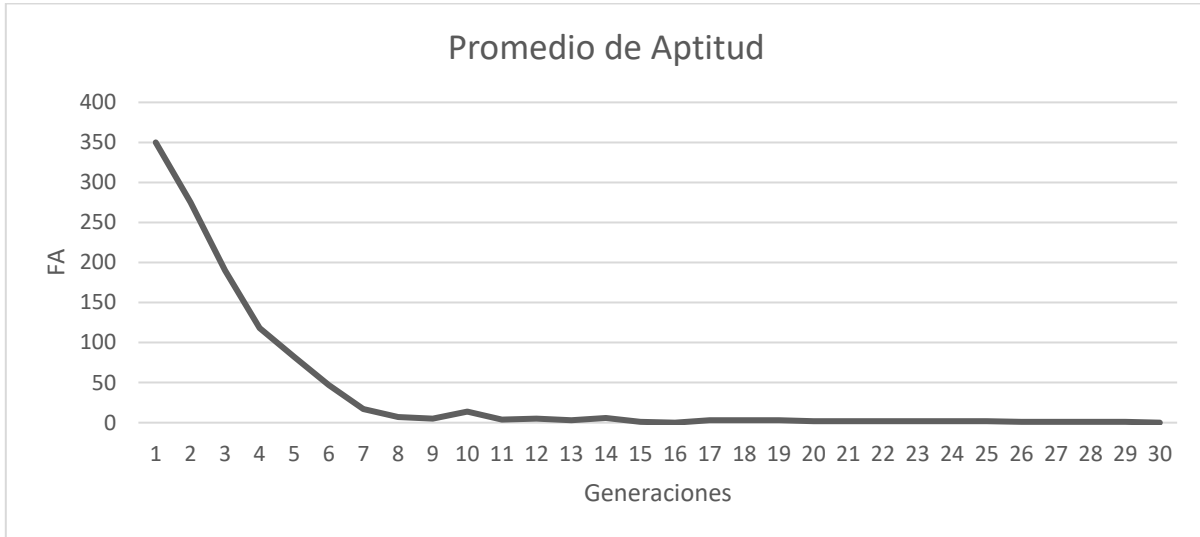
El Estudiante 1 (E1) forma parte de la comunidad estudiantil de la UAPT, cursa el 10º semestre de la ingeniería de software y tiene una situación académica irregular. Los horarios ofertados por la UAPT y de los cuales el estudiante podrá realizar su selección de unidades de aprendizaje se encuentran en el Anexo 1.

Las condiciones que establece el E1 para la selección de su carga horaria se muestran en la tabla 8, donde en la condición de tiempo real el estudiante no solicita un tiempo específico de búsqueda.

Tabla 8. Condiciones del Estudiante 1 (E1).

Condición	E1
CTS	37
CHT	0
CUAPDC	10
CCMAX	47
CCMIN	17
CSCHTR	-

El objetivo de la gráfica 1 es demostrar que el algoritmo genético está realizando adecuadamente la minimización de la Función de Aptitud (FA), cumpliendo con las condiciones del estudiante anteriormente mencionadas.

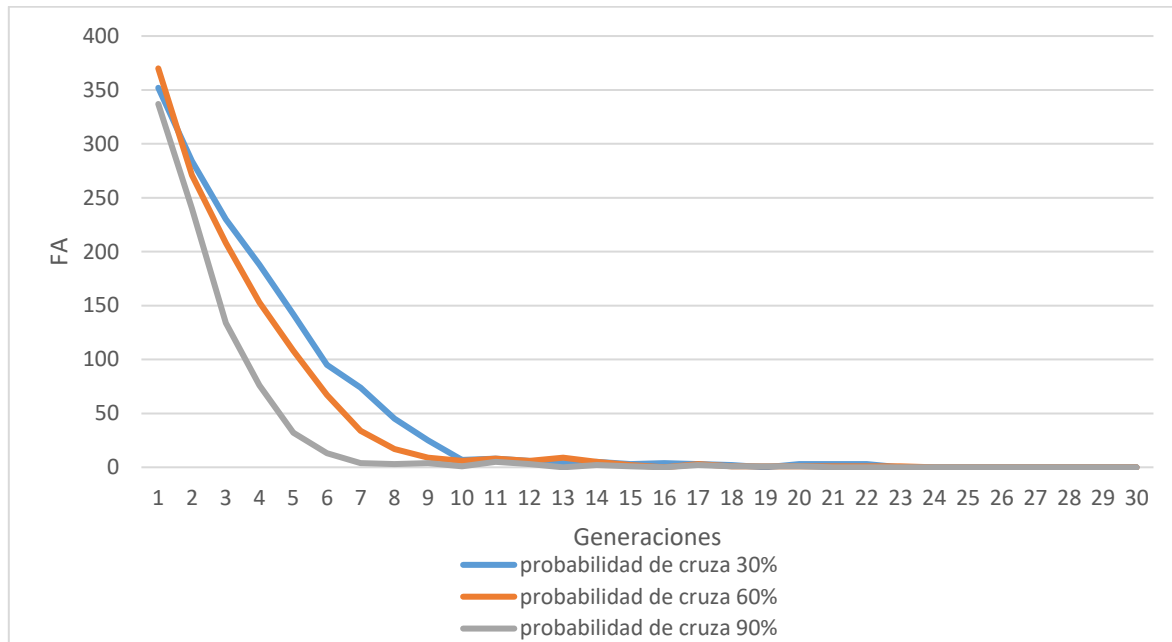


Gráfica 1. Población de 100 individuos, 30 generaciones, probabilidad de cruce 100%, probabilidad de mutación 0%

5.2 Ajuste de parámetros

Probabilidad de cruce.

Como se muestra en la gráfica 2, probabilidad de cruce de 30%, probabilidad de cruce de 60% y probabilidad de cruce de 90%. Debido a este análisis se puede decir que el mejor parámetro encontrado de probabilidad de cruce es 90%.



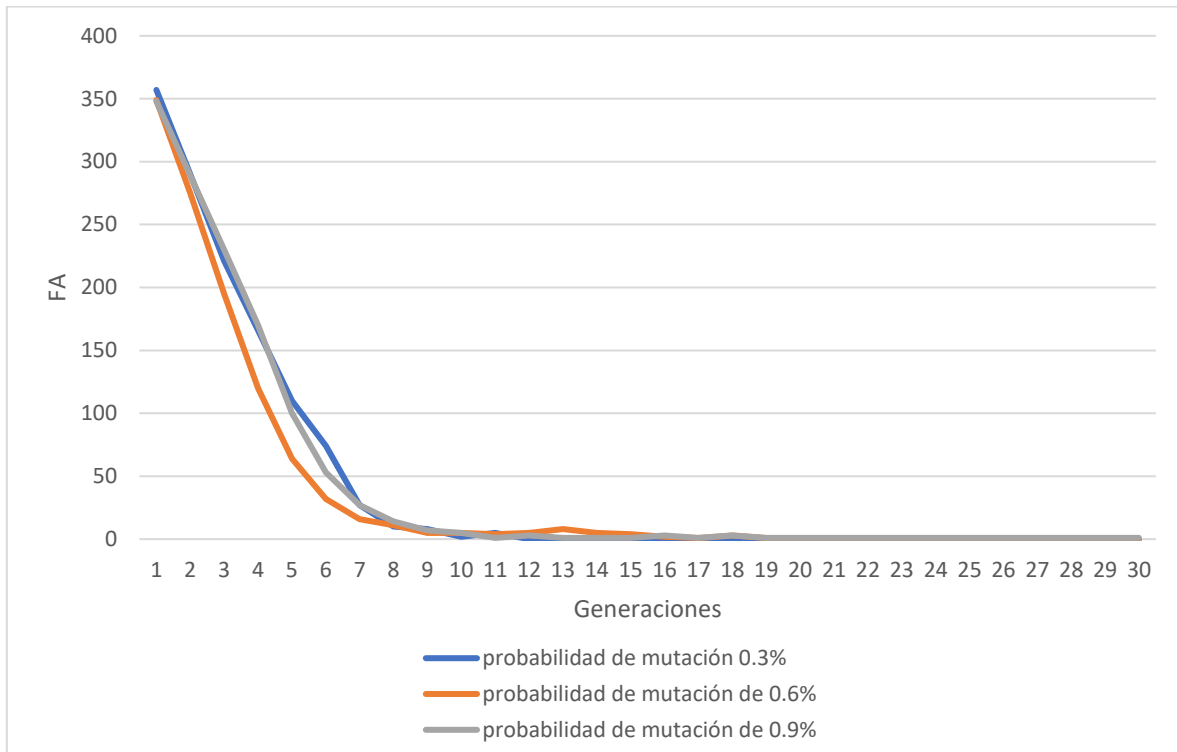
Gráfica 2. Teniendo una población de 100 individuos, 30 generaciones, se analizan distintos porcentajes de probabilidad de cruce.

Experimentación

Probabilidad de mutación.

En la gráfica 3 se muestra probabilidad de mutación de 0.3%, probabilidad de mutación de 0.6% y probabilidad de mutación de 0.9% de probabilidad de mutación.

Debido a este análisis se puede decir que el mejor parámetro encontrado de probabilidad de cruce es 0.3%.



Gráfica 3. Teniendo una población de 100 individuos, 30 generaciones, se analizan distintos porcentajes de probabilidad de mutación.

El Estudiante 1 de la ingeniería de software del 10° semestre selecciona las unidades de aprendizaje para el semestre actual, el estudiante tiene situación académica irregular, por lo que tiene que seleccionar unidades de aprendizaje de diferentes semestres, ocasionando traslapes con las unidades de aprendizaje y los horarios como se muestra en la tabla 9.

Las condiciones elegidas por el Estudiante 1 se encuentran en la tabla 8 anteriormente mencionada, enseguida se muestra el horario generado por el estudiante.

Tabla 9. Carga horaria seleccionada por el Estudiante1 (E1), en dónde se traslapan unidades de aprendizaje.

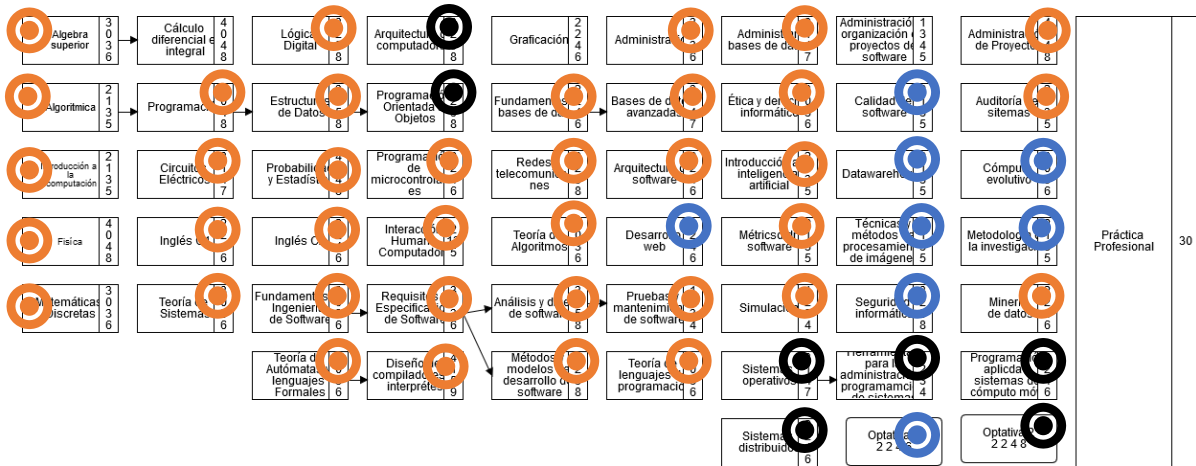
Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	Seguridad informática	Calidad de Software	Base de Datos Orientada a objetos	Seguridad Informática	Cómputo Evolutivo	Metodología de la Investigación
8:00 - 9:00		Traslape entre calidad de software y datawarehouse				
9:00 - 10:00		Datawarehouse				
10:00 - 11:00	Administración y Organización de Proyectos	Herramientas para la administración y programación de sistemas				
11:00 - 12:00					Desarrollo web	
12:00 - 13:00						
13:00 - 14:00						

La carga horaria seleccionada por el E1 tiene una FA (función de aptitud) de: 0.08

A continuación, se muestra un ejemplo de cómo el Estudiante 1 realiza la selección de su carga horaria:

Experimentación

Paso 1: Se toma en cuenta la carga académica del Estudiante 1, para poder verificar que unidades de aprendizaje le faltan por cursar



Dónde:

- Unidades de aprendizaje que aún no son seleccionadas
- Unidades de aprendizaje ya cursadas y aprobadas.
- Unidades de aprendizaje que se desea cursar en el período actual.

Experimentación

Paso 2: Se toman en cuenta los horarios que brinda la UAPT.

Tabla 10. Se muestra un horario del cual toma las unidades de aprendizaje: Administración y organización de proyectos de software, calidad de software y seguridad informática.

Unidad de Aprendizaje	Grupo	Cred	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Administración y organización de proyectos de software	s7	5	10:00-14:00					
Calidad de software	s7	5		7:00-10:00				
Datawarehouse	s7	5				11:00-14:00		
Técnicas y métodos de procesamiento de imágenes	s7	6			7:00-10:00			
Seguridad informática	s7	8	7:00-9:00			7:00-10:00		
Herramientas para la administración y programación de sistemas	s7	4		10:00-13:00				
Programación paralela	s0	6						7:00-11:00
Administración de empresas de desarrollo de software	s0	5			10:00-14:00			

Tabla 11. Se muestra un horario del cual toma las unidades de aprendizaje: Datawarehouse, Base de datos orientada a objetos.

Unidad de Aprendizaje	Grupo	Cred	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Administración y organización de proyectos de software	s8	5					10:00-14:00	
Calidad de software	s8	5				9:00-12:00		
Datawarehouse	s8	5		7:00-10:00				
Técnicas y métodos de procesamiento de imágenes	s8	6	7:00-9:00 10:00-11:00					
Seguridad informática	s8	8		10:00-12:00			10:00-12:00	
Herramientas para la administración y programación de sistemas	s8	4	11:00-14:00					
Redes neuronales y L.D.	s0	6		12:00-14:00		12:00-14:00		
Bases de datos orientada a objetos	s0	6			7:00-10:00			

Tabla 12. Se muestra un horario del cual toma las unidades de aprendizaje: metodología de la investigación, cómputo evolutivo.

Unidad de Aprendizaje	Grupo	Cred	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Metodología de la investigación	s9	5						8:00-11:00
Cómputo evolutivo	sR	6					7:00-10:00	
Práctica Profesional	s9	30		12:00-16:00				
Práctica Profesional	sX	30				11:00-15:00		

Paso 3: Se toma en cuenta la Lista de Unidades de Aprendizaje que Puede tomar (LUAP).

Tabla 13. Se muestran las unidades de aprendizaje que puede tomar el Estudiante 1 (E1).

LUAP
Administración y organización de proyectos de software
Calidad de software
Seguridad informática
Datawarehouse
Bases de datos orientada a objetos
Metodología de la investigación
Cómputo evolutivo

Tomando en cuentas las condiciones del estudiante, el algoritmo genético realiza la selección de unidades de aprendizaje que cumple con las condiciones del estudiante presentadas en la tabla 8.

Experimentación

En la tabla 14 se muestra las unidades de aprendizaje seleccionadas por el algoritmo genético de acuerdo con las condiciones del estudiante.

Tabla 14. Carga horaria seleccionada por el algoritmo genético para el Estudiante1 (E1).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	Seguridad informática	Calidad de Software	Base de Datos Orientada a objetos	Seguridad Informática	Cómputo Evolutivo	Metodología de la Investigación
8:00 - 9:00						
9:00 - 10:00						
10:00 - 11:00	Administración y Organización de Proyectos	Herramientas para la administración y programación de sistemas		Datawarehouse	Desarrollo web	
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00						

La carga horaria seleccionada por el algoritmo genético se asignó en un tiempo de 20 segundos, tiene una FA de: 0, lo cual indica que cumple con las condiciones establecidas por el estudiante.

Los mismos parámetros utilizados en el Experimento 1, son utilizados en las siguientes dos experimentaciones.

5.3 Experimento 2

El Estudiante 2 (E2) forma parte de la comunidad estudiantil de la UAPT, cursa el 8° semestre de la ingeniería de software y tiene una situación académica irregular.

Las condiciones que establece el E2 para la selección de su carga horaria se muestran en la tabla 15, donde en la condición de tiempo real el estudiante no solicita un tiempo específico de búsqueda.

Tabla 15. Condiciones del Estudiante 2 (E2).

Condición	E2
CTS	20
CHT	0
CUAPDC	5
CCMAX	42
CSCHTR	-

De acuerdo con los horarios establecidos por la UAPT y con relación a las condiciones del estudiante, se realiza la selección de unidades de aprendizaje para el E2, tal como se muestra en la tabla 16.

Tabla 16. Carga horaria seleccionada por el Estudiante 2 (E2).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00-8:00	Técnicas de proc de imágenes	Datawarehouse	Base de datos orientada a objetos			Metodología de la investigación
8:00-9:00						
9:00-10:00			Adm. De empresas			
10:00-11:00						
11:00-12:00						
12:00-13:00						
13:00-14:00						

La carga horaria seleccionada por el E2 tiene una FA de: 0, esto indica que se cumplen las condiciones del E2.

Experimentación

El algoritmo realiza la selección de unidades de aprendizaje, tal como se muestra en la tabla 17, de acuerdo con las condiciones del estudiante

Tabla 17. Carga horaria asignada por el genético para el Estudiante 2 (E2).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00-8:00	Técnicas de procesamiento de imágenes		Base de datos orientada a objetos			
8:00-9:00						
9:00-10:00						
10:00-11:00				Adm. De empresas	Datawarehouse	
11:00-12:00						
12:00-13:00						
13:00-14:00						

La carga horaria seleccionada por el algoritmo genético se asignó en 10 segundos, tiene una FA de: 0, lo cual indica que cumple con las condiciones establecidas por el estudiante.

5.4 Experimento 3

Las condiciones que establece el Estudiante 3 (E3) para la selección de su carga horaria se muestran en la tabla 18, donde en la condición de tiempo real el estudiante no solicita un tiempo específico de búsqueda.

Tabla 18. Condiciones del Estudiante 3 (E3).

Condición	E2
CTS	16
CHT	0
CUAPDC	5
CCMAX	38
CSCHTR	-

El Estudiante 3 (E3) forma parte de la comunidad estudiantil de la UAPT, cursa el 6° semestre de la ingeniería de software y tiene una situación académica irregular. El horario asignado por el estudiante se muestra en la tabla 19.

Tabla 19. Carga horaria seleccionada por el Estudiante 3 (E3).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00-8:00	Estructuras de datos	Cálculo	Arquitectura de computadoras			
8:00-9:00						
9:00-10:00	Compiladores e intérpretes					
10:00-11:00						
11:00-12:00						
12:00-13:00						
13:00-14:00						

La carga horaria seleccionada por el E3 tiene una FA de: 0, esto indica que se cumplen las condiciones del E3.

En la tabla 20 se muestra la selección de cargas horarias que realiza el algoritmo genético, de acuerdo con las condiciones establecidas por el estudiante.

Tabla 20. Carga horaria seleccionada por el genético para el Estudiante 3 (E3).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00-8:00			Arquitectura de computadoras		Compiladores e intérpretes	Química
8:00-9:00						
9:00-10:00		Estructuras de datos		Cálculo		
10:00-11:00						
11:00-12:00						
12:00-13:00						
13:00-14:00						

La carga horaria seleccionada por el algoritmo genético se asignó en 10 segundos, tiene una FA de: 0, lo cual indica que cumple con las condiciones establecidas. El algoritmo selecciona una carga horaria alternativa que satisface las necesidades del E3.

En la comparación que se realiza entre humano(estudiante) y genético, se mostró la asignación por el algoritmo genético al humano para que evaluará al genético, diciendo si realmente le es útil la sugerencia que se realiza. En dónde la respuesta de los tres estudiantes(E1,E2,E3) es que, podrían haber utilizado la asignación de cargas horarias que realiza el algoritmo, debido a que cumple con sus condiciones.

5.5 Resumen del capítulo

En este capítulo se describieron los parámetros utilizados en los experimentos realizados, tomando como ejemplos estudiantes de la UAPT específicamente de la ingeniería de software. Los horarios seleccionados por los estudiantes (E1, E2, E3) fueron comparados con los seleccionados por el algoritmo genético, utilizando los mismos parámetros para las experimentaciones, por lo cual se verifica que el algoritmo genético está funcionando correctamente, teniendo buenas soluciones con diferentes condiciones de los estudiantes.



CAPÍTULO 6.

Conclusiones

En este trabajo se implementó un algoritmo genético, para la asignación personalizada de cargas horarias para estudiantes. Los algoritmos genéticos están basados en la solución de problemas mediante la búsqueda y optimización, teniendo como pasos la población, cruza y mutación.

El problema que enfrenta el estudiante de asignar una carga horaria es muy importante, ya que actualmente no se cuenta con una herramienta que realice este trabajo de manera automática.

Con el presente trabajo se demostró que con el uso de algoritmos genéticos y optimización multiobjetivo es posible asignar cargas horarias de manera personalizada para estudiantes.

El problema de cargas horarias presenta 5 condiciones, las cuales forman parte de la función de aptitud. La función de aptitud minimiza los errores de acuerdo con el horario propuesto.

Los horarios obtenidos con el algoritmo genético fueron comparados con los horarios realizados por los estudiantes. El resultado fue que los horarios asignados con el algoritmo genético asignan una carga horaria mejor organizada que la de los estudiantes, donde se toman en cuenta las condiciones de éste.

Por lo cual se concluye que las cargas horarias generadas con el algoritmo genético resultan ser mejores a los asignados por el humano, de acuerdo con las condiciones académicas que se tomaron en cuenta.

6.1 Aportaciones

Se realizó la comparación de la carga horaria asignada por el humano (estudiante) con la carga horaria asignada por el algoritmo genético.

Se realizó la entrega de la asignación de cargas horarias a los estudiantes seleccionados para la experimentación del algoritmo, en general a los 3 estudiantes les agrado su carga horaria.

6.2 Trabajo futuro

Aplicar esta herramienta para todos los estudiantes de la UAPT, ya que esta herramienta puede ser funcional para cualquier estudiante, pero hay que implementar este modelo a una aplicación móvil que sea de fácil acceso.



Apartado de anexos

Anexo A

Horarios publicados por la UAPT para el período 2018A, para la ingeniería de software.

Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3
Administración y organización de proyectos de software	s7	5	10:00-14:00						•		
Calidad de software	s7	5		7:00-10:00					•		
Datawarehouse	s7	5				11:00-14:00					
Técnicas y métodos de procesamiento de imágenes	s7	6			7:00-10:00						

Anexos

Seguridad informática	s7	8	7:00-9:00				7:00-10:00			•		
Herramientas para la administración y programación de sistemas	s7	4		10:00-13:00						•		
Programación paralela	s0	6						7:00-11:00				
Administración	s0	5			10:00-14:00							
Unidad de Aprendizaje												
Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3	
Administración y organización de proyectos de software	s8	5					10:00-14:00					
Calidad de software	s8	5				9:00-12:00						
Datawarehouse	s8	5		7:00-10:00					•	•		
Técnicas y métodos de procesamiento de imágenes	s8	6	7:00-9:00 10:00-11:00								•	
Seguridad informática	s8	8		10:00-12:00			10:00-12:00					
Herramientas para la administración y programación de sistemas	s8	4	11:00-14:00									
Redes neuronales y lógica difusa	s0	6		12:00-14:00		12:00-14:00						
Bases de datos orientado a objetos	s0	6			7:00-10:00				•	•		
Administración	s0	5			11:00-15:00						•	
Unidad de Aprendizaje												
Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3	
Bases de datos avanzadas	s6			9:00-11:00		9:00-11:00						
Arquitectura de software	s6				9:00-11:00	7:00-9:00						
Desarrollo web	s6			7:00-9:00			10:00-12:00		•			
Pruebas y mantenimiento de software	s6		7:00-9:00				8:00-9:00					
Teoría de lenguajes de programación	s6		10:00-11:30		7:30-9:00							

Anexos

Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3
Metodología de la investigación	s9	5						8:00-11:00	•	•	
Cómputo evolutivo	sR	6					7:00-10:00		•		
Práctica Profesional	s9	30		12:00-16:00							
Práctica Profesional	sX	30				11:00-15:00					
Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3
Algorítmica	sR	6						8:00-11:00			
Arquitectura de computadoras	s3	6		9:00-11:00	7:00-9:00						•
Estructura de datos	s3	8	7:00-9:00	9:00-11:00							•
Progra de microcontroladores	s3	6	12:00-14:00		9:00-11:00						
Requisitos y especificación de software	s3	6				7:00-10:00					
Diseño de compiladores e intérpretes	s3	9	10:00-12:00				7:00-9:00				•
Interacción humano computadora	S3	5		11:00-14:00							
Inglés 7	s3	6				10:00-12:00	9:00-11:00				
Inglés c2		6					13:00-17:00				
Unidad de Aprendizaje	Grupo	Créd	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	E1	E2	E3
Programación	s2	8	11:00-13:00		9:00-12:00						
Cálculo diferencial e integral	s2	8	7:00-9:00			9:00-11:00					•
Circuitos electrónicos	s2	7		7:00-9:00			7:00-9:00				
Teoría de algoritmos	s2	6		9:00-12:00							
Inglés 5	s2	6		12:00-14:00			9:00-11:00				
Teoría de sistemas	s2	6			7:00-8:30	7:00-8:30					
Química	sR	5						8:00-12:00			•

Anexo B

Se presenta el mapa curricular de la Ingeniería de Software, mismo que pertenece a la Unidad Académica Profesional Tianguistenco.

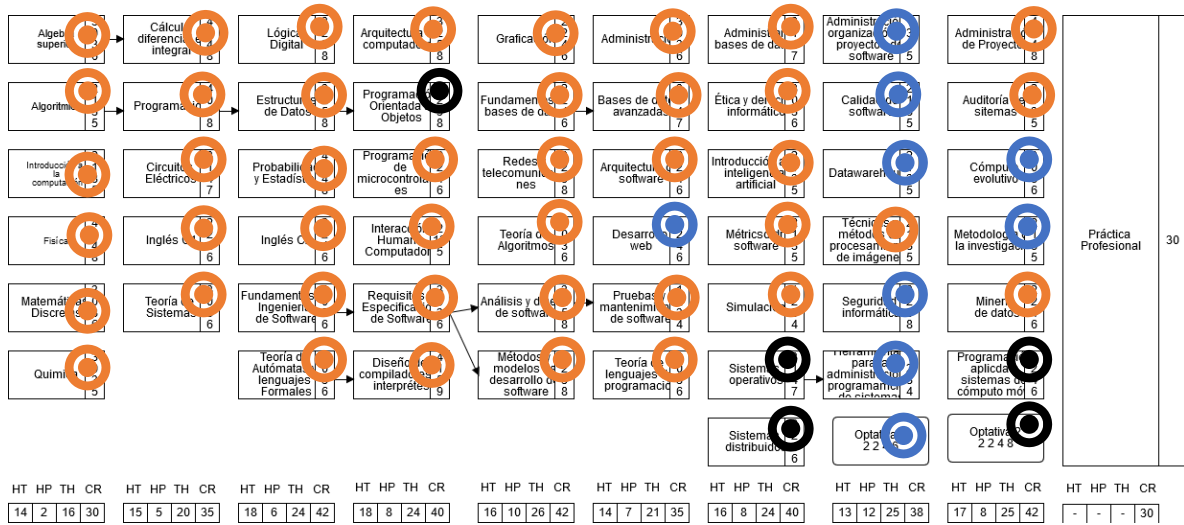
ALGEBRA SUPERIOR E _S	CÁLCULO DIFERENCIAL E INTEGRAL E _S	LÓGICA DIGITAL	ARQUITECTURA DE COMPUTADORAS	GRAFICACIÓN	ADMINISTRACIÓN	ADMINISTRACIÓN DE BASES DE DATOS	ADMINISTRACIÓN Y ORGANIZACIÓN DE PROYECTOS DE SOFTWARE	ADMINISTRACIÓN DE PROYECTOS	
ALGORITMICA	PROGRAMACIÓN E _S	ESTRUCTURAS DE DATOS	PROGRAMACIÓN ORIENTADA A OBJETOS	FUNDAMENTOS DE BASES DE DATOS	BASES DE DATOS AVANZADAS	ÉTICA Y DERECHO INFORMÁTICO	CALIDAD DEL SOFTWARE	AUDITORIA DE SISTEMAS	
INTRODUCCIÓN A LA COMPUTACIÓN	CIRCUITOS ELÉCTRICOS E _S	PROBABILIDAD Y ESTADÍSTICA E _L	PROGRAMACIÓN DE MICROCONTROLADORES	REDES Y TELECOMUNICACIONES	ARQUITECTURA DE SOFTWARE	INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL	DATAWAREHOUSE	CÓMPUTO EVOLUTIVO	
FISICA	INGLÉS C1 E _L	INGLÉS C2 E _L	INTERACCIÓN HUMANO COMPUTADORA	TEORÍA DE ALGORITMOS	DESARROLLO WEB	MÉTRICAS DE SOFTWARE	TECNICAS Y MÉTODOS DE PROCESAMIENTO DE BASES DE DATOS	METODOLOGÍA DE LA INVESTIGACIÓN C B	
MATEMÁTICAS DISCRETAS	TEORÍA DE SISTEMAS	FUNDAMENTOS DE INGENIERÍA DE SOFTWARE	REQUISITOS Y ESPECIFICACIÓN DE SOFTWARE	ANÁLISIS Y DISEÑO DE SOFTWARE	PRUEBAS Y MANTENIMIENTO DE SOFTWARE	SIMULACIÓN	SEGURIDAD INFORMÁTICA	MINERÍA DE DATOS	
	TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES	DISEÑO DE COMPILADORES E INTERPRETES	MÉTODOS Y MODELOS DE DESARROLLO DE SOFTWARE	TEORÍA DE LENGUAJES DE PROGRAMACIÓN	SISTEMAS OPERATIVOS	HERRAMIENTAS PARA LA ADMINISTRACIÓN Y PROGRAMACIÓN DE SISTEMAS	PROGRAMACIÓN APLICADA A SISTEMAS DE CÓMPUTO MÓVIL		
					SISTEMAS DISTRIBUIDOS	OPTATIVA 1	OPTATIVA 2		PRÁCTICA PROFESIONAL

HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR	HT HP TH CR
14 2 16 30	15 5 20 35	18 6 24 42	16 8 24 40	16 10 26 42	14 7 21 35	16 8 24 40	13 12 25 38	17 8 25 42	-- -- -- 30	-- -- -- 30
SIMBOLOGÍA	E _S Común entre: Ingeniería de Plásticos, Ingeniería de Producción Industrial, Ingeniería de Software y Seguridad Ciudadana E _L Equivalente entre: Ingeniería de Plásticos e Ingeniería de Producción Industrial E _R Equivalente entre: Ingeniería de Plásticos, Ingeniería de Producción Industrial e Ingeniería de Software y Seguridad Ciudadana		NÚCLEO BÁSICO OBLIGATORIAS CURSAR Y ACREDITAR 12 UA 34HT 10HP 44TH 78CR	NÚCLEO SUSTANTIVO OBLIGATORIAS CURSAR Y ACREDITAR 20 UA 78HT 23HP 108TH 185CR	NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 15 UA 25HT 19HP 74TH 96CR	NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 12 UA PARA CUBRIR 4HT, 4P, 8TH, 12CR	NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 15 UA PARA CUBRIR 28HT, 23HP, 62TH, 111CR	TOTAL DEL NÚCLEO BÁSICO 12 UA PARA CUBRIR 34HT, 10HP, 44TH, 78CR TOTAL DEL NÚCLEO SUSTANTIVO 20 UA PARA CUBRIR 78HT, 23HP, 108TH, 185CR TOTAL DEL NÚCLEO INTEGRAL 15 UA PARA CUBRIR 28HT, 23HP, 62TH, 111CR	TOTAL DEL PLAN DE ESTUDIOS UA OBLIGATORIAS 53 MÁS 1 ACTIVIDAD ACADÉMICA (PRÁCTICA PROFESIONAL) UA OPTATIVAS 2 DE 8 UA A ACREDITAR 55 MÁS 1 ACTIVIDAD ACADÉMICA (PRÁCTICA PROFESIONAL) CRÉDITOS 374	

Anexo C

Seguimiento académico de Estudiante 1.

El Estudiante 1 tiene el avance académico mostrado en la siguiente imagen.

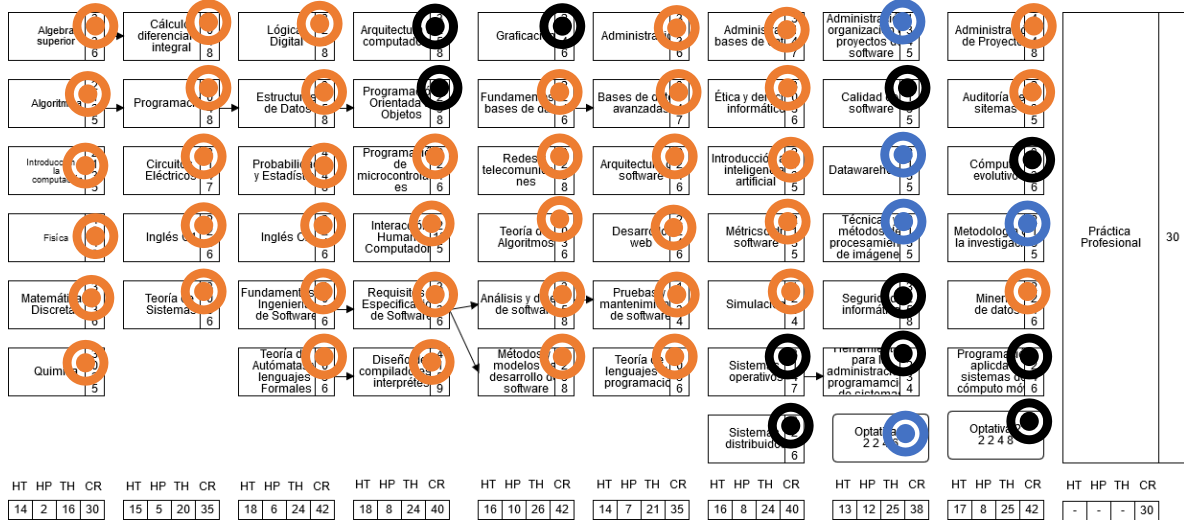


Dónde:

- Unidades de aprendizaje que aún no son seleccionadas
- Unidades de aprendizaje ya cursadas y aprobadas.
- Unidades de aprendizaje que son cursadas en el período actual.

Seguimiento académico del Estudiante 2.

El Estudiante 2 tiene el avance académico mostrado en la siguiente imagen.

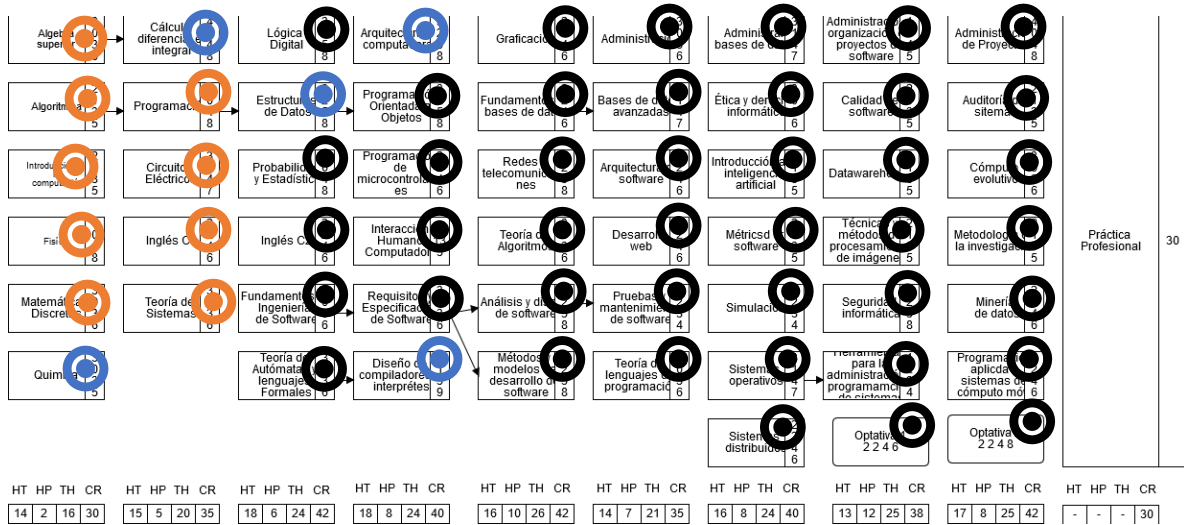


Dónde:




- Icono negro: Unidades de aprendizaje que aún no son seleccionadas
- Icono naranja: Unidades de aprendizaje ya cursadas y aprobadas.
- Icono azul: Unidades de aprendizaje que son cursadas en el período actual.

Seguimiento académico del Estudiante 3.

El Estudiante 3 tiene el avance académico mostrado en la siguiente imagen.



Dónde:

-  Unidades de aprendizaje que aún no son seleccionadas
-  Unidades de aprendizaje ya cursadas y aprobadas.
-  Unidades de aprendizaje que son cursadas en el período actual.



Bibliografía

- Pulgarín, R. S. (2016). Aplicación de optimización multiobjetivo con algoritmos genéticos a un problema de electrificación rural, 61–71.
- Aguilar, L. de M. O. (n.d.). Comparativa de Algoritmo Genético, Memético y SI para el diseño de horarios de una Institución Educativa Lucero de Montserrat Ortiz Aguilar, Juan Martín Carpio Valadez, Héctor José Puga Soberanes, *Guanajuato*, 1–6.
- Ahmed, L. N., Özcan, E., & Kheiri, A. (2015). Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications*, 42(13), 5463–5471. <https://doi.org/10.1016/J.ESWA.2015.02.059>

- Ahumada, J. A. A. (2014). Generación de Horarios Académicos en INACAP utilizando Algoritmos Genéticos, 95.
- Alvarado, 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.
- Álvarez, L. D. (2005). Seguridad en informática (Auditoría de sistemas). Universidad Iberoamericana, 1, 1–42.
- Andaluz, A. M. (2004). Algoritmos Evolutivos y Algoritmos Genéticos. Universidad Carlos III de Madrid, 1–14. Retrieved from <http://www.it.uc3m.es/~jvillena/irc/practicasyestudios/aeag>
- Araujo, K., & Martuccelli, D. (2010). La individuación y el trabajo de los individuos. *Educação e Pesquisa*, 36(spe), 77–91. <https://doi.org/10.1590/S1517-97022010000400007>
- Botero, Angélica María Ramírez, Julio Hernando Vargas, C. A. A. M. (2014). Método basado en programación genética para la solución simbólica de ecuaciones diferenciales, 26(1), 65–75.
- Broca, J. M. (2016). Algoritmos genéticos en la generación de horarios escolares. Atizapán de Zaragoza.
- Caballero, J. M. M., & Arboleda, C. P. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Revista Educación En Ingeniería*, 5(9), 140–149. <https://doi.org/10.26507/REI.V5N9.15>
- Canaria, S., Newton, I., & Matemáticas, D. P. De. (2009). *Números*. (M. D. Hugo Alfonso, Ed.) (Sociedad C, Vol. 71). Canaria.
- Caraballo, I., García-Mateo, C., Capdevila, C., & Capdevila, C. (2010). Diseño de redes neuronales con aprendizaje combinado de retropropagación y búsqueda aleatoria progresiva aplicado a la determinación de austenita retenida en aceros TRIP. *Revista de Metalurgia*, 46(6), 499–510. <https://doi.org/10.3989/revmetalmadrid.0924>
- Castrillón, O. D. (2014). ¿¿Combinación??n entre algoritmos Genéticos y Aleatorios para la programación??n de horarios de clases basado en ritmos cognitivos. *Información Tecnológica*, 25(4), 51–62. <https://doi.org/10.4067/S0718-07642014000400008>
- Chile, P. V.-A. del I. de I. de, & 1997, U. (1997). Optimización mediante algoritmos genéticos. *Researchgate.Net*, 11. Retrieved from https://www.researchgate.net/profile/Pablo_Estevez/publication/228708779_Optimizacion_Mediante_Algoritmos_Geneticos/links/0912f51111f82b2a61000000.pdf

- Coello, C., & Zacatenco, C. (2005). Introducción a la optimización multiobjetivo usando metaheurísticas. Retrieved from <http://www.academia.edu/download/40378699/Coello2.pdf>
- Cortez Vásquez, A., Rosales Gerónimo, G., Naupari Quiroz, R., & Vega Huerta, H. (2010). Sistema de apoyo a la generación de horarios basado en algoritmos genéticos. *Revista de Investigación de Sistemas e Informática*, 7(1), 37–55. Retrieved from <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3264/2729>
- Cruz, C. P. de la, & Rodríguez, J. R. (2011). Un algoritmo genético para un problema de horarios con restricciones especiales. *Revista de Matemática: Teoría y Aplicaciones*, 18(2), 215–229. Retrieved from <http://www.redalyc.org/articulo.oa?id=45326930002>
- Cuartas Torres, B. A. (2009). Metodología para la optimización de múltiples objetivos basada en ag y uso de preferencias. *UNIVERSIDAD NACIONAL DE COLOMBIA*, 113. Retrieved from <http://www.bdigital.unal.edu.co/2237/%5Cnhttp://www.bdigital.unal.edu.co/2237/1/43908352.2009.pdf>
- Cubillos, M., Pardo, E., & Salas, R. (2013). Problema Del School Timetabling Y Algoritmos Genéticos: Una Revisión. *Vínculos*, 10(2), 259–276. Retrieved from <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/6478>
- De, E. (2015). Aspiración Probabilístico-Aplicada a La Generación De Horarios Escolares Tabú Search With Aspiration Criterion for the Timetabling Problem, 22(1), 153–177.
- Eiben, A. E., & Smith, J. E. (2004). *Introduction to Evolutionary Computing*. (T. N. Leiden Center for Natural Computing Leiden University Leiden, Ed.), New York (Second Edition, Vol. 2). London. <https://doi.org/10.1162/evco.2004.12.2.269>
- Fernández, R. (2005). Sistemas En Tiempo Real: *Revista de Los Estudios de Informática de La UJI*, (1900), 1–12. Retrieved from [http://rei.uji.es/index.php?journal=rei&page=article&op=viewFile&path\[\]=10&path\[\]=9](http://rei.uji.es/index.php?journal=rei&page=article&op=viewFile&path[]=10&path[]=9)
- Flores Pichardo, M. (2001). Revisión de Algoritmos Genéticos Aplicados al Problema de la Programación de Cursos Universitarios. *Programación Matemática y Software*, 3(1), 17. Retrieved from <http://revistas.uaem.mx/index.php/progmat/article/view/237%0Ahttp://revistas.uaem.mx/index.php/progmat/article/view/237/167>
- Flores Brau, E., Monteverde, J. a, Salazar, N. F., Cadena, E., & Lizárraga, C. a. (2007). Experimentos con Algoritmos Genéticos para resolver un problema real de. *Sistemas, Cibernética e Informática*, 1(2), 42–46.

- García Gutiérrez, J. A. (2011). Análisis E Implementación De Algoritmos Evolutivos Para La Optimización De Simulaciones En Ingeniería Civil., 90.
- Gestal, M., Rivero, D., Rabuñal, J., Dorado, J., Coruña, A. P.-A., & 2010, U. (2010). Introducción a los algoritmos genéticos y la programación genética. Galeon.Com. Retrieved from <http://www.galeon.com/dantethedestroyer/algoritmos.pdf>
- Gestal, M. (2010). Introducción a algoritmos genéticos, y su aplicación. Retrieved from <http://www.galeon.com/dantethedestroyer/algoritmos.pdf>
- Glover, F., Melián, B., & Tabú, B. (2003). Búsqueda Tabú. *Inteligencia Artificial Revista Iberoamericana de Inteligencia Artificial*. No, 19, 29–48. Retrieved from <http://www.redalyc.org/pdf/925/92571902.pdf>
- Granada, M., Toro, E. M., & Franco, J. F. (2006). Programación óptima de horarios de clase usando un algoritmo memético. *Scientia et Technica*, XII (30), 255–260. Retrieved from <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/6531>
- 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. *Control*, 121–141.
- Johnston, J. D. (2000). Aplicación De Algoritmo Genético Para La Asignación De Carga Académica En Instituciones De Educación Superior. *Nuevo León*, 100. Retrieved from <http://eprints.uanl.mx/7603/1/1020130069.PDF>
- Levy, T. S., & Amaya, A. (2015). Desnutrición y obesidad: doble carga en México desnutrición y obesidad: doble carga en México resumen. *Castellanos*, 16(5), 1607–6079. <https://doi.org/ISSN:1607-6079>
- Lücken, C. Von. (2004). Algoritmos evolutivos para optimización multiobjetivo: Un estudio comparativo en un ambiente paralelo asíncrono. In *X Congreso Argentino de ...* Retrieved from <http://sedici.unlp.edu.ar/handle/10915/22476>
- Marcela Astudillo Moya. (2012). *Fundamentos De Economía*. México. Retrieved from <http://ru.iiec.unam.mx/2462/1/FundamentosDeEconomiaSecuenciaCorrecta.pdf>
- Martí Cunquero, R. (2007). Algoritmos Heurísticos en Optimización Combinatoria. *Mdereg1-Fercarpetass.Googlecode*. ..., 1–27. Retrieved from <http://yalma.fime.uanl.mx/~roger/work/teaching/mecbs5122/1-Introduction/Intro-by-RafaMarti.pdf>
- Martínez Morales, J., & Ortega Aguirre, A. (2011). La problemática actual de la deserción escolar, un análisis desde lo local, 1–112. Retrieved from <http://www.eumed.net/libros-gratis/2011a/906/>

- Martínez, A. A. (2012). Algoritmo basado en tabú search para el problema de asignación de horarios de clases. Universidad de Carabobo, 11. Retrieved from <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&act=8&ved=0ahUKEwjAgJTuhq7bAhX5GzQIHbQYAbkQFgggMAE&url=http%3A%2F%2Faraute.facyt.uc.edu.ve%2Fsites%2Fdefault%2Ffiles%2Farticulos%2FAlgoritmo-basado-en-Tabu-Search-para-el-problema-d>
- Mejía Caballero, J., & Paternina Arboleda, C. (2010). Asignación de horarios de clases universitarias mediante algoritmos evolutivos. *Revista Educación En Ingeniería*, 9(February), 140–149. <https://doi.org/10.1017/CBO9781107415324.004>
- Menéndez, E. F. M. (1999). Algoritmos evolutivos aplicados a la generación de horarios para colegio. Quito, 1–151.
- Mercedes, M., Mieles, B., Lisett, K., & Montero, K. Metodología basada en el método heurístico de polya para el aprendizaje de la resolución de problemas matemáticos Based Methodology in Polya's Heuristic method for the learning of solving mathematical problems, 10(2), 7–19. Retrieved from <http://ojs.uac.edu.co/index.php/escenarios/article/viewFile/214/198>
- Moscato, P., & Cotta, C. (2003). An Introduction to Memetic Algorithms. *Revista Iberoamericana de Inteligencia Artificial*, 7, 131–148. <https://doi.org/10.4114/ia.v7i19.721>
- Obando Paredes, E. D. (2017). Algoritmos genéticos y PSO aplicados a un problema de generación distribuida. *Scientia et Technica*, 22(1), 15. <https://doi.org/10.22517/23447214.14301>
- Ochoa, G. Introducción a la Computación Evolutiva y la Morfogénesis Artificial. Maths.Stir.Ac.Uk. Retrieved from <http://www.maths.stir.ac.uk/~goc/papers/GOchoaIntroCompEv.pdf>
- Páez, J. J. M. (2001). Conceptos básicos de programación genética. Universidad Nacional de Colombia, (Revista de la Facultad de Medicina), 110–114.
- 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*, XIII (35), 175–180. <https://doi.org/http://dx.doi.org/10.22517/23447214.5393>
- Pinilla, J. A., Ingeniería, J. O. C.-, & 2015. (n.d.). Una revisión de la literatura Multiobjective optimization in biofuel supply chain management. A review of the literature. Scielo.Org.Co. Retrieved from http://www.scielo.org.co/scielo.php?pid=S0121-750X2015000100004&script=sci_arttext&tlng=es

- Ramón, F., Soto, M. D., & Garcillán, J. J. (n.d.). Optimización multicriterio en el contexto de la programación matemática.
- Ramos Milla, F. R. (2012). Sistema para la generación de horarios académicos en instituciones universitarias usando algoritmo Tabú, 155. Retrieved from http://cybertesis.unmsm.edu.pe/xmlui/bitstream/handle/cybertesis/4913/Ramos_mf.pdf?sequence=1
- Reyes, F. J. P. (2011). Uso de algoritmos evolutivos para resolver el problema de asignación de horarios escolares en la Facultad de Psicología de la Universidad Veracruzana. Universidad Veracruzana, 84.
- Ricci, T. I. A. (2013). Aplicación de algoritmos genéticos para el problema de asignación de horarios en la división de ingenierías civil y geomática. Universidad Nacional Autónoma de México, 112.
- Robu, R., & Holban, S. (2011). A genetic algorithm for Classification. ICC'11 Proceedings of the International Conference on Computers and Computing, (May 2011), 52–56. Retrieved from <http://www.wseas.us/e-library/conferences/2011/Lanzarote/COMCOM/COMCOM-08.pdf>
- Rodríguez Varona, K. (2012). Aplicación de algoritmos genéticos en la generación automática de horarios docentes en la Facultad Regional de Granma. (Spanish). Application of Genetic Algorithms in the Automatic Generation of Class Schedules at the Regional School in Granma. (English), 6(4), 37–43. Retrieved from <http://ebiblio.cetys.mx:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=90570403&site=ehost-live>
- Rodríguez, É. (2016). Inteligencia artificial para la automatización de horarios escolares. Zacatecas, 2016.
- Ruiz, R., Garcia, J., & Perez, M. (2014). Causas y consecuencias de la deserción escolar. Revista de Investigación Educativa, 10(5), 51–74. Retrieved from <http://www.redalyc.org/pdf/461/46132134004.pdf>
- Saldaña Crovo, A., Oliva San Martin, C., & Pradenas Rojas, L. (2007). Models of Integer Programming for an University Timetabling Problem. Ingeniare., 15(3), 245–259. <https://doi.org/10.4067/S0718-33052007000300005>
- Schaerf, A. (1999). Survey of automated timetabling. Artificial Intelligence Review, 13(2), 87–127. <https://doi.org/10.1023/A:1006576209967>
- SEP. (2013). Glosario de términos. Glosario Educación Superior, 56. Retrieved from http://dsia.uv.mx/cuestionario911/Material_apoyo/Glosario_911.pdf

- Solano Sabatier, Y., Calvo Marín, M., & Trejos Picado, L. (2008). Implementación de un algoritmo genético para la asignación de aulas en un centro de estudio. *Uniciencia*, 22, 115–121.
- Sprockel, J. (2014). Programación Genética: Introducción y Aspectos Generales Programación Genética: (March 2013).
- 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>
- Torres-Ovalle, C., Montoya-Torres, J. R., Quintero-Araújo, C. L., Sarmiento-Lepesqueur, A., & Castilla-Luna, M. (2014). Programación de horarios y asignación de aulas de clases universitarias. *Ingeniería y Universidad*, 18(1). <https://doi.org/10.11144/JAVERIANA.IYU18-1.PHAA>
- Toscano Pulido Cd Victoria, G. (2009). Optimización de problemas con más de tres objetivos mediante algoritmos evolutivos. Retrieved from http://www.tamparcinestav.mx/~mgarza/files/garzafabre_msc.pdf
- UAEM. (2016). Tutor Académico UAEM. Retrieved from <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0ahUKEwi30pmMmLDhAhUUNn0KHf1PD08QFghHMAM&url=http%3A%2F%2Friae.uam.mx%2Fbitstream%2Fhandle%2F20.500.11799%2F66704%2Ftesis%2520final-split-merge.pdf%3Fsequence%3D1&usq=AOvVaw3VqNk2x8dxFnZ>
- Ubando, A., ... K. A.-A. 2013, & 2013, undefined. (n.d.). Fuzzy multi-objective approach for designing of biomass supply chain for poly generation with triple footprint constraints. *Asmedigitalcollection.Asme.Org*. Retrieved from <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1859164>
- Valenzuela Luna, M. L., & Rios Mercado, R. (2004). Comparando métodos heurísticos para secuenciar tareas en líneas de flujo. *Ingenierías*, 7(25), 35–39.
- Weare, R. F., Burke, E. K., & Elliman, D. G. (1995). A hybrid genetic algorithm for highly constrained timetabling problems. *Science*, 605–610. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.1047&rep=rep1&type=pdf>