

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS



Universidad
Tecnológica
de Pereira

**ALGORITMO HIPERHEURÍSTICO PARA GENERAR UNA SOLUCIÓN
FACTIBLE AL PROBLEMA DE LA PROGRAMACIÓN DE DOCENTES EN UNA
INSTITUCIÓN DE EDUCACIÓN MEDIA PARA UNA JORNADA ESCOLAR
ESPECÍFICA.**

TESIS DE MAESTRÍA

DIRIGIDA POR:

JORGE IVÁN RÍOS PATIÑO

PRESENTADA POR:

CARLOS ANDRÉS LÓPEZ

SANDRA MILENA CAICEDO

DEDICATORIA

A mis hijos Adrián y Damián, a mi esposa Ana
que siempre me apoyó y creyó en mi, los amo,
este trabajo es para ustedes.

Carlos Andrés.

A mi Familia, mis hijos Juan Pablo Y Miguel Angel y

A mi esposo Pablo, Siempre han sido mi motor
Para seguir mis sueños y metas, Los amo y doy

Gracias a Dios por tenerlos

Sandra Caicedo

AGRADECIMIENTOS

En nombre de quienes elaboramos esta tesis queremos agradecer a nuestras familias por el apoyo incondicional brindado durante este proceso, a ellos mil gracias.

También queremos agradecer a nuestros docentes, que se esmeraron en aportar tanto como le fue posible para transmitir sus conocimientos.

Al ingeniero Jorge Ivan Rios por su paciencia y apoyo incondicional, muchas gracias.

Al ingeniero Omar Trejos, gracias por el tiempo y las recomendaciones.

Contenido

GENERALIDADES.....	8
1.1 TITULO.....	8
1.2 ANTECEDENTES.....	8
1.3 JUSTIFICACIÓN.....	11
1.4 IDENTIFICACIÓN DEL PROBLEMA.....	13
1.5 OBJETIVO GENERAL.....	19
1.6 OBJETIVOS ESPECÍFICOS.....	20
1.7 METODOLOGÍA.....	20
1.8 HIPÓTESIS.....	21
ESTADO DEL ARTE.....	23
2.1 Documentos.....	23
MARCO TEÓRICO.....	45
3.1 Hiperheurística.....	45
3.1.1 Características.....	45
3.1.2 Clasificación.....	46
3.2 Bases de datos NoSQL.....	47
3.3 Unqlite.....	51
3.4 Algoritmos genéticos.....	52
3.5 UML.....	54
3.5.1 Diagrama de Clases.....	55
3.5.2 Diagrama de caso de uso.....	56
3.5.3 Diagrama de componentes.....	58
METODOLOGÍA.....	59
4.1 Heurísticas.....	59
4.1.1 Programación basada en áreas (áreas primero).....	60
4.1.2 Programación basada en intensidad horaria (asignaturas con más intensidad horaria primero).....	60
4.1.3 Programación basada en docentes (docentes primero).....	61
4.1.4 Programación basada en grados (primero los grados).....	61
4.2.1 Casos de Uso.....	61
4.2.3 Diagrama de componentes.....	73
4.3 Implementación.....	74
4.3.1 Elementos de la implementación.....	74
4.3.2 Heurística e hiperheurística.....	78
4.3.3 Heurística de bajo nivel.....	82
4.3.4 Hiperheurística.....	90
4.4 Análisis de Datos.....	96
CONCLUSIONES Y BIBLIOGRAFIA.....	100
5.1 CONCLUSIONES.....	100
5.2 Trabajo futuro.....	101
5.3 Bibliografía.....	101
5.4 Webgrafía.....	103
Anexo 1.....	104

LISTADO DE TABLAS

Tabla 1 Descripción de variables del Modelo Matemático.....	18
Tabla 2 Algoritmo Genético General.....	52
Tabla 3 Nombre del Caso de Uso	62
Tabla 4 Cargar Registros.....	63
Tabla 5 Configurar Restricciones	64
Tabla 6 Confirmar Restricciones	64
Tabla 7 Almacenar Registros.....	65
Tabla 8 Consulta de Datos	65
Tabla 9 Seleccionar y Ejecutar Heurística	66
Tabla 10 Gestión de Resultados	67
Tabla 11 Configuración Metaheurística	67
Tabla 12 Ejecutar Metaheurística	68
Tabla 13 Generar Horarios	68
Tabla 14 Consultar Horarios.....	69
Tabla 15 Muestra.....	70
Tabla 16 Hora.....	71
Tabla 17 Grupo.....	72
Tabla 18 AG.....	73
Tabla 19 Fragmento de datos de archivo de carga docente y fragmentos de archivo de asignaturas.....	77
Tabla 20 Algoritmo de Pre procesamiento.....	78
Tabla 21 Algoritmo de asignación horaria.....	79
Tabla 22 Análisis de complejidad	79
Tabla 23 Heurística basada en áreas	83
Tabla 24 Ajuste de horario.....	84
Tabla 25 Salida del sistema, carga docente	85
Tabla 26 Información de carga horaria esperada por grado.....	86
Tabla 27 Programación de carga horaria por grado arrojada por el sistema.....	87
Tabla 28 Carga docente arrojada por la heurística.....	88
Tabla 29 Datos de asignación de heurística de bajo nivel.....	89
Tabla 30 Colección en bse de datos Soporte.uql con Descripción de id, nombre y carga esperada	90
Tabla 31 Descripción de carga horaria arrojada por el sistema para cada grupo, registro de la forma grupo, lista de la forma asignatura-carga horaria -id docente asignado.....	90
Tabla 32 Genoma del algoritmo genético adaptado al problema	91
Tabla 33 Algoritmo Hiperheurístico para la asignación de horarios	92
Tabla 34 Valores arrojados por la clase soporte. En la parte superior listado de carga docente por grupo. En la parte inferior diccionario de información de grupos.....	94
Tabla 35 Configuración Algoritmo Genético	97
Tabla 36 Configuración de la Hiperheurística	97
Tabla 37 Datos Arrojados de Heurística de bajo nivel	97
Tabla 38 Datos Arrojados Hiperheurística	98

Tabla 39 Muestra de horario de programación docente, carga asignada por docente..	98
Tabla 40 Muestra de horario generado para el día lunes de todos los grupos	99

LISTADO DE ILUSTRACIONES

Ilustración 1 Triangulo CAP Fuente: DataStax.com.....	49
Ilustración 2 Modelo de Cruce de un Punto y Dos Puntos [18]	53
Ilustración 3 Diagrama de Clases [12].....	56
Ilustración 4 Diagrama de Casos de Uso [12].....	57
Ilustración 5 Diagrama de Componentes fuente : [21]	58
Ilustración 6 Diagrama de Casos de Uso	62
Ilustración 7 Diagrama de Clases	69
Ilustración 8 Diagrama de Componentes Software	73
Ilustración 9 Archivo de MS Excel Con Valores Iniciales.....	76

CAPITULO 1

GENERALIDADES

1.1 TITULO

Análisis e Implementación de un algoritmo hyper heurístico para la generación de una solución factible al problema de la programación de docentes en una institución de educación media para una jornada escolar específica.

ÁREA: Aprendizaje de máquina, heurística, hiper heurística.

LÍNEA DE INVESTIGACIÓN: inteligencia artificial

1.2 ANTECEDENTES

Cuando abordamos problemas cuyo espacio de posibles soluciones se nos facilita conocer, nuestra búsqueda se reduce a hallar un óptimo (un máximo o un mínimo) que da una solución al problema dadas unas restricciones. Sin embargo existen problemas de optimización combinatoria complejos en diversos campos como la economía, el comercio, la ingeniería, re-ingeniería de software, comunicaciones, la industria o la medicina, que a menudo son muy difíciles de resolver en la práctica y cuyo modelamiento matemático “Formulado coherentemente”, no permite una solución con las herramientas analíticas al alcance. Por eso evaluando la complejidad actual generada en las instituciones y el mundo empresarial, así como la diversidad de fases en que se desarrolla el proceso de cambio institucional, se hace necesario el desarrollo

y uso de métodos que permitan el manejo de situaciones problemáticas desde el conjunto de sus partes y desde distintos puntos de vista.

Debido a ese bajo rendimiento de los algoritmos exactos para muchos problemas, se han desarrollado un gran número de algoritmos aproximados, que proporcionan soluciones de alta calidad para estos problemas combinatorios (aunque no necesariamente la óptima) en un tiempo computacional breve; estos algoritmos incluyen las denominadas técnicas heurísticas, Metaheurísticas e Hiper heurísticas.

El objetivo de la optimización combinatoria es encontrar un objeto matemático finito que maximice o minimice, dependiendo del problema en una función objetivo, este contiene unos objetos llamados estados y al conjunto de todos los estados candidatos se les llama espacio de búsqueda. La naturaleza de los estados y del espacio de búsqueda son usualmente específicos del problema. La solución del problema es aquella que tiene mejor función objetivo y que cumple todas las restricciones

Heurísticas: Son algoritmos que encuentran soluciones de baja calidad para problemas combinatorios complejos de tipo NP. Los Algoritmos heurísticos son más fáciles de implementar y encuentran soluciones aceptables locales con esfuerzos computacionales relativamente pequeños; sin olvidar que estos algoritmos renuncian a encontrar soluciones óptimas globales de un problema.

Metaheurísticas: representa una familia de técnicas de optimización aproximada que provee una solución aceptable en un tiempo razonable a problemas de alta complejidad en ciencia e ingeniería pero que no garantizan la calidad de la solución obtenida pues no es posible establecer que tan cerca se está de la solución óptima .

Las metaheurísticas principales se refieren a métodos de relajación, procesos constructivos, búsquedas por entornos y procedimientos evolutivos. Las metaheurísticas de búsqueda constituyen el paradigma central de estas técnicas en la resolución de problemas de optimización.

Hiper Heurísticas: Método de búsqueda o mecanismo de aprendizaje para seleccionar o generar heurísticas que resuelven problemas de alta dificultad computacional [3]. Una hiper heurística es un tipo de metodología de búsqueda de propósito general con la meta de automatizar el diseño y adaptación de métodos heurísticas y metaheurísticos. Las hiper heurísticas actúan sobre las metaheurística buscando mejorar su desempeño. Sus características son:

- ✓ Operar sobre el espacio de búsqueda de las heurísticas en vez de actuar sobre el espacio de búsqueda de la solución.
- ✓ Es posible usar heurísticas ya existentes o generadas al interior de la hiper heurística.
- ✓ Ayuda a sacar ventaja de las fortalezas y a evitar las debilidades de cada heurística.
- ✓ Ningún conocimiento específico es necesario durante la búsqueda en el espacio heurísticos, por ello los componentes de la hiperheurística son re utilizables.
- ✓ Fácil de implementar, descargar, usar (fácil, barato, rápido).
- ✓ Aplicable a un amplio espectro de problemas reales.

Es decir que una solución posible a un problema computacionalmente costoso puede ser abordarlo mediante la implementación de una hiper heurística aplicada sobre alguna metaheurística o grupo de estas.

Algunas metaheurísticas que pueden llegar a ser articuladas con una hiper heurística son:

- ✓ Algoritmos Genéticos: son métodos adaptativos que pueden utilizarse para resolver todo tipo de problemas complejos de búsqueda y optimización, lo que lo convierte en una metaheurística. Es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años.

- ✓ Colonia de Hormigas: Las colonias de Hormigas y en general las sociedades de insectos, presentan sistemas sociales altamente desarrollados que les permiten ejecutar tareas altamente complejas a pesar de la simplicidad de sus individuos. esta estructura es motivo de estudio en ramas de la ciencia que van desde el campo de la biología hasta las matemáticas y la computación, esta idea de algoritmo fue iniciada por Dorigo, Naniezzo y Colorni .

El principio de esta metodología se fundamenta en estudiar como las hormigas buscan el alimento y construyen el camino más corto para ir y volver entre el hormiguero y la fuente del alimento.

1.3 JUSTIFICACIÓN

Los problemas de asignación de horarios y recursos en las instituciones educativas han sido de gran interés y ampliamente estudiados por años. la literatura presenta diferentes variaciones de este problema, dependiendo de los requerimientos

específicos de cada institución; en muchas investigaciones se le denomina el “Scheduling Problem” que no solamente se utiliza para asignación de horarios a profesores sino que también por ejemplo la programación de actividades, asignación de personal, en transporte, clínica, entre otros. Debido a su gran extensión, se hace necesario clasificar “Scheduling Problems” hacia actividades especializadas, como el “Timetabling” (Programación de horarios), que en el campo de la educación pueden clasificarse en tres categorías, a saber: “Examination Timetabling”, “School Course Timetabling” y “University Course Timetabling [8]; en el caso de nuestra investigación nos enfocaremos en la programación y asignación de carga docente para una institución educativa.

Es bien sabido que cada Institución Educativa, al inicio del año escolar, programa currículos y horarios para todo el resto del año, un aspecto importante es el Recurso Humano con el que se cuenta para esta asignación de materias, salones y profesores, permitiendo así una eficiencia y calidad educativa dentro del Plantel .

Cada profesor tiene unas restricciones específicas, todas estas deben estar presentes a la hora de la asignación para brindar mayor satisfacción a la planta docente; realizar esto manualmente significa que una persona encargada de esta labor en particular, debe evaluar y mirar cada una de las restricciones para así, cuadra perfectamente el horario de cada maestro, el cual según la ley (DECRETO NUMERO 1850 DE 2002) [9] debe cumplir a la semana 30 horas de las cuales 22 horas son en clase y 8 horas de actividades tales como reuniones, preparación de clases, asistencia a padres de familia etc..; también se deben tener presente que la institución puede contratar docente por horas cátedra; un tarea bastante compleja que puede acarrear

varios errores, si analizamos que se programan muchas asignaturas con diferentes tipos e intensidad horaria, y que al final pueden presentarse modificaciones. En la institución educativa que facilitó la información la persona encargada de la programación docente es el coordinador y este proceso le toma hasta un mes.

La importancia de encontrar métodos más eficientes y eficaces a la hora de la selección y asignación de todos los recursos (humano, Materias, salones), permitiría a estas Instituciones Educativas una mejor labor administrativa y así responder tanto a la satisfacción de los requerimientos del PEI como a los individuales del docente, dentro del contexto laboral, asumiendo los nuevos retos institucionales y haciendo partícipes a los docentes en la implementación de los planes, programas y proyectos, de tal manera que se combinen los fines de la educación y los del Estado, con sus fines de desarrollo como persona .

Los algoritmos de Inteligencia Artificial, como los algoritmos genéticos son una herramienta que tiene ventajas competitivas sobre otros métodos tradicionales de optimización, ya que este genera soluciones eficientes apropiadas en los casos particulares de asignación de horarios, salones y personal, por tener la capacidad de captar toda la complejidad que conlleva la información imprecisa.

1.4 IDENTIFICACIÓN DEL PROBLEMA

El problema de asignación de horarios pertenece a la categoría conocida como *scheduling* (*Planificación*) y que versa sobre la asignación de recursos, tareas o tiempos necesarios para completar una tarea, siendo el resultado un modelo de asignación planificado.

El problema de planificación (*scheduling*) define cuatro elementos básicos [10]:

- ✓ Duración de las actividades
- ✓ Restricción de precedencia para cada actividad
- ✓ Restricción de inicio y terminación para cada trabajo
- ✓ Restricción de capacidad sobre el uso de cada recurso

Dependiendo del resultado esperado este puede ser factible, es decir que satisface un conjunto de restricciones preestablecidas, o puede ser óptima, en cuyo caso además de cumplir con las restricciones son optimizadas una o varias de ellas.

Además se sabe que existen dos tipos de problemas de planificación, aquellos que versan sobre el problema de la producción, que posee cuatro categorías generales (open-shop, job-shop, flow-shop, permutation flow-shop) y los de asignación de recursos[10].

En la asignación de recursos se dispone de un conjunto de operaciones que deben ser realizadas, un conjunto de recursos idénticos pero no equivalente que realizan el mismo tipo de operación y que pueden ser utilizados, nuestro problema pertenece a esta categoría.

Podemos establecer para este problema en particular los siguientes objetos de análisis:

- ✓ Grado: audiencia que requiere un turno en la unidad de tiempo. Un grado escolar es un conjunto de alumnos que cursan el mismo año pero ocupan un espacio diferente de clase al cual se debe asignar un horario, cada grado se trata como un problema de asignación de recursos independiente, pero relacionado, con los demás grados.

✓ Horario: asignación periódica semanal de turnos. Se asume que las tareas programadas para el lunes serán las mismas para el lunes siguiente. Una semana tiene un número de unidades diarias asignadas.

✓ Unidad: hora de clase. Número de horas por día que requiere recursos.

✓ Asignatura: característica de la unidad de tiempo. Una unidad de tiempo está asociada a una característica única, la asignatura, que restringe el turno.

✓ Hora: personal asignado en la unidad de tiempo. Recurso necesario empleado durante una hora de clase y que debe pertenecer a una asignatura en particular. Un turno solo puede ser asignado a un individuo a la vez. Una vez terminada la hora de clase el recurso puede ser liberado. Es posible que al finalizar el turno todos los recursos sean liberados.

✓ Docente: recurso humano. Un docente es un individuo con áreas específicas y que puede tomar un turno a la vez. Los docentes están limitados en el número de horas que es posible asignarles en una semana y a un conjunto de asignaturas para el que está capacitado. Puede existir restricción en cuanto al grado que puedan ser asignados.

La asignación de grados tiene la misma jerarquía que la asignación de docente, pues para ambos se debe hallar un horario para cada día de la semana, con un número de horas de trabajo en relación con una asignatura específica.

Las variables que deben ser especificadas para la asignación del horario factible serán:

- ✓ Cantidad de grados: cantidad de grupos por grado. Pueden existir por ejemplo dos grados noveno, tres grados octavo.
- ✓ Cantidad de horas diarias: una institución establece cuántas horas diarias por grado requieren en la semana. Por ley en educación media son treinta (30) horas por semana, pero en instituciones privadas puede variar.
- ✓ Asignaturas por grado. Conjunto de asignaturas por grado que deben ser distribuidas en la semana.
- ✓ Horas por asignatura. Cantidad de horas por asignatura que deben ser distribuidas en la semana.
- ✓ Docentes. Conjunto de docentes disponibles.
- ✓ Horas por docente. Cantidad de horas por docente que deben ser distribuidos en la semana.
- ✓ Asignatura por docente. Conjunto de docentes asociados a una asignatura.

El problema resulta en encontrar la mejor combinación de asignación horaria para cada grado acorde a los recursos docentes disponibles, resultando un problema NP-Hardn (NP-duro), que implica que el espacio de solución a un problema es tan amplio que computacionalmente es difícil o imposible explorar todas las combinaciones en el espacio solución del problema.

También al problema se encuentran asociadas las siguientes restricciones:

- ✓ Solo puede haber un docente por hora escolar

✓ un docente solo puede laborar n horas a la semana, en instituciones públicas es actualmente de 22 pero si la legislación varía este parámetro también.

✓ Un día escolar es de 6 horas, pero puede variar de una institución a otra, de un grado a otro o según la legislación.

✓ Un docente puede ser asignado a un conjunto de asignaturas comodín para completar su carga docente.

✓ Un docente puede tener más horas de ser necesario (horas extra).

La tabla 1 muestra las variables que se requieren para formular matemáticamente el problema.

Se puede expresar que una solución factible es aquella que consigue para el número de horas a la semana (nhs) asignar para cada grado (g), para cada asignatura (as), para cada docente (d), tal forma que no supere el límite de horas permitido (hd) para los docentes, que cada docente asignado cumpla con la relación docente asignatura (das), y que la suma del número de horas en cada grado (nas) debe ser igual al número de horas a la semana.

Variable	
G	Conjunto de grados, $g=\{\text{sexto, séptimo,}\dots, \text{once}\}$ que puede representarse como $g=\{1,2,\dots,6\}$ cuya relación es 1:sexto, 2:séptimo, etc.
Cg	Cantidad de grados es la cardinalidad del conjunto g
Ng	Conjunto de cantidad de grados, p.e. $ng=\{4,2,2,2,2,1\}$ representa 4 grados sexto, 2 grados séptimo, etc.
Días	Conjunto de días de clase, $días=\{\text{lunes, martes,}\dots\}$
Cd	Numero de días de la semana, cardinalidad del conjunto días
Nhs	Número de horas a la semana
Horas	Cantidad de horas diarias
As	Conjunto de asignaturas, $as=\{\text{matemáticas, castellano,}\dots\}$
Cas	Cantidad de asignaturas es la cardinalidad del conjunto as
Nd	Número de docentes
Hd	Número de horas máximas asociadas a un docente
D	Conjunto de docentes, $d=\{\text{profesor1, profesor2,}\dots\}$
Had	Conjunto de horas asignadas a un docente, $had=\{0,19,\dots\}$
Das	Conjunto que expresa la relación docente asignatura, $das=\{(\text{profesor1, matemáticas}), (\text{profesor1, álgebra}), (\text{profesor2, castellano}),\dots\}$
Asg	Conjunto que expresa la relación asignatura grado, $asg=\{(\text{matemáticas,1}), (\text{álgebra,4}), (\text{castellano,1}),\dots\}$
Nas	Conjunto que expresa el número de horas por cada asignatura en un cada grado, posee la cardinalidad del conjunto asg. P.e. sea $nas=\{4,4,5,\dots\}$ el cual representa el numero de horas (4) para matemáticas en el grado 1
Casg	Cantidad de asignaturas por grado es la cardinalidad del conjunto asg

Tabla 1 Descripción de variables del Modelo Matemático

De esta forma:

$$z = nhs \sum_{i=1}^{cg} g_i n g_i \sum_{j=1}^{cas} as_j \sum_{k=1}^{nd} d_k \quad (1)$$

S.a.

$$\sum_{i=1}^{nd} had_i \leq hd \quad (2)$$

$$\forall y (das(x, y) \wedge nas(y, z)) \quad (3)$$

$$\sum_{i=1} nas_i asg_i = nhs \quad (4)$$

Donde d:docente, as:asignatura, z: grado

Se desea encontrar una asignación de horaria que satisfaga las condiciones anteriores. Se observa que se trata de un problema que requiere una solución en el ámbito de las variables discretas.

Por tanto, se desea encontrar una hiperheurística que permita hallar una solución factible al problema de programación docente en una institución educativa para la jornada media escolar acorde a las variables identificadas y las restricciones del problema.

1.5 OBJETIVO GENERAL

Implementación de un algoritmo hiperheurístico para generar una solución factible al problema de la programación de docentes en una institución de educación media para una jornada escolar específica.

1.6 OBJETIVOS ESPECÍFICOS

- ✓ Modelar heurística de bajo nivel para el problema de la asignación de horario
- ✓ Implementar heurística de bajo nivel para el problema de asignación de horario
- ✓ Modelar algoritmo genético para la asignación de horario
- ✓ Implementar hiperheurística de adaptación de algoritmo genético para asignación de horarios
- ✓ Establecer criterios que posibiliten la medición del mejoramiento de la hiperheurística

1.7 METODOLOGÍA

Se tendrá como escenario de desarrollo una institución educativa pública que cuenta con datos de docentes y su carga horaria actual, grados y cursos, cantidad de horas asignadas a los grupos, no se facilitará información de horario.

El objetivo consistirá en generar un horario para todos los cursos y para toda la semana, asignando los docentes y sus asignaturas de tal forma que se cumpla con la cantidad de horas especificada para cada curso y las asignaturas y su carga horaria esperada para cada grado. Finalmente, no debe existir cruce horario entre docentes en la programación semanal.

Para alcanzar el objetivo se implementará una hiperheurística la cual consta de dos elementos generales: heurística de bajo nivel y metaheurística.

Las heurísticas de bajo nivel son aquellas que permitirán encontrar la carga asociada a cada docente, adaptando la solución a las necesidades de la institución y en consonancia con el área de trabajo de cada docente. Como segunda instancia disminuir la complejidad del problema al reducir las variables con las que debe interactuar el sistema.

Una heurística de bajo nivel es por ejemplo, asignar la carga tomando las asignaturas y asociando esta a los docentes, otra heurística podría ser aquella en donde se asigna la carga tomando los grupos y luego asignando la carga docente. Existen diversas combinaciones posibles para realizar este proceso, en este estudio se implementarán al menos dos.

Una vez la heurística de bajo nivel arroja como salida una carga horaria que relaciona grado, grupo, docente y asignatura se pasará a implementar una metaheurística para establecer el horario adecuado.

Una metaheurística es un método heurístico refinado para entregar resultados de buena calidad.

Las hiperheurística es entonces un modelo que se encuentra por encima de las heurísticas y las metaheurísticas y tiene como función adecuar las entradas, las salidas y la configuración de los parámetros bajo los cuales estas realizan su tarea de una manera automática de tal forma que se encuentre una solución de buena calidad.

1.8 HIPÓTESIS

Ante distintas instancias y momentos del problema la aplicación de las heurísticas y metaheurísticas cuentan con variables de configuración que permiten adaptar mejor el modelo a la solución, estos momentos son identificables y medibles.

Es posible la construcción de una heurística superior a las heurísticas y metaheurísticas, la hiperheurística, capaz de ajustar parámetros y adaptar el modelo heurístico o metaheurístico empleado para encontrar una mejor solución del problema.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 Documentos

Título Evolving hyper heuristic using Genetic Programming

Fuente: Ben-Gurion University of the Negev, 2014

Autor Achiya Eliasay (correo electrónico) – PhD Philosophy

Resumen

The application of computational intelligence techniques within the vast domain of games has been increasing at a breathtaking speed. Over the past few years my research has produced a plethora of results in games of different natures, evidencing the success and efficiency of evolutionary algorithms in general and genetic programming in particular at producing Top-notch, human-competitive game strategies.

Studying games may advance our knowledge both in cognition and artificial intelligence, and, last but not least, games possess a competitive angle that coincides with our human nature, thus motivating researchers. In this dissertation I explore the application of genetic programming to the development of search heuristics for difficult games. I apply GP to the evolution of solvers for the Rush Hour puzzle and the game of FreeCell, along the way demonstrating a general method for evolving heuristics.

palabras claves:

Evolutionary Algorithms, Genetic Algorithms, Genetic Programming, Heuristic, Hyper Heuristic, FreeCell, Rush Hour, HH-Evolver, IDA

Descripción

El uso de las técnicas de la inteligencia computacional, en los juegos se ha incrementado en los últimos años, haciendo uso de los algoritmos evolutivos y la programación genética, permitiendo así la creación de juegos más competitivos y de mejor calidad , El enfoque que adopta este trabajo se encuentra dentro del marco hiper heurístico, en el que el sistema está provisto de un conjunto de heurísticas predefinidas o preexistentes para resolver un determinado problema, e intenta descubrir la mejor manera de aplicar estas en diferentes etapas del proceso de búsqueda. El objetivo es encontrar nuevas heurísticas de nivel superior o hiper heurísticas.

Aportes a la Tesis

Mediante las múltiples heurísticas que existen, este trabajo brinda un enfoque diferente a lo acostumbrado en la literatura normal sobre el uso de las técnicas de optimización; en esta tesis doctoral la enfocan a la creación de juegos más competitivos, mostrando que una buena utilización y combinación de diferentes tipos de heurísticas permite la construcción de una hiper heurística con mejores desempeños y mejor resultado. También presenta diferentes perspectivas y modelos en entornos distintos, es un abanico de problemas y soluciones.

Título Hyper-heuristics for cross-domain search

Fuente: Institute of Computing Science, Poznań University of Technology,2012

Autor T. CICHOWICZ, M. DROZDOWSKI

M. FRANKIEWICZ, G. PAWLAK

(grzegorz.pawlak@cs.put.poznan.pl)

F. RYTWINSKI, and J. WASILEWSKI

Palabras claves:

hyper-heuristics, cross-domain heuristic search, HyFlex.

Resumen

In this paper we present two hyper-heuristics developed for the Cross-Domain Heuristic Search Challenge. Hyper-heuristics solve hard combinatorial problems by guiding low level heuristics, rather than by manipulating problem solutions directly. Two hyper-heuristics are presented: Five Phase Approach and Genetic Hive. Development paths of the algorithms and testing methods are outlined. Performance of both methods is studied. Useful and interesting experience gained in construction of the hyper-heuristics are presented. Conclusions and recommendations for the future advancement of hyper-heuristic methodologies are discussed.

Descripció

En este artículo el autor muestra y experimenta en la construcción de 2 hiper heurísticas para la búsqueda de heurísticas entre dominios cruzados, una las HH fue la de 5 fases construida a partir de una compleja arquitectura de control con heurísticas de bajo nivel, a su vez se subdividió en 3 fases (clasificación, Intensificación y Diversificación) y de allí se fue mejorando hasta llegar al nivel de 5 fases mejorando el algoritmo, la otra HH fue colonia genética, basada en la colonia de abejas esta fue Desarrollada como una solución alternativa y competitiva, dando esta última mejor eficiencia Sin embargo, sin tener mucho éxito en la competencia, mostrando así los logros y lecciones adquiridas dejando así una valiosa contribución al área de investigación.

Aportes a la Tesis

El autor aborda diferentes problemas combinatoriales duros de resolver en el mundo real, y muestra una gran variedad en técnicas y algoritmos de optimización basados en heurísticas, metaheurísticas; tomados de la naturaleza y de procesos artificiales; donde muestra que estas técnicas son eficientes pero solo sirven para problemas de un dominio particulares, el interés real del autor es mostrar , cómo por medio de una combinación de estas técnicas se puede construir y potencializar una mejor herramienta que permita que estos problemas de difícil solución, se puedan resolver por medio de las hiper heurísticas mediante métodos constructivos aplicables para resolver una amplia gama de problemas difíciles.

Título Algorithm Selection: From Meta-Learningto Hyper-Heuristics

Fuente: Instituto Tecnológico de Cd. Madero

 Centro Nacional de Investigación y Desarrollo Tecnológico

 Universidad Politécnica de Nuevo León

 Universidad de Ciudad Juárez

 México

 www.intechopen.com

Autor Laura Cruz-Reyes , Instituto Tecnológico de Cd. Madero

 Claudia Gómez-Santillán, Instituto Tecnológico de Cd. Madero

 Joaquín Pérez-Ortega, Centro Nacional de Investigación y Desarrollo

 Tecnológico

 Vanesa Landero, Universidad Politécnica de Nuevo León

 Marcela Quiroz, Instituto Tecnológico de Cd. Madero

Alberto Ochoa, Universidad de Ciudad Juárez

Palabras claves:

hyper-heuristics, learning system, heurísticas, Decision rule

Resumen

In order for a company to be competitive, an indispensable requirement is the efficient management of its resources. As a result derives a lot of complex optimization problems that need to be solved with high-performance computing tools. In addition, due to the complexity of these problems, it is considered that the most promising approach is the solution with approximate algorithms; highlighting the heuristic optimizers. Within this category are the basic heuristics that are experience-based techniques and the metaheuristic algorithms that are inspired by natural or artificial optimization processes.

Descripción

Dada la cantidad de estrategias disponibles y los diferentes algoritmos para la Selección el mejor y que más se adapte al problema es una tarea bastante ardua; dado que no existe un algoritmo que se desempeñe mejor para todas las situaciones; en este artículo el autor juego con diferentes estrategias para el mejor desempeño en los algoritmos, herramientas como los sistemas inteligentes, aprendizaje de máquina y redes neuronales son algunos de los técnicas que utiliza para la eficiencia y eficacia en la construcción de las hiper heurística

Aportes a la Tesis

Este artículo aborda temas relacionados con los diferentes problemas de optimización que se presentan a diario, proponiendo un marco de soluciones con las diferentes

metodologías existentes; temas como: aprendizaje de máquina, mediante la clasificación de mejores soluciones, la extracción de reglas con algoritmos genéticos, análisis de desempeño, asignación y clasificación de tarea, heurísticas para el mejoramiento en su rendimiento; hace un aporte significativo a la hora de construir una hiperheurística con todos estas posibles herramientas mejorando así su desempeño

Título DESIGN OF A HYPERHEURISTIC FOR PRODUCTION SCHEDULING
 IN JOB SHOP ENVIRONMENTS

Fuente: Departamento de Ingeniería Industrial, Universidad Nacional de Colombia
 Campus la Nubia Manizales, Colombia

Autor Omar Danilo Castrillón (adcastrillong@unad.edu.co)
 William Ariel Sarache (jaiagiraldo@unad.edu.co)
 Jaime Alberto Giraldo (wasarachec@unad.edu.co)

Palabras claves:

Job shop, scheduling, heuristics, simulation, fitness, evaluation function, makespan, idle time.

Resumen

The objective of the present work is to diminish the total process time (Makespan) and to increase the machine process time, by diminishing the idle time in a job-shop environment. Through the design of a hyperheuristic based on an ant colony and genetic algorithms. This work is developed in two phases: in the first phase, a hyperheuristic identification and definition is carried out for sequencing processes in job shop environments. In the second phase, the system effectiveness in the traditional production programming is shown. In the investigation project, an enterprise from the

metal mechanic sector was chosen, where by means of a combination of an ant colony and genetic algorithms, the optimal route for an order is scheduled, achieving the optimization or suboptimization of its respective total process time in an upper percentage of 95%.

Descripción

En este trabajo los autores muestra el uso de las diferentes heurísticas utilizadas como técnicas o procedimientos para llevar a cabo una tarea específica, siendo esta en muchos caso ineficientes y conduciendo a la construcción de las metaheurísticas que son un conjunto de estrategias inteligentes para la mejora de los procedimientos heurísticos, problemas tales como distribución de recursos, asignación de tareas, dificultad en el manejo de órdenes de compra e inventario son algunos ejemplos que muestran que pueden existir múltiples heurísticas para tratar de resolverlos, pero están son técnicas estáticas o de corto alcance ..

Aporte a la Tesis

Algoritmos Tales como colonia de hormigas y algoritmos genéticos son llamados a resolver problemas de Hard NP. uno de los objetivos de estos algoritmos es el de encontrar las reglas de secuencias prioritarias que permitan la optimización de un objetivo deseado, muy útiles en el campo Industrial; estos algoritmos utilizados conjuntamente con el objetivo de optimizar o suboptimizar la solución de un tipo de problema en general, produciendo y mejor resultado que si se utilizaran independientemente

Título: Hyper-heuristics: a survey of the state of the art

Fuente: Journal of the Operational Research Society advance online publication,
10 July 2013

Autor: Edmund K Burke , University of Stirling, Scotland, UK
Michel Gendreau , University of Montreal, Montreal, Canada
Matthew Hyde, University of East Anglia, Norwich, UK
Graham Kendall, University of Nottingham, Nottingham, UK
Gabriela Ochoa University of Stirling, Scotland, UK
Ender Ozcan, University of Nottingham, Nottingham, UK
Rong Qu, University of Nottingham, Nottingham, UK

Palabras clave:

Hyper-heuristics, evolutionary computation, metaheuristics, machine learning,
combinatorial optimisation, scheduling.

Resumen:

Hyper-heuristics comprise a set of approaches that are motivated (at least in part) by the goal of automating the design of heuristic methods to solve hard computational search problems. An underlying strategic research challenge is to develop more generally applicable search methodologies. The term hyper-heuristic is relatively new; it was first used in 2000 to describe heuristics to choose heuristics in the context of combinatorial optimisation. However, the idea of automating the design of heuristics is not new; it can be traced back to the 1960s. The definition of hyper-heuristics has been recently extended to refer to a search method or learning mechanism for selecting or generating heuristics to solve computational search problems. Two main hyper-heuristic categories can be considered: heuristic selection

and heuristic generation. The distinguishing feature of hyper-heuristics is that they operate on a search space of heuristics (or heuristic components) rather than directly on the search space of solutions to the underlying problem that is being addressed. This paper presents a critical discussion of the scientific literature on hyper-heuristics including their origin and intellectual roots, a detailed account of the main types of approaches, and an overview of some related areas. Current research trends and directions for future research are also discussed.

Descripción

Este artículo recoge información sobre el estado del arte de las hiperheurísticas, describe el antecedente histórico del modelo, describe en qué consiste el aprendizaje para métodos heurísticos para luego entrar en detalles del modelo arquitectónico del sistema. Identifica y explica las formas en que operan las heurísticas de bajo nivel en el modelo y menciona los dominios de aplicación de las hiperheurísticas y menciona además autores y soluciones planteadas por diferentes investigadores. Por último, expone sobre diferentes metodologías de generación de hiperheurísticas y realiza un análisis de dominios donde han sido aplicadas

Aportes a la Tesis

Aporta en gran medida a la comprensión de la parte teórica de las hiperheurísticas ya que describe los elementos constitutivos como arquitectura y componentes. Contiene además una descripción general de las metodologías de selección y generación, todo ello asociado al estado del arte a través de la descripción del aporte y el autor del aporte enmarcado en los diferentes casos de implementación

agregando también una descripción del contexto de la implementación. Entre los contextos mencionados se halla el problema de asignación de horarios escolares.

Título: Modelo matemático para la programación de un horario escolar con multi-localización de docentes

Fuente Tesis Maestría en Ingeniería – Énfasis en Ingeniería Industrial, Maestría en Ingeniería industrial, Santiago de Cali 21 de Septiembre de 2014

Autor Linda Lucía Esquivel T., Universidad del Valle

Palabras clave

Programación lineal, programación entera, school timetabling problem (problema de asignación de horarios escolares), Lingo

Resumen

El documento comienza con la descripción de la legislación del estado colombiano acerca de la educación y la contratación docente, ya que esto establece métricas en cuanto a tiempos y límites. Posteriormente describe distintas formas en que se ha abordado el problema de asignación de horarios escolares y categoriza los tipos de soluciones en dos grandes grupos, las que se basan en programación lineal y las que emplean metaheurísticas. Menciona también que el problema de asignación de horarios al carecer de una función objetivo es un problema de búsqueda y no de optimización por lo que el objetivo principal es encontrar un horario válido (pag 21 documento citado). Finalmente modela el problema y encuentra una solución factible empleando el Lingo, un software privativo para solución de problemas de programación lineal y se publica en los anexos tanto el programa como el resultado.

Descripción

El objetivo general es la asignación de horario escolar donde los docentes comparten dos instituciones educativas y además se encuentran en situaciones contractuales diversas, es decir, desarrollar un modelo matemático para la programación de horarios con restricciones de multilocalización y necesidades del cuerpo docente. En la descripción del problema se puede observar que el incluir dos instituciones aumenta el espacio de búsqueda de un problema que ya es grande en sí mismo. En el estado del arte menciona el uso de las metaheurísticas de recocido simulado, algoritmo genético y búsqueda tabú e incluye información sobre modelos de programación lineal y programación entera, camino que finalmente aborda en la búsqueda de la solución. En el desarrollo de la tesis identifica y formula el problema de programación lineal, describe las variables y realiza análisis de sensibilidad de estas, finalmente construye una solución a empleando Lingo, una herramienta de software privativo para resolver problemas de programación lineal y genera un horario que satisface, para probar la viabilidad del modelo encontrado.

Aportes a la Tesis

Realiza una descripción de diversas estrategias empleadas para resolver el problema de asignación de horarios. No aborda la perspectiva de las ciencias computacionales si no la perspectiva de la programación lineal, son particularmente interesantes las técnicas para describir el modelo, pues divide en distintas instancias las restricciones, lo que plantea la posibilidad de abordar la solución del problema por etapas. Permite observar que es posible definir conjuntos de solución asociado a las restricciones, aporte importante ya que en las hiperheurísticas la capacidad de describir

conjuntos solución donde aplicar heurísticas o metaheurísticas es parte de la metodología. En el estado del arte menciona como principales líneas de investigación algoritmos que emplean metaheurísticas y además describe cómo se emplean modelos de programación lineal, recopila esta información en un cuadro resumen; cabe mencionar que las hiperheurísticas no se encuentran en el espectro de soluciones encontradas por la autora.

Título Multi-stage Hyper-heuristics for Optimisation Problems

Fuente Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy. December 2014.

Autor Ahmed Kheiri, B.Sc. (Hons), M.Sc.

Palabras clave

hyper-heuristic, meta-heuristics, multi-stage hyper-heuristic, optimization, school timetabling

Resumen

There is a growing interest towards self configuring/tuning automated general-purpose reusable heuristic approaches for combinatorial optimisation, such as, hyper-heuristics. Hyper-heuristics are search methodologies which explore the space of heuristics rather than the solutions to solve a broad range of hard computational problems without requiring any expert intervention. There are two common types of hyper-heuristics in the literature: selection and generation methodologies. This work focusses on the former type of hyper-heuristics. Almost all selection hyper-heuristics perform a single point based iterative search over the space of heuristics by selecting and applying a suitable heuristic to the solution in hand at each decision point. Then the

newly generated solution is either accepted or rejected using an acceptance method. This improvement process is repeated starting from an initial solution until a set of termination criteria is satisfied. The number of studies on the design of hyper-heuristic methodologies has been rapidly increasing and currently, we already have a variety of approaches, each with their own strengths and weaknesses. It has been observed that different hyper-heuristics perform differently on a given subset of problem instances and more importantly, a hyper-heuristic performs differently as the set of low level heuristics vary. This thesis introduces a general “multi-stage” hyper-heuristic framework enabling the use and exploitation of multiple selection hyper-heuristics at different stages during the search process. The goal is designing an approach utilising multiple hyper-heuristics for a more effective and efficient overall performance when compared to the performance of each constituent selection hyper-heuristic. The level of generality that a hyper-heuristic can achieve has always been of interest to the hyper-heuristic researchers. Hence, a variety of multi-stage hyper-heuristics based on the framework are not only applied to the real-world combinatorial optimisation problems of high school timetabling, multi-mode resource-constrained multi-project scheduling and construction of magic squares, but also tested on the well known hyper-heuristic benchmark of CHeSC 2011. The empirical results show that the multi-stage hyper-heuristics designed based on the proposed framework are still inherently general, easy-to-implement, adaptive and reusable. They can be extremely effective solvers considering their success in the competitions of ITC 2011 and MISTA 2013. Moreover, a particular multi-stage hyper-heuristic outperformed the state-of-the-art selection hyper-heuristic from CHeSC 2011.

Descripción

Esta tesis recoge el estado del arte de las hiperheurísticas y consta de cuatro partes: revisión literaria en el campo de las hiperheurísticas, análisis del dominio de las soluciones, teoría de hiperheurísticas multietapa, estado del arte de las hiperheurísticas multietapa. El documento comienza describiendo los conceptos generales de las hiperheurísticas y la relación con las metaheurísticas, a continuación, analiza el dominio de aplicación de las hiperheurísticas específicamente problemas de programación de horarios. Posteriormente aborda el problema de las hiperheurísticas multietapa, que se diferencian de las hiperheurísticas de etapa simple en que estas últimas utilizan una heurística para el proceso de selección y una heurística para el proceso de validación, mientras que las primeras emplean más de una heurística durante el proceso de selección por lo que se hace necesaria adicionar una capa más al modelo de tal forma que pueda tomar decisiones sobre el comportamiento de la heurística. La última parte del documento consiste en un análisis de la literatura y el estado del arte en las hiperheurísticas multietapa.

Aportes a la Tesis

El documento incluye descripción de material de calidad acerca del estado del arte, así como conceptos básicos y avanzados sobre las hiperheurísticas, además se centra en el problema de asignación de horarios escolares y trata de forma amplia el tema. En la parte del análisis del dominio, muestra distintos tipos de modelado matemático del problema y en el tema de interés, define la variable y el modelo matemático, expresa las heurísticas de alto y bajo nivel empleadas y los algoritmos y cálculos empleados. Adjunta definiciones de casos y posibilidades, que después

retoma para abordarlos con una estrategia de hiperheurística multietapa. También expone información sobre el estado del arte de esta hiperheurística pero con enfoque multietapa.

Título Aplicación de un algoritmo ACO al problema de flowshop flexible con tiempos de preparación dependientes de la secuencia y minimización de la tardanza total

Fuente Ingeniare. Revista chilena de ingeniería, 2016

Autor Eduardo Salazar Hornig, (E-mail: esalazar@udec.cl), Departamento de Ingeniería Industrial. Universidad de Concepción.

Deisy Torres Pérez, (deisytorres@udec.cl), Departamento de Ingeniería Industrial. Universidad de Concepción.

Palabras Clave

Flowshop flexible, colonia de hormigas, búsqueda en vecindad, intercambio de pares.

Resumen

Este estudio considera el problema de flowshop flexible con tiempos de setup anticipatorios dependientes de la secuencia y minimización de la tardanza total. Se propone un algoritmo de optimización de colonia de hormigas ACS (Ant Colony System), hibridizado con una búsqueda en vecindad de intercambio de pares, evaluado en un conjunto de problemas de prueba generados en estudios anteriores. Los resultados se comparan con otros métodos de solución, presentando el algoritmo propuesto mejores resultados.

Descripción

En la Industria llama mucho la atención la Planificación y Control de la Producción (Flow Shop) y es considerado como un problema NP hard, ya que muchos administradores de procesos productivos les interesa el aumento en su producción y la utilización eficiente de sus recursos . Se debe tener en cuenta que dicha programación debe ser simple, clara, fácil de comprender, flexible y realista, Los autores de este artículo presenta un algoritmos híbrido para una Posible solución en el problema flow shop, realizando pruebas y obteniendo buenos resultados

Aporte a la Tesis

Dado que el problema de flow shop flexible es considerado un NP hard , es abordado en la literatura con diferentes estrategias como la utilización de métodos analíticos, heurísticas, metaheurísticas tales como : algoritmos genéticos, recocido simulado y búsqueda tabú; los autores de este artículo le apuestan a la construcción de un híbrido entre optimización de colonia de hormigas con búsqueda en vecindad de intercambio de pares; aplicados al problema FFS, luego comparan este algoritmo con métodos como el algoritmos genético básico, algoritmos genético mejorado, AGB_EDD y AGB_slack, luego de realizar pruebas se puede comprobar que el híbrido ACS con búsqueda en vecindad IP muestra un rendimiento significativo superando al resto. es interesante ver cómo la combinación de diferentes técnicas se puede potencializar un mejor resultado.

Título A Review of Hyper-Heuristic Frameworks

Fuente <https://www.researchgate.net/publication/261361800>, 2014

Autor Patricia Ryser-Welch, York University,England
(patricia.ryser-welch@york.ac.uk)

Julian F. Miller, York University, England (julian.miller@york.ac.uk).

Palabras Clave

Hyper-heuristics, evolutionary computation, metaheuristics

Resumen

Hyper-heuristic frameworks have emerged out of the shadows of meta-heuristic techniques. In this very active field, new frameworks are developed all the time. Shared common features that help to classify them in different types of hyper-heuristic. Similarly to an iceberg, this large subfield of artificial intelligence hide a substantial amount of bio-inspired solvers and many research communities. In this paper, the tip of the iceberg is reviewed; recent hyper heuristic frameworks are surveyed and the overall context of the field is presented. We believe its content complements recent reviews and offers another perspective of this important and developing field to the research community. Some hyper-heuristic frameworks tend to be largely constrained and prevent the state-of-the-art algorithms being obtained. We suggest in addition to relaxing constraints together with analysis of the evolved algorithms may lead to human-competitive results.

Descripción

Los frameworks nacen entre las sombras de las diferentes metaheurísticas realizadas por una amplia comunidad de investigadores, esta herramienta permite la clasificación de diferentes tipos de hiper heurísticas, en este artículo se revisando la punta del iceberg, examinando los framework recientes y el contexto general sobre este campo, en algunos casos estos framework pueden ser limitados, pero aun así proporcionando una excelente herramienta para la comunidad de investigadores, mostrándoles otra perspectiva ya que provee una gran cantidad de soluciones

inspirados por ellos mismos; se puede ver la variedad de modelos y framework hiperheurísticos, mostrando su objetivo principal y los problemas o dificultades que tienen.

Aporte A la Tesis

El conocer que hay desarrollados frameworks para la clasificación de las diferentes modelos hiperheurísticos es un gran aporte; nos permite explorar y visualizar las distintas formas en las que se pueden construir estas. Los autores en este artículo recopilan información de cómo jugando con las diferentes metaheurísticas ya existentes, se pueden alcanzar un mejor resultado; sin embargo, también enseñan sus limitaciones, pero para la comunidad de investigación en hiper heurísticas es una aporte significativo para continuar la discusión y el mejoramiento de estos algoritmos, tema en el cual nosotros vemos un campo fértil a desarrollar.

Título	Un algoritmo eficiente para problemas single machine con tiempos de procesamiento difusos
Fuente	Revista Cubana de Ciencias Informáticas Vol. 10, No. 4, Octubre-Diciembre, 2016
Autor	Edwin Lazo Eche , Universidad Nacional de Piura. Urb. Miraflores s/n, Castilla, Piura, Perú. lazoedwin88@gmail.com Flabio Gutiérrez Segura, Universidad Nacional de Piura. Urb. Miraflores s/n, Castilla, Piura, Perú, flabio@unp.edu.pe Edmundo Vergara Moreno, Universidad Nacional de Trujillo. Calle Diegode Almagro 344, Trujillo, Perú. evergara@unitru.edu.pe

Palabras Clave

scheduling difuso, single machine, números difusos, intervalo esperado.

Resumen

Los modelos clásicos de scheduling en single machine han sido estudiados con diversas características de procesamiento y funciones objetivo. La mayoría de técnicas propuestas para resolver este tipo de problemas no consideraron la naturaleza imprecisa de algunas variables que intervienen en su formulación. En este trabajo se propone un algoritmo para el cálculo del máximo tiempo de culminación ponderada en problemas de single machine

con tiempos de procesamiento difusos y asignación de prioridades (pesos) a los trabajos. La metodología planteada representó la imprecisión de los tiempos de procesamiento mediante números difusos triangulares y utilizó una medida de comparación robusta y flexible basada en el concepto del intervalo esperado de un número difuso. Se obtuvo un schedule óptimo y robusto que soporta variaciones en los tiempos de procesamiento. El schedule óptimo se

representó en forma gráfica mediante el diagrama de Gantt difuso. El algoritmo propuesto resulta ser eficiente en pruebas para un determinado conjunto de trabajos en la búsqueda de un schedule óptimo para el cálculo del máximo tiempo de culminación ponderada.

Descripción

En la literatura se puede encontrar los diversos casos en donde los métodos clásicos del scheduling en single machine, es estudiado con diversas características de procesamiento y funciones objetivo, sin considerar en muchos casos la naturaleza

imprecisa de algunas variables que intervienen en la formulación del problema, en este artículo los autores proponen una solución a este problema mediante la incorporación de lógica difusa para una mejor optimización del algoritmo.

Aporte A la Tesis

El problema scheduling es utilizada en muchas áreas como son: asignación de Horarios, planes de aterrizaje o despegue de aviones, atraques de barcos etc., en este trabajo se enfoca específicamente sobre scheduling en single machine, mostrando como mediante la utilización de lógica difusa, específicamente con el manejo de números difusos triangulares se puede obtener un algoritmo para el cálculo del máximo tiempo de culminación ponderada del problema con tiempos de procesamientos difusos y asignación de prioridades (pesos) a los trabajos; permitiendo manejar la imprecisión en los tiempos de procesamiento de los trabajos; tema interesante y del cual se puede obtener una nueva metodología a la hora de construir un algoritmos para la asignación de tareas .

Título	Performance of Selection Hyper-heuristics on the Extended HyFlex Domains
Fuente	Computer and information sciences: 31st International Symposium, ISCIS 2016, Kraków, Poland, October 27–28, 2016, proceedings. Communications in computer and information science (659). Springer, pp.154-162. ISBN 978-3-319-47217-1
Autor	Alhanof Almutairi , Ender Özcan, Warren G. Jackson ASAP Research Group, School of Computer Science, University of Nottingham, Wollaton Road, Nottingham NG8 1BB, UK Ahmed Kheiri

Operational Research Group, School of Mathematics, Cardiff University,
Senghennydd Road, Cardiff CF24 4AG, UK

Palabras clave

Metaheuristic, parameter control, adaptation, move acceptance , optimisation.

Resumen

Selection hyper-heuristics perform search over the space of heuristics by mixing and controlling a predefined set of low level heuristics for solving computationally hard combinatorial optimisation problems. Being reusable methods, they are expected to be applicable to multiple problem domains, hence performing well in cross-domain search. HyFlex is a general purpose heuristic search API which separates the high level search control from the domain details enabling rapid development and performance comparison of heuristic search methods, particularly hyper-heuristics. In this study, the performance of six previously proposed selection hyper-heuristics are evaluated on three recently introduced extended HyFlex problem domains, namely 0–1 Knapsack, Quadratic Assignment and Max-Cut. The empirical results indicate the strong generalising capability of two adaptive selection hyper-heuristics which perform well across the ‘unseen’ problems in addition to the six standard HyFlex problem domains.

Descripción

El artículo se centra en el proceso de selección de las secuencias de heurísticas empleadas en el proceso de búsqueda de la solución, estas secuencias pueden ser halladas ya sea empleando una heurística como por ejemplo seleccionar al azar la

heurística a aplicar, o seleccionarlás todas. Se exponen diferentes formas de selección y se asocia una descripción del contexto donde fueron empleadas. Luego el artículo muestra datos de las secuencias heurísticas empleadas y los resultados obtenidos. Es importante mencionar que la implementación se realizó usando la librería HyFlex, la cual es una librería construida en Java diseñada para implementar hiperheurísticas.

Aportes a la Tesis

La información que brinda el artículo en cuanto a diferentes tipos de selección que es posible emplear durante el proceso de selección en la hiperheurística resulta muy útil como aporte metodológico para la implementación ya que menciona métodos claramente definidos y además menciona alternativas nuevas que investigadores en el área han venido probando como es el usar cadenas ocultas de markov para hallar patrones ocultos que mejoran el desempeño, lo que desemboca en la exposición de un método de ponderación que permite evaluar la calidad de diferentes implementaciones. También es un aporte la construcción de estos modelos empleando HyFlex, una librería en Java, que si bien es cierto no será usada durante la implementación de nuestro modelo si es importante mencionarla dentro del estado del arte ya sea como tecnología asociada a los modelos hiperheurísticos o considerarla una herramienta creada para tal fin.

CAPÍTULO 3

MARCO TEÓRICO

3.1 Hiperheurística

Una hiperheurística es un método de búsqueda o mecanismo de aprendizaje para seleccionar o generar heurísticas que resuelvan problemas de alta dificultad computacional.

Es un tipo de metodología de búsqueda de propósito general con el objetivo de automatizar el diseño y automatización de métodos heurísticos sobre las metaheurísticas, el objetivo es mejorar su desempeño.

Una hiperheurística también es un método de búsqueda para seleccionar o generar (combinando o adaptando) heurísticas o componentes heurísticos con el objetivo de resolver un rango de problemas de optimización.

3.1.1 Características

Operan sobre el espacio de búsqueda de las heurísticas en vez de actuar directamente sobre el espacio búsqueda de la solución. Es decir que la hiperheurística puede dilatar o contraer el espacio donde la heurística está actuando, se centra en el espacio actual de explotación.

Es posible utilizar heurísticas ya existentes o generadas al interior de la hiperheurística, es decir que el método de búsqueda de la solución puede cambiar durante el proceso.

La razón de esto es que la hiperheurística se diseña para que explote las ventajas de un método heurístico y evite las debilidades que conlleva emplearlo.

Una de las características más importantes de la hiperheurística es que no requiere ningún conocimiento específico durante la búsqueda en el espacio heurístico, por lo que son reutilizables.

Por principio una hiperheurística debe ser fácil de implementar, de descargar y de usar (fácil, barato, rápido).

La hiperheurística de ser aplicable a un espectro de problemas del mundo real, se comporta mejor en modelos no idealizados ya que está pensada para adaptarse a los cambios.

Es extremadamente deseable usar técnicas de ciencia de datos como aprendizaje de máquina, por ejemplo, para dar respuesta a las necesidades de aprendizaje del modelo.

3.1.2 Clasificación

Es importante observar que diferentes heurísticas poseen diferentes fortalezas y debilidades, la clave está en conocer y establecer heurísticas para:

- ✓ Transformar el estado del problema (estrategia constructiva)
- ✓ Realizar y mejorar los pasos o etapas de la solución del problema

(estrategia perturbativa)

Las hiperheurísticas suelen clasificarse en dos grandes grupos[2] :

- ✓ Basados en la naturaleza del espacio heurístico de búsqueda
- ✓ Basadas en la retroalimentación durante el aprendizaje

Los basados en la naturaleza del espacio heurístico de búsqueda se caracterizan por:

i) Selección heurística: metodología para seleccionar una heurística existente. En este proceso se lanza una heurística y se toma el mejor resultado.

ii) Generación heurística: metodología para generar nuevas heurísticas desde componentes de heurísticas ya existentes.

Las basadas en retroalimentación son aquellas que toman información arrojado por el proceso de búsqueda, sin embargo, hay hiperheurísticas sin aprendizaje y estas en particular no usan ningún tipo de retroalimentación.

De acuerdo con el origen de la retroalimentación durante el aprendizaje este puede ser en línea o fuera de línea.

El aprendizaje en línea tiene lugar cuando el algoritmo está resolviendo una instancia del problema.

El aprendizaje fuera de línea consiste en recolectar información a manera de reglas o programas para establecer un conjunto de instancias de entrenamiento que son útiles para la generalización desde el proceso de resolución en instancias no visibles.

En ambos casos se pueden generar heurísticas constructivas o perturbativas.

3.2 Bases de datos NoSQL

Debido a la naturaleza del problema es necesario emplear algún repositorio de información, las opciones son archivos, bases de datos Sql y bases de datos NoSql.

Los archivos requieren invertir tiempo en gestión de los datos y poseen fallas en integridad lógica de los datos, por lo que debe descartarse esta modalidad salvo en el caso que deseemos almacenar salidas del sistema.

Las bases de datos Sql en el problema que tratamos requeriría una cantidad de tablas considerable, además de la una estructura de tablas que implicaría un modelo de consultas complejo ya que la información a tratar para que sea útil tendría que proceder de cruce entre varias tablas y sería necesario implementar un modelo CRUD (Create, Read, Update, Delete) durante el proceso de interacción entre la hiperheurística y el repositorio de datos.

Finalmente se tiene las bases de datos NoSql, que se caracterizan por un modelo simplificado llave valor, y una organización laxa pero que puede garantizar integridad de los datos.

Otra razón importante es la compatibilidad del modelo llave valor con los diccionarios en Python que hacen casi transparente el uso de los datos en las colecciones.

A continuación, se explica en que son las bases de datos no relacionales (NoSql) ya que en el modelo a implementar se tiene una base de datos particionada y un modelo de almacenamiento llave valor.

NoSQL, es hablar de cantidades de información y datos no estructurados que no se pueden almacenar en las que las bases de datos relacionales; son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos, sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos.

Las Bases de Datos NoSQL permiten escalabilidad y rendimiento a miles de usuarios concurrentes y con millones de consultas diarias.

3.2.1 Teorema CAP

El teorema CAP ha sido ampliamente adoptado por las grandes compañías de internet, al igual que por la comunidad NoSQL.

Las siglas CAP hacen referencia a:

- ✓ **Coherencia:** En sistema distribuido, habitualmente se dice que se encuentra en un estado consistente si, después de una operación de escritura, todas las operaciones de lectura posteriores son capaces de ver las actualizaciones desde la parte del sistema desde la que están leyendo.
- ✓ **Disponibilidad (Availability):** La alta disponibilidad se produce cuando el sistema ha sido diseñado e implementado de modo que se pueda continuar operando (lecturas, escrituras), incluso después de que un nodo quede indisponible, o que algunas partes de hardware tengan que ser retiradas, debido a errores o actualizaciones.
- ✓ **Tolerancia a Particiones (Partition Tolerance):** Entendido como la habilidad de un sistema de tener diferentes regiones o divisiones lógicas en la red, y de ser capaz de seguir funcionando aunque una de estas partes quede inaccesible durante un tiempo.



Ilustración 1 Triángulo CAP Fuente: DataStax.com

Los modelos de transacciones también pueden diferir de un modelo de base de datos a otro. Los modelos transaccionales a tener en cuenta son:

Modelo transaccional ACID (Atomicity, Consistency, Isolation and Durability):

- ✓ Atomicidad: implica que al realizar una transacción todo debe ocurrir correctamente o simplemente la transacción no se realiza, es decir, atomicidad en las transacciones.
- ✓ Consistencia: asegura que cualquier transacción en una base de datos debe partir de un estado válido y llegar a otro estado igualmente válido.
- ✓ Aislamiento: propiedad transaccional donde se garantiza que una operación que acceda al sistema de manera concurrente junto a otras preserve sus propiedades.
- ✓ Durabilidad: asegura que una vez una transacción se realiza queda almacenada en el sistema.

LAS bases de datos NoSql renuncian a las propiedades ACID y cogen las propiedades BASE (Basic Availability Soft-state Eventual consistency):

- ✓ Disponibilidad básica: cada solicitud garantiza una respuesta de ejecución exitosa o ejecución fallida.
- ✓ Estado blando: el estado del sistema puede cambiar con el tiempo, a veces sin ninguna entrada, por consistencia eventual.
- ✓ Consistencia eventual: una base de datos puede ser inconsistente en el momento, pero lo será con el tiempo.

3.3 Unqlite

Unqlite es una base de datos embebida NoSql que se enlaza directamente a nuestro código fuente, es decir que su funcionamiento está ligado a la aplicación, y por lo tanto sus servicios comienzan y terminan con la aplicación.

Las bases de datos embebidas son aquellas que dependen de una aplicación y en el modelo planteado, creadas en tiempo de ejecución.

Esta nos permitirá almacenar información necesaria para que la hiperheurística aprenda y posteriormente para procesar nueva información. Estas bases de datos serán la 'memoria' de la hiperheurística.

Sus principales características son:

- ✓ Embebida, cero configuraciones
- ✓ Estándar transaccional ACID
- ✓ Almacenamiento en memoria o en archivo
- ✓ Almacenamiento clave – valor
- ✓ Soporte de cursores
- ✓ Almacenamiento en formato JSON
- ✓ Modelo de procesamiento de hilos seguro
- ✓ Almacenamiento de hasta un Terabyte (1 TB)
- ✓ Compatible con Python 2 y Python 3
- ✓ Soporta scripts JX9 desde Python

Para su uso desde Python requiere la instalación de librería unqlite que depende a su vez de la librería Cython.

3.4 Algoritmos genéticos

Los algoritmos genéticos son métodos adaptativos que pueden utilizarse para resolver todo tipo de problemas complejos de búsqueda y optimización, lo que lo convierte en una metaheurística. Es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años. El algoritmo Genético usa una población de individuos, que en los problemas combinatorios representa un conjunto de configuraciones, para resolver un problema de optimización complejo [4]; un algoritmos genéticos opera como se explica en el algoritmo en la tabla 2.

Los pasos en el al interior del ciclo, en conjunto, son conocidos como ciclo generacional, También es necesario mencionar que existe una equivalencia entre los términos usados en genética y en un problema de optimización matemática

Algoritmo Genético
<ol style="list-style-type: none"> 1. Genera una población Inicial, después de elegir el tipo de codificación, para representar cada configuración. 2. Mientras criterio de parada no se cumpla <ol style="list-style-type: none"> 2.1 Calcula la función objetivo de cada configuración de la población y almacena la incumbente (la mejor configuración encontrada durante el proceso). 2.2 Realizar selección 2.3 Realizar combinación 2.4 Realizar mutación 2.5 Generar nueva población

Tabla 2 Algoritmo Genético General

La selección es método mediante el cual el algoritmo genético empareja los individuos con el fin de crear nuevos individuos hijos y de esta forma mejorar la carga genética, algunos de las más comunes son:

- ✓ selección basada en promedio: los individuos son emparejados con individuos cuyo promedio en la función de adaptación es similar.
- ✓ Selección por torneo: es los individuos se emparejan al azar con otros y se reproduce el mejor.

La recombinación o cruce también posee varios métodos, a grandes rasgos se clasifican en cruzamiento de un punto o de dos puntos.

Los de un punto definen una posición de división y sobre este realizan el intercambio genético. En el de dos puntos la idea es fijar dos puntos de tal manera que la carga genética del individuo queda segmentada en tres regiones, donde una de ellas es usada para intercambio genético.

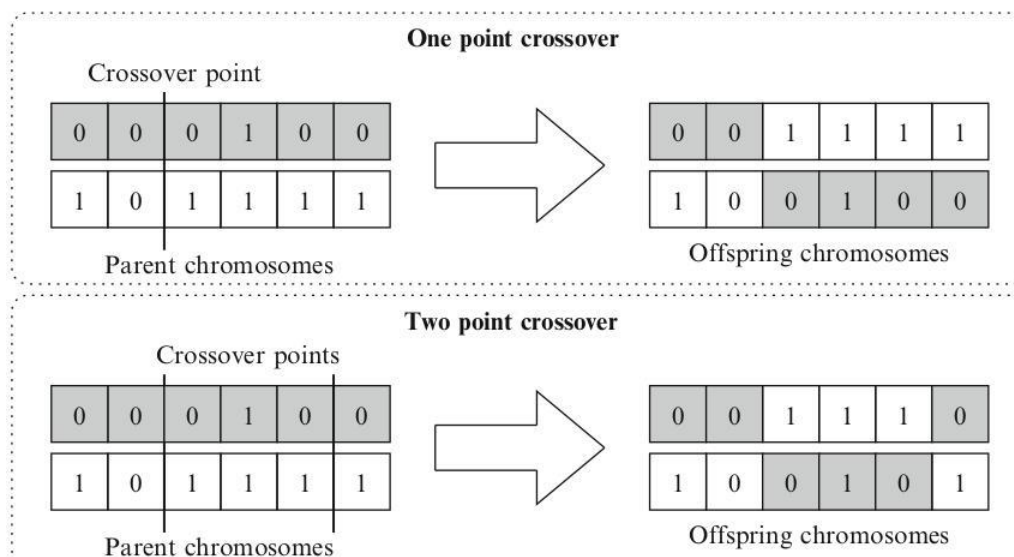


Ilustración 2 Modelo de Cruce de un Punto y Dos Puntos [18]

Otros métodos de combinación son:

✓ PMX (Partially Mapped Crossover) o cruce por emparejamiento parcial es un método en el cual crea una descendencia a partir de una franja seleccionada en cada uno de los padres, el intercambio de los individuos ocurre tratando de preservar el orden y la posición de los genes buscando mantener la coherencia.

✓ CX (Cycle crossover) o cruce básico de ciclos, crea un descendiente a partir de una franja seleccionada para intercambio de cada uno de los padres, de tal manera que cada posición es ocupada por el correspondiente elemento en cada uno de los padres.

La mutación es el proceso por el cual el algoritmo genético busca escapar de óptimos locales al alterar de forma aleatoria un individuo en su carga genética, al igual que en la naturaleza su ocurrencia es muy baja por lo que suele implementarse un lanzamiento aleatorio de baja ocurrencia con el que se busca simular el evento.

3.5 UML

La construcción del prototipo requiere una planificación que permita implementar de forma coherente cada uno de los componentes que requiere una hiperheurística sin perder de vista que se trata de un modelo que a pesar de ser en si mismo un sistema se comporta como un actor, ya que interactúa con el sistema mediante entradas que genera y requiere de las salidas para refinar su comportamiento.

El lenguaje de modelado unificado (UML) es un lenguaje de modelado visual de propósito general, indicador que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Captura decisiones y comprensión sobre los sistemas que deben ser construido. Se usa para comprender,

diseñar, explorar, configurar, mantener y controlar la información sobre dichos sistemas. Está destinado para su uso con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicaciones y medios. El lenguaje de modelado está destinado a unificar experiencias pasadas sobre técnicas de modelado e incorporar las mejores prácticas de software actuales en un enfoque estándar.

UML incluye conceptos semánticos, notación y pautas. Tiene partes estáticas, dinámicas, ambientales y organizacionales. Se pretende que sea compatible con herramientas de modelado visual interactivo que tengan generadores de códigos y redactores de informes. La especificación UML no define un proceso estándar, pero tiene la intención de ser útil con un proceso de desarrollo iterativo. Está diseñado para admitir la mayoría de los procesos de desarrollo orientados a objetos existentes.

UML es una herramienta que permite mediante diferentes Diagramas Modelar los requerimientos y procesos de un software.

A continuación, se exponen los diagramas utilizados en el desarrollo del software.

3.5.1 Diagrama de Clases

Los diagramas de clases representan las estructuras estáticas de un sistema, incluidas sus clases, atributos, operaciones y objetos. Un diagrama de clases puede mostrar datos computacionales u organizacionales en la forma de clases de implementación y clases lógicas, respectivamente. Puede haber superposición entre estos dos grupos.

- ✓ Las clases se representan con una forma rectangular dividida en tercios. La sección superior muestra el nombre de la clase, mientras que la

sección central contiene los atributos de la clase. La sección inferior muestra las operaciones de la clase (también conocidas como métodos).

✓ Agrega formas de clases a tu diagrama de clases para modelar la relación entre esos objetos. Además, podría ser necesario que agregues subclases.

✓ Usa líneas para representar asociación, traspaso, multiplicidad y otras relaciones entre clases y subclases. Tu estilo de notación preferido informará la notación de estas líneas.

Se emplearán para describir las clases necesarias para construir la hiperheurística, modelar las entidades asociadas al problema y para implementar el algoritmo genético.

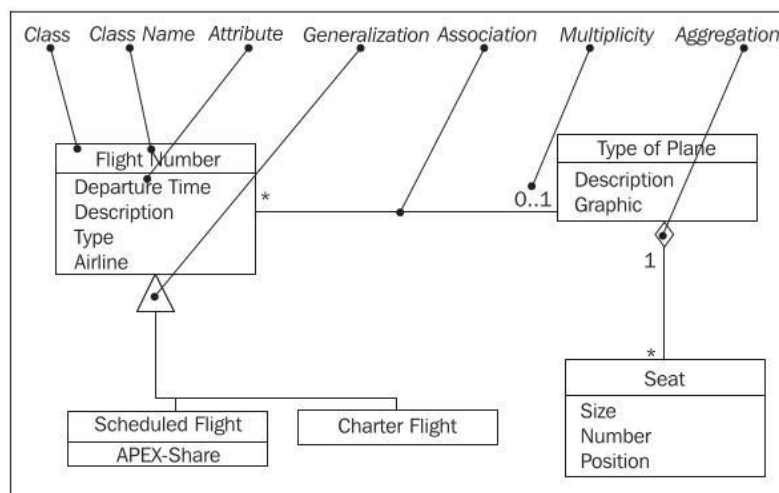


Ilustración 3 Diagrama de Clases [12]

3.5.2 Diagrama de caso de uso

Un caso de uso es una lista de pasos que definen la interacción entre un actor (un humano que interactúa con el sistema o un sistema externo) y el sistema

propriadamente dicho. Los diagramas de casos de uso representan las especificaciones de un caso de uso y modelan las unidades funcionales de un sistema. Estos diagramas ayudan a los equipos de desarrollo a comprender los requisitos de su sistema, incluida la función de la interacción humana en el mismo y las diferencias entre diversos casos de uso. Un diagrama de caso de uso podría mostrar todos los casos de uso del sistema o solo un grupo de casos de uso con una funcionalidad similar.

- ✓ Para iniciar un diagrama de casos de uso, agrega una forma ovalada en el centro del dibujo.
- ✓ Escribe el nombre del caso de uso dentro del óvalo.
- ✓ Representa a los actores con una figura humana cerca del diagrama, luego usa líneas para modelar las relaciones entre los actores y los casos de uso.

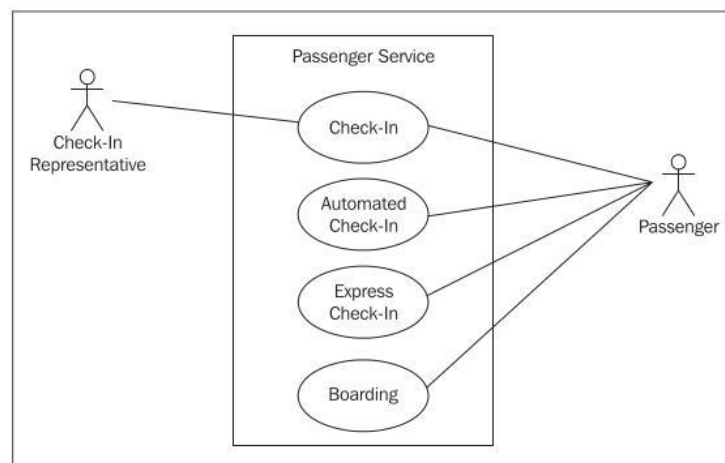


Ilustración 4 Diagrama de Casos de Uso [12]

En el proceso de construcción del modelo, el diagrama de casos de uso servirá para explicar los requerimientos que justifican los distintos componentes de la implementación.

3.5.3 Diagrama de componentes

El diagrama de componentes tiene como finalidad describir la implementación lógica de un sistema.

- ✓ Se agregan componentes del sistema
- ✓ A cada componente se le deben agregar las interfaces que se requieran o provean, un componente puede:
 - Proveer interfaces
 - Requerir interfaces
- ✓ Un componente puede tener una relación de dependencia con otro.

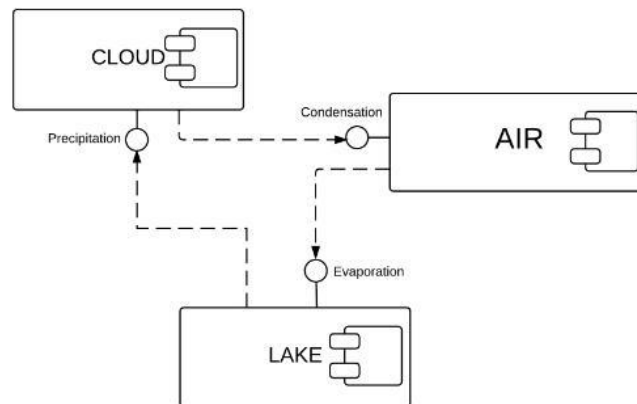


Ilustración 5 Diagrama de Componentes fuente : [21]

CAPÍTULO 4

METODOLOGÍA

4.1 Heurísticas

Para construir una hiperheurística es necesario establecer o definir cuáles son las heurísticas que desean emplearse sobre la metaheurística o metaheurísticas que intentan resolver el problema. Las heurísticas son los métodos que un experto podría usar durante el proceso de resolución del problema y para cada heurística es posible que posea su propio modelo de evaluación de desempeño, aunque el objetivo en apariencia sea el mismo.

También se deben definir algunas restricciones que aplican para todas las heurísticas, estas son:

- ✓ Número máximo de horas por docente es 22 (requisito de ley).
- ✓ Número máximo de horas por semana por curso es de 30 horas (requisito de ley).
- ✓ Un grado posee varios cursos, por ejemplo, el grado seis posee 4 cursos.
- ✓ Un docente posee varias asignaturas.
- ✓ Un docente no puede estar asignado el mismo día y la misma hora a más de un curso, esto es a un docente no se le deben cruzar horas.
- ✓ Una asignatura tiene asociada una intensidad horaria por grado, por ejemplo dibujo 2 horas por semana en grado sexto.

Método de evaluación:

- ✓ Cantidad de horas programadas por semana por grupo \leq máximo de horas por semana
- ✓ Cantidad de horas programadas por docente \leq máximo de horas de carga docente por semana
- ✓ Cantidad de horas cruzadas por docente = 0
- ✓ Cantidad de horas programadas por asignatura \leq intensidad horaria esperada por asignatura

En los numerales siguientes se describen las posibles heurísticas de bajo nivel que es posible emplear.

En todos los modelos siguientes se cumple la asociación docente – área – asignatura, y se debe diferenciar el concepto de grado escolar y grupo, este último es una asociación de estudiantes del mismo grado pero que no comparten el mismo espacio físico, por ejemplo, 6A y 6B, pero se tratará la relación grado grupo como grupo pues es más general y permite diferenciar y delimitar los espacios de asignación de mejor forma.

4.1.1 Programación basada en áreas (áreas primero).

En este modelo se toman las áreas, luego se buscan las asignaturas asociadas a cada área y finalmente se asociando a cada asignatura en cada grupo un docente.

Este proceso está condicionado por las restricciones iniciales y está sujeto al método de evaluación descrito al inicio.

4.1.2 Programación basada en intensidad horaria (asignaturas con más intensidad horaria primero)

En este modelo heurístico el sistema toma las asignaturas, se ordenan de mayor a menor de tal forma que aquella con la mayor carga horaria esperada sea primero.

Luego para cada asignatura de este conjunto ordenado se busca en cada grupo y de encontrarse se le asocia un docente.

Las restricciones de este modelo son las generales y está sujeto al método de evaluación descrito al inicio.

4.1.3 Programación basada en docentes (docentes primero)

En este modelo para cada docente en el grupo de docentes, se busca en cada grupo las asignaturas asociadas a su área de trabajo, de encontrarse, se asocia a la asignatura el docente.

Las restricciones de este modelo son las generales y está sujeto al método de evaluación descrito al inicio.

4.1.4 Programación basada en grados (primero los grados)

En este modelo para cada grado se toman los grupos asociados a este grado, por ejemplo los grados 6, y para cada grupo asociado a este grado se toman las asignaturas y se les asocia un docente.

Las restricciones de este modelo son las generales y está sujeto al método de evaluación descrito al inicio.

4.2 Modelo del sistema

4.2.1 Casos de Uso

Los casos de uso serán descritos empleando el siguiente formato:

Código	Código de identificación del caso de uso
Nombre	Nombre del caso de uso
Actor principal	Actor principal en el caso de uso
Actores secundarios	Actores adicionales que intervienen en el caso de uso
Precondiciones	Descripción de condiciones previas necesarias para completar las tareas y alcanzar el objetivo del caso de uso
Objetivo	Salida esperada del sistema que puede ser alcanzada mediante el proceso descrito
Proceso	Conjunto de actividades necesarias para alcanzar un objetivo
Extensiones	Tareas alternativas que pueden aparecer

Tabla 3 Nombre del Caso de Uso

Se puede observar el diagrama de casos de uso y la descripción de los casos de uso.

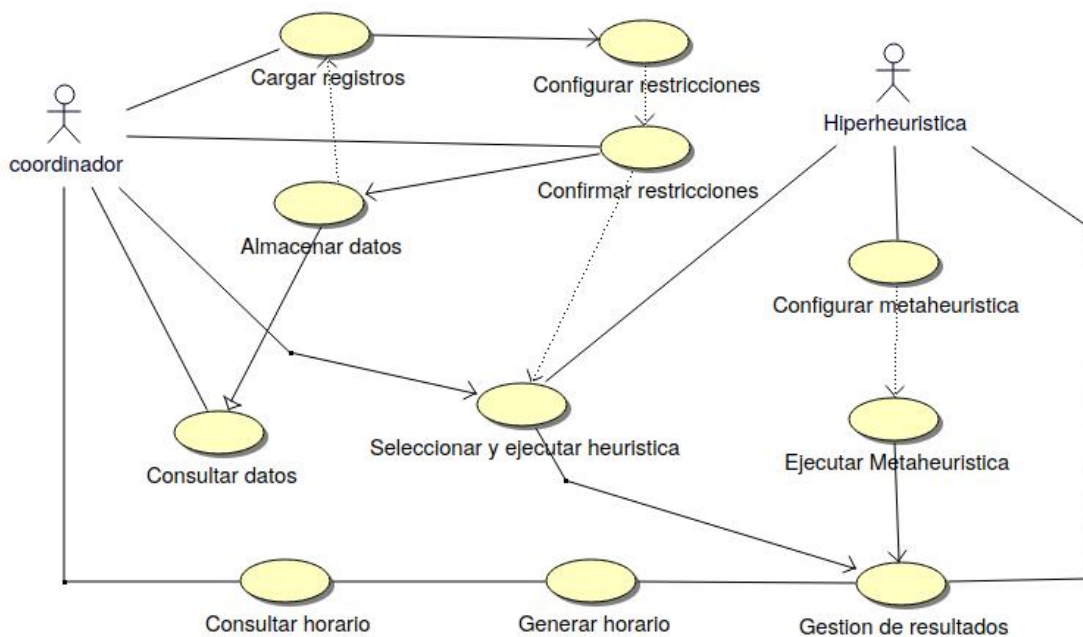


Ilustración 6 Diagrama de Casos de Uso

Código	CU-001
Nombre	Cargar registros
Actor principal	Coordinador
Actores secundarios	No aplica
Precondiciones	Archivos csv (archivo separado por comas) con información de docentes, grupos y carga horaria
Objetivo	Crear registros de tipo clave valor y almacenar en base de datos no relacional con la información inicial del problema
Proceso	
<ol style="list-style-type: none"> 1. Usuario selecciona archivos de extensión csv (archivos separados por comas). 2. Sistema toma archivos y descompone en registros clave valor. 3. Crea base de datos no sql y almacena registros con información de carga horaria de docentes, asignaturas y grupos. Esta base de datos almacena la información inicial del problema. 4. Generar archivo con extensión txt o csv con registros de tipo clave valor, contiene la colección de datos que relaciona área - asignatura. 	
Extensiones	Ninguna

Tabla 4 Cargar Registros

Código	CU-002
Nombre	Configurar restricciones
Actor principal	Sistema
Actores secundarios	No aplica
Precondiciones	Base de datos con información inicial de docentes, áreas, asignaturas y grupos
Objetivo	Crear archivos con extensión txt o csv que contengan información de docentes, áreas, asignaturas y grupos
Proceso	
<ol style="list-style-type: none"> 1. El sistema extrae información de la base de datos de registros iniciales. 2. El sistema generar archivos con extensión txt o csv con registros de tipo clave valor, cada archivo es una colección de datos que relacionan docente - asignatura, 	

docente - carga, área - docente, docente – limite, grado – grupos, grupo – vacío. Estos archivos son almacenados en una carpeta de restricciones.
Extensiones
Ninguna

Tabla 5 Configurar Restricciones

Código	CU-003
Nombre	Confirmar restricciones
Actor principal	Coordinador
Actores secundarios	No aplica
Precondiciones	Archivos que almacenan colecciones de datos que relacionan docentes, áreas, asignaturas y grupos. Carpeta de restricciones debe existir.
Objetivo	Permitir que usuario observe y modifique los parámetros iniciales de docentes, áreas, grados y asignaturas.
Proceso	
<ol style="list-style-type: none"> 1. Usuario Analiza archivos con registros clave valor de carga horaria por cada grupo, limite de carga de docentes, docente y asignaturas asociadas, áreas y las asignaturas asociadas, área y docentes asociados, grados y grupos, horarios que deben permanecer vacíos. 2. El coordinador realiza ajustes necesarios respecto a las áreas en cuanto a asignaturas y docentes. 3. El coordinador realiza ajustes necesarios respecto a los docentes en cuanto a su carga previa y los límites de carga horarios permitidos. 4. El coordinador realiza ajustes necesarios respecto a los grados y la carga horaria de las asignaturas por cada grado y grupos por cada grado. 5. El coordinador realiza ajustes necesarios respecto a los horarios que deberán ser vacíos. 6. Una vez el coordinador revisa y ajusta los valores de cargas y asignaturas confirma que las restricciones son adecuadas. 	
Extensiones	Ninguna

Tabla 6 Confirmar Restricciones

Código	CU-004
Nombre	Almacenar registros
Actor principal	Sistema
Actores secundarios	No aplica
Precondiciones	Archivos txt o csv con información de docentes, áreas, asignaturas y grados. Carpeta de restricciones debe existir. Restricciones deben estar confirmadas CU-003
Objetivo	Generar base de datos de soporte para la hiperheurística
Proceso	
<ol style="list-style-type: none"> 1. El sistema crea base de datos de soporte. 2. El sistema toma todos los archivos de extensión txt o csv de la carpeta de restricciones. 2. El sistema extrae los datos de tipo clave valor de los archivos y los crea colección de datos. 3. El sistema almacena colección de datos en la base de datos de soporte. 	
Extensiones	Ninguna

Tabla 7 Almacenar Registros

Código	CU-005
Nombre	Consulta de datos
Actor principal	Coordinador
Actores secundarios	Sistema
Precondiciones	Base de datos de soporte
Objetivo	Permitir que el usuario observe los datos almacenados en base de datos de soporte
Proceso	
<ol style="list-style-type: none"> 1. El sistema genera listado de colección en base de datos de soporte. 2. Coordinador selecciona colección. 3. El sistema extrae información de la base de datos de la colección seleccionada y la muestra en pantalla. 	
Extensiones	Ninguna

Tabla 8 Consulta de Datos

Código	CU-006
Nombre	Seleccionar y ejecutar heurística
Actor principal	Hiperheurística
Actores secundarios	Coordinador
Precondiciones	Base de datos de soporte.
Objetivo	Seleccionar heurística de bajo nivel para generar carga horaria
Proceso	
<ol style="list-style-type: none"> 1. La hiperheurística seleccionada heurística de bajo nivel. 2. La hiperheurística genera listas de objetos de grupos y docentes 3. La hiperheurística toma valores de carga en base de datos de soporte y configura listados de grupos y docentes. 4. La hiperheurística ejecuta heurística de bajo nivel. 5. La heurística de bajo nivel asocia asignaturas a grupos y docentes. 6. Se crean registros de asignación de carga horaria por grupos y por docentes. 	
Extensiones	
<ol style="list-style-type: none"> 5a. La hiperheurística muestra horas faltantes de asignaturas no programadas. 5b. El coordinador decide si esta se asigna a un docente en particular o permite que la hiperheurística seleccione el docente, en este caso el criterio de selección es docente con carga disponible. 5c. Se asocia asignatura faltante a un docente. 	

Tabla 9 Seleccionar y Ejecutar Heurística

Código	CU-007
Nombre	Gestión de resultados.
Actor principal	Hiperheurística
Actores secundarios	Sistema
Precondiciones	Seleccionar y ejecutar heurística CU-006, Ejecutar Metaheurística CU-008
Objetivo	Generar y analizar resultados
Proceso	
<ol style="list-style-type: none"> 1. La hiperheurística recopila datos de asignación de programación de horas esperada vs horas asignadas. 2 La hiperheurística recopila datos de docentes con carga horaria disponible después del proceso. 3. La hiperheurística recopila datos de horas extras no esperadas después del proceso. 	

Extensiones
<p>4. La hiperheurística toma datos de salida del algoritmo genético y analiza el resultado, si la función de adaptación (función fitness) cumple con restricciones, solución se almacena, de lo contrario se procede a ajustar algoritmo genético.</p> <p>5. Si la solución es adecuada se modifica el espacio de búsqueda del algoritmo genético, adaptando nuevos valores a los datos de grupos y/o de docentes.</p> <p>6. Si se ha encontrado la totalidad del horario terminar, en caso contrario regresar a configurar metaheurística CU-008</p>

Tabla 10 Gestión de Resultados

Código	CU-008
Nombre	Configurar metaheurística
Actor principal	Hiperheurística
Actores secundarios	No aplica
Precondiciones	Asignación de carga horaria completa. Seleccionar y ejecutar heurística CU-006. Gestión de resultados CU-007
Objetivo	Establecer valores iniciales para la metaheurística del algoritmo genético
Proceso	
	<p>1. La hiperheurística define la longitud del genoma de tal manera que sea de 0 hasta n-1 de longitud, donde n es el número de grupos de la institución.</p> <p>2. La hiperheurística establece cantidad de individuos en la población.</p> <p>2. La hiperheurística establece número de generaciones.</p>
Extensiones	
	<p>4. Se toman lecturas de arrojadas por la metaheurística de franja horaria asignada.</p> <p>5. Si salida esperada es inferior a 95% se deben ajustar variables y ejecutar de nuevo la metaheurística CU-009.</p>

Tabla 11 Configuración Metaheurística

Código	CU-009
Nombre	Ejecutar metaheurística
Actor principal	Sistema
Actores secundarios	No aplica
Precondiciones	Configurar metaheurística CU-008,
Objetivo	Ejecutar metaheurística para encontrar programación de horarios docentes
Proceso	
<ol style="list-style-type: none"> 1. Se define genoma de trabajo, el genoma es un espacio de asignación de identificación de docente (de 0 hasta n-1, donde n número de docentes) y cada espacio es un grupo de la institución. 2. Se establecen número de individuos por generación. 3. Se establece el número de generaciones que deberá procesar el algoritmo. 4. Se inicia algoritmo genético. 5. Se generan resultados por parte del algoritmo genético. 	
Extensiones	Ninguna

Tabla 12 Ejecutar Metaheurística

Código	CU-010
Nombre	Generar horario
Actor principal	Sistema
Actores secundarios	No aplica
Precondiciones	Gestión de resultados CU-007
Objetivo	Construir horario con información recopilada del proceso hiperheurístico.
Proceso	
<ol style="list-style-type: none"> 1. El sistema toma información de horarios encontrados por franja horaria. 2. El sistema organiza los horarios acordes a criterio temporal. 3. Se genera horario de asignaturas para todos los grupos. 4. Se genera horario de grupos para todos los docentes. 	
Extensiones	Ninguna

Tabla 13 Generar Horarios

Código	CU-011
Nombre	Consulta de horario
Actor principal	Coordinador
Actores secundarios	No aplica
Precondiciones	Información de horario recopilada CU-010
Objetivo	Crear interfaz maquina hombre para observar resultados
Proceso	<ol style="list-style-type: none"> 1. El sistema presenta opciones de visualización de datos 2. Coordinador selecciona opción de visualización 3. Sistema muestra datos de horarios
Extensiones	<ol style="list-style-type: none"> 4. Sistema genera archivos de horarios

Tabla 14 Consultar Horarios

4.2.2 Diagramas de clase

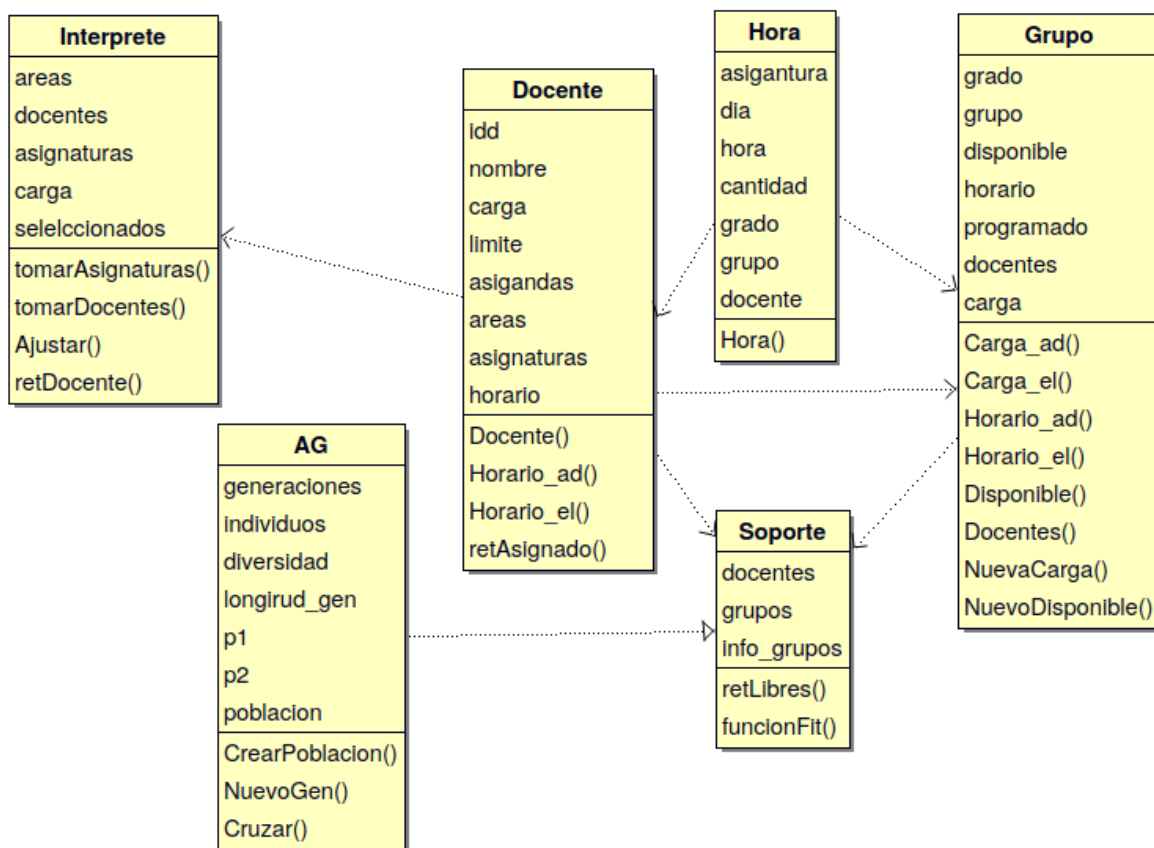


Ilustración 7 Diagrama de Clases

Las clases serán descritas usando el siguiente formato:

Nombre de la clase	
Atributos	
Métodos	

Tabla 15 Muestra

Nombre de la clase:	CL-001 Hora
Atributos	
Asignatura	Nombre de la asignatura
Dia	Dia de la semana programada
Hora	Hora programada
Cantidad	Cantidad de horas descritas
Grado	Grado asociado a la hora
Grupo	Grupo asociado a la hora
Docente	Número de identificación del docente asociado a la hora
Metodos	
Hora	Constructor, requiere nombre de la asignatura
Nombre de la clase:	CL-002 Docente
Atributos	
Idd	Número de identificación del docente
Nombre	Nombre del docente
Carga	Carga asociada al docente
Limite	Carga máxima establecida para el docente
Asignadas	Lista de horas, almacenas horas asignadas al docente
Áreas	Lista de áreas a las que pertenece el docente
Asignaturas	Lista de asignaturas que el docente puede dictar
Horario	Lista de horas asociadas al docente
Metodos	
Docente	Constructor, requiere número de identificación, nombre del docente y carga horaria
Horario_ad	Adicionar hora al horario, requiere hora y cantidad de las

	horas
Horario_el	Elimina una hora del horario, requiere hora y cantidad de horas
retAsignado	Retorna la cantidad de horas asignadas al docente

Tabla 16 Hora

Nombre de la clase:	CL-003 Grupo
Atributos	
Grado	Grado del grupo
Grupo	Identificador del grupo
Disponibile	Cantidad de horas no asignadas al grupo
Horario	Lista de horas con información de docente y cantidad semanal de horas asignadas
Programados	Lista de horas programadas para el grupo
Docentes	Lista de docentes asociados al grupo
Carga	Lista de horas esperadas por programar para el grupo
Metodos	
Grupo	Constructor, requiere grado y grupo
Carga_ad	Adiciona horas a la carga del grupo
CArga_el	Elimina horas de la carga del grupo
Horario_ad	Adicionar hora al horario, requiere hora y cantidad de la horas
Horario_el	Elimina una hora del horario, requiere hora y cantidad de horas
Disponibles	Asocia carga horaria a hora en la lista horario
Docentes	Almacena en la lista de docentes que se encuentran en la lista horario
NuevaCarga	Calcula la nueva carga del grupo
NuevoDisponibile	Calcula nuevo valor de disponibilidad para el grupo
Nombre de la clase:	CL-004 Interprete
Atributos	
Áreas	Lista de diccionario de áreas – asignaturas
Docentes	Lista de docentes pertenecientes a un área
Asignaturas	Lista de asignaturas asociadas a un área

carga	Lista de información de carga asociada a cada hora
Seleccionados	Lista de docentes que pertenecen a un área
Metodos	
Interprete	Constructor, requiere lista de áreas y docentes
TomarAsiganturas	Almacena en la lista de asignaturas aquellas asociadas a un área. Requiere parámetro área
TomarDocentes	Almacena en la lista de docentes aquellos asociadas a un área. Los docentes son almacenados en la lista de seleccionados. Requiere parámetro área
ajustarCarga	Recibe una lista de horas de carga horaria y ajusta la cantidad de horas en la lista de asignaturas. La carga horaria es la carga horaria de un grupo
retDocente	Retorna un docente de la lista de docentes seleccionados que cumplan con un mínimo de carga horaria disponible. Requiere cantidad de horas.

Tabla 17 Grupo

Nombre de la clase:	CL-005 AG
Atributos	
Generaciones	Cantidad de generaciones que debe ejecutar el algoritmo genético
Individuos	Cantidad de individuos en la población
Diversidad	Cantidad de tipos distintos de cromosomas
longitud_gen	Longitud del gen que contiene una combinación de cromosomas
p1	Punto inicial de corte
p2	Punto final de corte
Población	Conjunto de individuos con características diversas.
Metodos	
AG	Constructor, requiere generaciones, individuos, diversidad y longitud de los genes
CrearPoblacion	Crea una población con el número de individuos establecido
NuevoGen	Crea un nuevo individuo como una lista de cromosomas tomados al azar entre el número de tipos de cromosomas disponibles, este gen es una lista con tantos individuos como ha sido establecido en la variable longitud_gen

Cruzar	Recibe dos individuos y realiza cruce genético empleando cruzamiento CX de dos puntos.
Nombre de la clase:	CL-006 Soporte
Atributos	
Docentes	Lista de docentes
Grupos	Lista de grupos
info_grupos	Lista de diccionarios de información docente - carga
Metodos	
retLibres	Retorna una lista con diccionarios que contienen la información de carga disponible por docente en cada grupo
funcionFit	Calcula la función de adaptación para los genes de un individuo, representado en una lista de valores enteros, cada numero es el identificador de un docente

Tabla 18 AG

4.2.3 Diagrama de componentes.

A continuación, se describen los componentes del sistema.

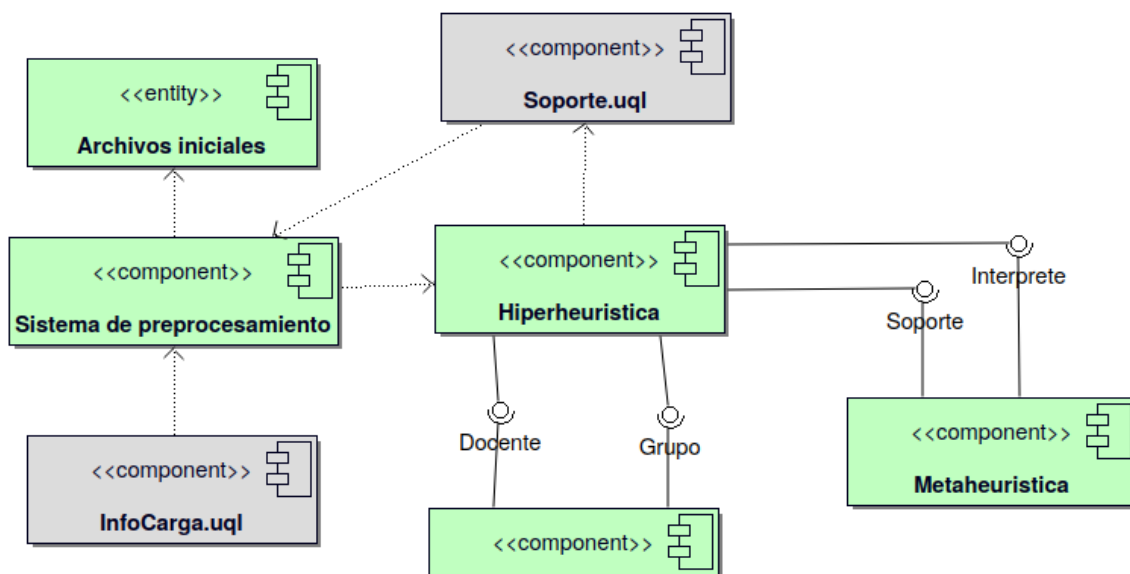


Ilustración 8 Diagrama de Componentes Software

El sistema cuenta con los siguientes componentes:

- ✓ Archivos iniciales: información provista para entrenar el modelo
- ✓ Sistema de pre procesamiento: sistema encargado de convertir información depositada en archivos a registros en bases de datos o clases y viceversa.
 - ✓ InfoCarga.uql: estructura de base de datos no relacional que almacena la información inicial del problema
 - ✓ Soporte.uql: estructura de base de datos que almacena colecciones de datos que relacionan variables como docente – asignatura o grado – carga horaria.
 - ✓ Hiperheurística: Sistema que ejecuta la heurística de bajo nivel y recopila información para configurar el espacio de búsqueda la metaheurística.
 - ✓ Heurística de bajo nivel: permite asignar a cada grupo de cada grado la carga docente mas no programa el horario.
 - ✓ Metaheurística: implementa un algoritmo genético que encuentra franjas horarias ajustadas a las restricciones, es configurada y gestionada por la hiperheurística.

4.3 Implementación

4.3.1 Elementos de la implementación.

La implementación descrita a continuación fue realizada en el lenguaje Python, desarrollada en máquina con sistema operativo Linux, y como sistema de almacenamiento de datos se empleó Unqlite, una base de datos embebida con modelo

de almacenamiento NoSql . Finalmente se ha usado el sistema de versionamiento Git para el control y gestión de versiones en los prototipos.

El uso de Python se debe principalmente a dos factores: rapidez en el desarrollo, facilidad en el uso de estructuras de tipo lista.

En cuanto a la versatilidad del lenguaje se escapa al alcance de este trabajo un análisis en cuanto a las razones de porque lo es, sin embargo, se puede argumentar que la facilidad de implementar librerías, clases y objetos, y la facilidad de uso, la pericia del equipo de trabajo en el desarrollo de aplicaciones con este lenguaje lo posicionan como la opción más adecuada para el desarrollo.

En cuanto al segundo elemento, las listas, se debe saber para interpretar mejor este capítulo que las listas en Python tienen varias versiones, listas, tuplas y diccionarios.

En la implementación se usa con frecuencia el concepto de diccionario y el concepto de lista. Una lista es un arreglo que puede contener cualquier tipo de dato, por ejemplo:

```
ls = [ 2, 4.5, 'hola', [6,8] , True]
```

La lista se comporta como arreglo y puede contener variables, constantes, otras listas u objetos.

Un diccionario es una lista de tipo clave valor, la única condición imperante es que la clave no debe repetirse, funciona como una tabla Hash, por ejemplo:

```
dcc= { 2: 'ANA' , 10: [2,3,4]}
```

Para extraer el valor se invoca la clave, es decir dcc[2] retorna 'ANA'.

Los diccionarios son particularmente importantes en la implementación puesto que al comunicarse con bases de datos NoSql tienen en común de manipular datos de forma clave valor y con frecuencia en la presentación de datos podrá observarse el uso de estos.

El sistema crea durante la ejecución dos bases de datos NoSql, InfoCarga.uql y Soporte. Ambas son embebidas, es decir que son creadas por la aplicación y su estructura fue definida en consecuencia, en tiempo de ejecución. Esto quiere decir que ante una entrada de datos distinta el contenido cambia, la estructura es inmutable pues todos los registros sin excepción son del tipo clave valor.

La primera base de datos llamada InfoCarga.uql contiene la información original cargada de la fuente de información provista por la institución, un archivo de MS Excel (de extensión xls) donde se relaciona la carga horaria de los docentes, esta es la única fuente de información que se tiene al inicio (Ilustración 9).

	B	C	D	E	F	G	H
63							
64	12	DIEGO ALVAREZ	HUMANIDADES	INGLÉS 7 A, B	6	2	
65		LIC LENGUA EXTRANJERA		INGLÉS 8 A, B	6		
66				INGLÉS 9 A, B	6		
67				INGLES 6 C	6		
68				TOTAL	24		
69							
70	13	RICARDO PIEDRAHITA	C SOCIALES	SOCIALES 7 A, B	8	0	
71		LIC SOCIALES		SOCIALES 9 A, B	8		
72				FILOSOFÍA 10 A	2		
73				FILOSOFÍA 11A, B	4		
74				TOTAL	22		
75							

Ilustración 9 Archivo de MS Excel Con Valores Iniciales

Este archivo fue pre procesado para obtener dos archivos que contiene la información que alimentará el sistema, en la tabla 19 puede observarse un fragmento de la información en los archivos de carga docente y de asignaturas.

20,GIOVANNI OSORIO,MATEMATICAS,MATEMATICAS,5,9B,,,,, 21,GIOVANNI OSORIO,MATEMATICAS,ESTADISTICA ,1,10A,10B,11A,11B,,, 22,ANDRES,EDUCACION FISICA,EDUCACION FISICA,2,6A,6B,6C,7A,7B,, 23,ANDRES,EDUCACION FISICA,EDUCACION FISICA,2,9A,9B,10A,10B,11A,11B, 24,ALBERT MENA,L. EXTRANJERAS,INGLES,5,6A,6B,,,,, 25,ALBERT MENA,L. EXTRANJERAS,INGLES,3,10A,10B,11A,11B,,, 26,DIEGO ALVAREZ,L. EXTRANJERAS,INGLES,3,7A,7B,8A,8B,9A,9B, 27,DIEGO ALVAREZ,L. EXTRANJERAS,INGLES,5,6C,,,,, 28,RICARDO PIEDRAHITA,C SOCIALES,SOCIALES,4,7A,7B,9A,9B,,
1,C. NATURALES,BIOLOGIA,QUIMICA,LAB. QUIMICA,FISICA, 2,C.POLITICAS,GESTION ,ANALISIS DE PREP,,, 3,HUMANIDADES,ESPAÑOL ,INGLES,,, 4,EDUCACION ETICA,ETICA,CATEDRA DE PAZ,,,

Tabla 19 Fragmento de datos de archivo de carga docente y fragmentos de archivo de asignaturas

Se tiene como resultado un repositorio con la información original del problema y del cual el sistema extraerá los datos necesarios para realizar la programación del horario.

La segunda base de datos llamada Soporte.uql, también embebida, almacena colecciones de datos, entendiendo una colección como un modelo de tipo clave valor agrupado bajo algún criterio, la cual podemos asimilar a una tabla en un modelo tradicional Sql.

Esta base de datos contiene colecciones que relacionan las entidades docentes, área, asignaturas, grados y grupos, es decir, las variables del problema.

El programa entrega como salida archivos de texto distinto tipo de información, entre ellas el horario y la carga final para cada grupo.

Antes de iniciar el proceso de entrenamiento y resolución el sistema realiza el pre procesamiento indicado en el algoritmo 2 tabla 20.

Algoritmo 2. Pre proceso de datos
<ol style="list-style-type: none">1. Indicar archivos csv y migrar a base de datos NoSql Soporte2. Generar archivos de restricciones automáticas y manuales3. Revisar archivos de restricciones4. Confirmar archivo de restricciones5. Generar base de datos NoSql con colecciones derivadas de las restricciones

Tabla 20 Algoritmo de Pre procesamiento

4.3.2 Heurística e hiperheurística

La hiperheurística cuenta con dos heurísticas de bajo nivel y una metaheurística. Las heurísticas de bajo nivel tienen como objetivo asignar la carga docente a los grupos y el de la metaheurística es encontrar el horario de tal forma que no se generen cruces.

Una forma de expresar la búsqueda de cada valor adecuado implica que debe ser posible recorrer cada elemento del problema, el algoritmo 3 muestra una expresión del algoritmo que realiza esta tarea.

Algoritmo 3: Asignación de horas
<p>1. Para cada día (<i>ndías</i>) de la semana:</p> <p>1.1 Para cada hora del día (<i>hd</i>):</p> <p>1.1.1 Para cada grado (<i>ng</i>):</p> <p>1.1.1.1 Para cada grupo (<i>gr</i>):</p> <p>1.1.1.1.1 Para cada asignatura (<i>nas</i>):</p> <p>1.1.1.1.1.1 Para cada docente (<i>nd</i>):</p> <p>1.1.1.1.1.1.1 asignar horas (<i>nh</i>)</p>

Tabla 21 Algoritmo de asignación horaria

La razón de esto es que en el problema original cuenta con la siguiente complejidad:

Ciclo	Operaciones	Complejidad
Para cada día (<i>ndías</i>) de la semana:	<i>ndías</i>	$O(n)$
Para cada grado (<i>ng</i>):	<i>ndías</i> * <i>ng</i>	$O(n^2)$
Para cada grupo (<i>gr</i>):	<i>ndías</i> * <i>ng</i> * <i>ngr</i>	$O(n^3)$
Para cada asignatura (<i>nas</i>):	<i>ndías</i> * <i>ng</i> * <i>ngr</i> * <i>nas</i>	$O(n^4)$
Para cada docente (<i>nd</i>):	<i>ndías</i> * <i>ng</i> * <i>ngr</i> * <i>nas</i> * <i>nd</i>	$O(n^5)$
asignar horas (<i>nh</i>)	<i>ndías</i> * <i>ng</i> * <i>ngr</i> * <i>nas</i> * <i>nd</i> * <i>n</i> horas	$O(n^6)$

Tabla 22 Análisis de complejidad

En la tabla 22 se puede apreciar que para recorrer los valores asociados a los días se trata de un ciclo con complejidad lineal, al contener una segunda sumatoria se tiene un ciclo dentro de otro ciclo la complejidad aumenta a polinómico de grado 2 (cuadrática), en la tercera sumatoria se tiene un ciclo adicional a los dos anteriores por lo que se dice que es polinómico de grado 3 (complejidad cúbica), para la siguiente sumatoria se dice que en caso del índice ser mayor a 3 es de compleja polinómica, de allí en adelante sigue siendo polinómico, es así como se puede decir que se está ante un problema NP

Se puede resumir este problema como de tipo job shop scheduling (JSS) que requiere planificar un conjunto de n trabajos $\{J_1, \dots, J_n\}$ en un conjunto de m máquinas o recursos $R = \{R_1, \dots, R_m\}$, donde cada trabajo J_i consta de una secuencia de operaciones o tareas $\{T_{i1}, \dots, T_{im}\}$ donde cada tarea T_{ij} necesita el uso de un recurso R_{Tij} durante un tiempo de procesamiento P_{Tij} , donde el objetivo es asignar a cada operación un tiempo de inicio de modo que se satisfagan tres tipos de restricciones:

- ✓ Operaciones de un mismo trabajo deben cumplir un orden o secuencia.
- ✓ La máquina tiene capacidad para procesar una única operación en un instante dado.
- ✓ No se puede interrumpir una operación una vez haya comenzado.

El problema JSS está categorizado como un problema de optimización con restricciones y se encuentra dentro de la categoría de problemas NP-hard.

Es posible construir una analogía entre el problema JSS y el problema de asignación docente:

- ✓ J conjunto de trabajos: clase en grupo
- ✓ R recurso: docente
- ✓ T tareas que requieren recurso: asignatura
- ✓ Tiempo de utilización P: hora de clase

Además cumple con las tres condiciones:

- ✓ Se requiere secuencia ya que las horas de clase están programadas diariamente en orden estricto (primera hora, segunda hora, etc.).
- ✓ Una asignatura solo puede dictarse en una hora de clase. No es posible dictar dos asignaturas en una hora de clase.
- ✓ Una hora de clase es atómica, se entiende una hora de clase como el trabajo de dictar una asignatura en dicho espacio de tiempo, durante la hora de clase se dicta una y solo una asignatura.

Por consiguiente, la solución planteada consiste en dividir el problema en dos partes:

- ✓ Asignar carga horaria de los docentes en relación a los grupos.
- ✓ Construir el horario dado que la asignación docente es coherente con la cantidad de horas esperadas para cada grado – grupo del modelo.

De esta forma la hiperheurística delegará la responsabilidad de asignar la carga horaria a las heurísticas de bajo nivel y la organización del horario al algoritmo genético.

4.3.3 Heurística de bajo nivel

Se implementa la heurística de bajo nivel basada en asignar áreas primero, esta heurística es una de tantas posibles tal como asignar grupos primero o asignar docentes primero.

Esta heurística fue implementada con el fin de distribuir la carga horaria de los docentes, el resultado final fue una lista de grupos (grado – grupo) que contiene una lista de asignaturas con carga específica donde a cada asignatura corresponde un docente.

En este modelo, para cada curso se deben tomar las asignaturas que deben programarse, luego se debe buscar docente para asociar la asignatura. Se respetan las restricciones descritas en el numeral 4.1.

Esta heurística posee las siguientes condiciones previas:

- ✓ Se tiene conjunto de docentes con información de áreas de desempeño y carga esperada.
- ✓ Se tiene conjunto de asignaturas asociadas a un grado. Cada asignatura posee cantidad de horas requeridas, ejemplo ingles cantidad 2 horas.
- ✓ Se tiene conjunto de áreas y asignaturas relacionadas con estas áreas, ejemplo matemáticas tiene asociado matemáticas, estadística y geometría.
- ✓ Se tiene conjunto de grados y grupos. Se simplifica como conjunto de grupos, ejemplo 6a, 7b, etc.

- ✓ El algoritmo 4 muestra la manera en que la heurística asigna la carga horaria a los docentes.

Algoritmo 4: Heurística basada en áreas
<ol style="list-style-type: none"> 1. Crear Hora 2. Para cada área en el conjunto de áreas tomar un área <ol style="list-style-type: none"> 3. Tomar el conjunto de asignaturas asociadas al área seleccionada <ol style="list-style-type: none"> 3.1 Tomar asignatura del conjunto de asignaturas y asociar a Hora asignatura 3.2 Para cada elemento del conjunto de grupos (grado, grupo) buscar asignatura seleccionada <ol style="list-style-type: none"> 3.2.1 Si asignatura encontrada asociar grado y grupo a la Hora 3.2.2 Ir a paso 3.3 3.2.3 Para cada docente en el conjunto de docentes buscar Docente que pertenezca al área. 3.2.4 Si el docente posee carga esperada mayor a cero (0) <ol style="list-style-type: none"> 3.2.4.1 Asociar docente a Hora 3.2.4.2 Disminuir carga esperada en Docente en cantidad de horas asociadas a la asignatura. 3.2.4.3 Ir a paso 3.5 3.2.5 Adicionar Hora a lista de horas programadas en grupo 3.3 Si hay mas asignaturas ir a paso 3.1 (tomar próxima asignatura). Si no hay mas asignaturas ir a 4 4. Si hay mas áreas tomar próxima área e ir a 3. Si no hay mas asignaturas terminar.

Tabla 23 Heurística basada en áreas

- ✓ La heurística implementada tiene una política algorítmica perezosa, que en oposición a la política algorítmica ambiciosa o golosa, no trata de explorar todos los espacios posibles sino que al encontrar un valor posible y viable se detiene.
- ✓ Es posible que una vez finalizado el proceso algunas asignaturas no hayan sido asociadas a docentes, la razón de ello es que los docentes no

tienen espacio suficiente en la carga para tomarla. Debido a esto el sistema tiene un paso adicional de ajuste expresado en el algoritmo 5 tabla 24.

Algoritmo 5: Ajuste de asignaturas.
<ol style="list-style-type: none"> 1. Para cada grupo en conjunto de grupos tomar grupo <ol style="list-style-type: none"> 1.1 Para cada hora asignada al grupo <ol style="list-style-type: none"> 1.1.1 Si horas en lista de horas esperadas terminar 1.1.2 Si hora no está en lista adicionar a lista de faltantes 2. Para cada hora en lista de faltantes <ol style="list-style-type: none"> 2.1 Solicitar a usuario si asignar manual (ir a 2.3) o automático (2.2). 2.2 Para cada docente en grupo de docentes <ol style="list-style-type: none"> 2.2.1 Si carga en docente > 0 y menor a carga de la hora entonces asignar docente a Hora y terminar 2.2.2 Si no ir a 2.2 2.3 Para cada docente en grupo docentes <ol style="list-style-type: none"> 2.3.1 Si docente pertenece a área de la Hora <ol style="list-style-type: none"> 2.3.1.1 Mostrar docente y carga horaria disponible 2.3.2 Solicitar a usuario docente de la lista 2.3.3 Asignar a Hora docente seleccionado 2.4 Terminar

Tabla 24 Ajuste de horario

La asignación presenta el siguiente modelo:

- ✓ Para cada área en el conjunto de áreas tomar un área.
- ✓ Tomar el conjunto de asignaturas asociadas al área seleccionada.
- ✓ Tomar asignatura del conjunto de asignaturas.
- ✓ Para cada elemento del conjunto de grupos (grado, grupo) buscar asignatura seleccionada, si encontrada asociar grado y grupo a la asignatura.
- ✓ Para cada docente en el conjunto de docentes buscar docente que pertenezca al área.
- ✓ Si el docente posee carga esperada mayor a cero (0), asociar docente a asignatura y disminuir carga esperada en cantidad de horas asociadas a la asignatura.

- ✓ Tomar próxima asignatura e ir a iii. Si no hay más asignaturas terminar e ir a vii.
- ✓ Tomar próxima área e ir a ii. Si no hay más asignaturas terminar.

Este modelo considera la intervención humana como posible, pues en el mundo real ocurre y se evidencia en el resultado que existen falencias en la asignación (Tabla 24). En la práctica los docentes pueden tener carga incompleta o sobre carga (horas extra), en el caso de los docentes con carga por debajo del valor esperado de carga les compensan con labores administrativas o tareas que cuentan como carga horaria, tales tareas son como supervisión de servicio social, seguimiento a pasantías, descarga por dirección de grupo, etc.

ALBERT M ; 22	{ALBERT M : 0}
GIOVANNI O ; 26	{GIOVANNI O : 0}
SONIA ; 22	{SONIA : 0}
DIANA MILENA R ; 6	{DIANA MILENA R : 0}
PAOLA ; 22	{PAOLA : 0}
RICARDO P ; 22	{RICARDO P : 0}
AMANDA D ; 22	{AMANDA D : 0}
ANDRES ; 22	{ANDRES : 0}
STELLA T ; 24	{STELLA T : 0}
MANUEL V ; 26	{MANUEL V : 0}
JOAQUIN ; 24	{JOAQUIN : 0}
SERGIO M ; 22	{SERGIO M : 0}
DIEGO A ; 26	{DIEGO A : 0}
MIGUEL ANGEL G ; 22	{MIGUEL ANGEL G : 0}
ANA CRISTINA B ; 22	{ANA CRISTINA B : 0}
ROGER ; 22	{ROGER : 0}
MIRYAM T ; 10	{MIRYAM T : -1}
LISETH ; 22	{LISETH : -7}
CARLOS T ; 22	{CARLOS T : 0}
JOHN ; 22	{JOHN : 5}
ALVARO ; 20	{ALVARO : 0}

Tabla 25 Salida del sistema, carga docente

Izquierda carga esperada, derecha carga asignada por heurística de bajo nivel. Los valores negativos indican sobrecarga.

En la tabla 26 se puede observar la carga horaria esperada para cada grado, en el ejercicio de prueba se cuenta con 3 grados 6 y 2 de cada uno de los restantes grados (13 grupos).

Grado	Carga esperada
6	[['SOCIALES', 'ETICA', 'DIBUJO', 'TECNOLOGIA', 'INGLES', 'EDUCACION FISICA', 'MATEMATICAS', 'ESPAÑOL', 'BIOLOGIA', 'ARTES', 'RELIGION'], [6, 1, 1, 2, 5, 2, 7, 7, 5, 2, 1]]
7	[['BIOLOGIA', 'DIBUJO', 'ETICA', 'TECNOLOGIA', 'SOCIALES', 'INGLES', 'EDUCACION FISICA', 'MATEMATICAS', 'ESPAÑOL', 'ARTES', 'RELIGION'], [4, 1, 1, 2, 4, 3, 2, 5, 5, 2, 1]]
8	[['SOCIALES', 'ETICA', 'BIOLOGIA', 'DIBUJO', 'TECNOLOGIA', 'INGLES', 'MATEMATICAS', 'ESPAÑOL', 'ARTES', 'RELIGION', 'EDUCACION FISICA'], [4, 1, 4, 1, 2, 3, 5, 5, 2, 1, 2]]
9	[['TECNOLOGIA', 'EDUCACION FISICA', 'ESPAÑOL', 'BIOLOGIA', 'DIBUJO', 'RELIGION', 'ARTES', 'SOCIALES', 'INGLES', 'MATEMATICAS', 'ETICA'], [2, 2, 5, 4, 1, 1, 2, 4, 3, 5, 1]]
10	[['TECNICAS DE IMP', 'DISEÑO', 'TECNOLOGIA', 'FILOSOFIA', 'INGLES', 'EDUCACION FISICA', 'ESTADISTICA', 'CONTABILIDAD', 'LAB. QUIMICA', 'CATEDRA DE PAZ', 'RELIGION', 'ETICA', 'ARTES', 'MATEMATICAS', 'FISICA', 'QUIMICA', 'SEG. INDUSTRIAL', 'ESPAÑOL'], [2, 4, 2, 2, 3, 2, 1, 1, 1, 1, 1, 2, 4, 3, 4, 1, 4]]
11	[['TECNICAS DE IMP', 'ARTES', 'TECNOLOGIA', 'FILOSOFIA', 'INGLES', 'EDUCACION FISICA', 'ESTADISTICA', 'ANALISIS DE PREP', 'ESPAÑOL', 'LAB. QUIMICA', 'GESTION', 'CATEDRA DE PAZ', 'RELIGION', 'DISEÑO', 'MATEMATICAS', 'FISICA', 'QUIMICA', 'ETICA'], [2, 2, 2, 2, 3, 2, 1, 1, 4, 1, 1, 1, 1, 3, 4, 3, 4, 1]]

Tabla 26 Información de carga horaria esperada por grado

Además cada docente tiene un límite de horas establecido, este criterio obedece a varias razones, si bien la carga docente esperada es de 22, se tienen dos docentes que provienen de otra sede y que completan su carga en la que es objeto de estudio. También está el caso de docentes que han solicitado horas extra y de allí el incremento sobre el valor estipulado por la ley (ver tabla 2, asignación de carga esperada y entregada).

En la tabla 27 puede observarse la carga horaria asignada a cada grado. Esta carga aparece como resultado de asignar la carga horaria en dos instancias, la primera asigna docente preservando el criterio de relación área - asignatura – docente, la segunda instancia toma horas que no han sido asignadas y permite que un ser humano

(el coordinado o un experto) decida si la asignación es automática o si desea establecerla el mismo.

Grupo	Carga
6	[['DIBUJO - 1', 'ARTES - 2', 'MATEMATICAS - 7', 'EDUCACION FISICA - 2', 'ESPANOL - 7', 'TECNOLOGIA - 2', 'ETICA - 1', 'RELIGION - 1', 'INGLES - 5', 'BIOLOGIA - 5', 'SOCIALES - 6'], 39]
7	[['DIBUJO - 1', 'ARTES - 2', 'MATEMATICAS - 5', 'EDUCACION FISICA - 2', 'ESPANOL - 5', 'TECNOLOGIA - 2', 'ETICA - 1', 'RELIGION - 1', 'INGLES - 3', 'BIOLOGIA - 4', 'SOCIALES - 4'], 30]
8	[['DIBUJO - 1', 'ARTES - 2', 'MATEMATICAS - 5', 'EDUCACION FISICA - 2', 'ESPANOL - 5', 'TECNOLOGIA - 2', 'ETICA - 1', 'INGLES - 3', 'BIOLOGIA - 4', 'SOCIALES - 4', 'RELIGION - 1'], 30]
9	[['DIBUJO - 1', 'ARTES - 2', 'MATEMATICAS - 5', 'EDUCACION FISICA - 2', 'ESPANOL - 5', 'TECNOLOGIA - 2', 'ETICA - 1', 'INGLES - 3', 'BIOLOGIA - 4', 'SOCIALES - 4', 'RELIGION - 1'], 30]
10	[['ARTES - 2', 'MATEMATICAS - 4', 'CONTABILIDAD - 1', 'ESTADISTICA - 1', 'EDUCACION FISICA - 2', 'ESPANOL - 4', 'TECNOLOGIA - 2', 'FISICA - 3', 'ETICA - 1', 'CATEDRA DE PAZ - 1', 'INGLES - 3', 'TECNICAS DE IMP - 2', 'DISENO - 4', 'SEG. INDUSTRIAL - 1', 'LAB. QUIMICA - 1', 'QUIMICA - 4', 'FILOSOFIA - 2', 'RELIGION - 1'], 39]
11	[['ARTES - 2', 'MATEMATICAS - 4', 'ESTADISTICA - 1', 'EDUCACION FISICA - 2', 'ESPANOL - 4', 'ANALISIS DE PREP - 1', 'TECNOLOGIA - 2', 'FISICA - 3', 'ETICA - 1', 'CATEDRA DE PAZ - 1', 'INGLES - 3', 'TECNICAS DE IMP - 2', 'DISENO - 3', 'LAB. QUIMICA - 1', 'FILOSOFIA - 2', 'GESTION - 1', 'RELIGION - 1', 'QUIMICA - 4'], 38]

Tabla 27 Programación de carga horaria por grado arrojada por el sistema

Esto se debe a que en la práctica para completar la carga a algunos docentes les programan asignaturas que no son de su área, tales como ética o religión, sin embargo, pueden quedar horas sin asignación que no pueden ser programadas de esta forma, tales como estadística o laboratorio de química, cuyo nivel de especificidad requiere un docente con capacitación propia del área.

Por este motivo una vez concluye el proceso general de asignación la hiperheurística flexibiliza el proceso de estas preguntando si se desea asignar manualmente o si la hiperheurística completa el proceso. En el último caso, el criterio empleado es buscar por área y asignar al de mayor carga horaria esperada sin asignar, o en caso contrario, si no hay docente de área disponible solo tiene en cuenta el criterio

de carga esperada sin asignar. En la tabla 29 se muestran los valores asociados a la asignación de carga de la heurística de bajo nivel.

Docente	Carga
ALBERT M	[INGLES 5, INGLES 5, INGLES 5, INGLES 3, INGLES 3, RELIGION 1]
GIOVANNI O	[MATEMATICAS 7, MATEMATICAS 7, MATEMATICAS 7, MATEMATICAS 5]
SONIA	[TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2, TECNOLOGIA 2]
DIANA MILENA R	[ESPANOL 5, ETICA 1]
PAOLA	[ESPANOL 7, ESPANOL 7, ESPANOL 7, RELIGION 1]
RICARDO P	[SOCIALES 6, SOCIALES 6, SOCIALES 6, SOCIALES 4]
AMANDA D	[ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, ETICA 1, CATEDRA DE PAZ 1, ETICA 1, CATEDRA DE PAZ 1, ETICA 1, CATEDRA DE PAZ 1, ETICA 1, CATEDRA DE PAZ 1, RELIGION 1, RELIGION 1, RELIGION 1, RELIGION 1, RELIGION 1, RELIGION 1]
ANDRES	[EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2, EDUCACION FISICA 2]
STELLA T	[ANALISIS DE PREP 1, ANALISIS DE PREP 1, BIOLOGIA 5, BIOLOGIA 5, BIOLOGIA 5, BIOLOGIA 4, LAB. QUIMICA 1, LAB. QUIMICA 1, LAB. QUIMICA 1]
MANUEL V	[SOCIALES 4, SOCIALES 4, SOCIALES 4, SOCIALES 4, SOCIALES 4, FILOSOFIA 2, FILOSOFIA 2, FILOSOFIA 2]
JOAQUIN	[MATEMATICAS 5, MATEMATICAS 5, MATEMATICAS 5, MATEMATICAS 5, MATEMATICAS 4]
SERGIO M	[MATEMATICAS 5, CONTABILIDAD 1, ESTADISTICA 1, MATEMATICAS 4, CONTABILIDAD 1, ESTADISTICA 1, MATEMATICAS 4, ESTADISTICA 1, MATEMATICAS 4]
DIEGO A	[INGLES 3, INGLES 3, INGLES 3, INGLES 3, INGLES 3, INGLES 3, INGLES 3, INGLES 3, RELIGION 1, RELIGION 1]
MIGUEL ANGEL G	[DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1, ARTES 2, DIBUJO 1]
ANA CRISTINA B	[ARTES 2, DIBUJO 1, ARTES 2, ARTES 2, ARTES 2, ARTES 2, ARTES 2, ARTES 2, TECNICAS DE IMP 2, DISENO 4, SEG. INDUSTRIAL 1, TECNICAS DE IMP 2]
ROGER	[ESPANOL 5, ESPANOL 5, ESPANOL 5, ESPANOL 5, RELIGION 1, GESTION 1]
MIRYAM T	[EDUCACION FISICA 2, EDUCACION FISICA 2, TECNOLOGIA 2, TECNOLOGIA 2, RELIGION 1, FILOSOFIA 2]
LISETH	[BIOLOGIA 4, BIOLOGIA 4, BIOLOGIA 4, BIOLOGIA 4, BIOLOGIA 4, LAB. QUIMICA 1, QUIMICA 4, QUIMICA 4]
CARLOS T	[ESPANOL 5, ESPANOL 4, ESPANOL 4, ESPANOL 4, ESPANOL 4, ESTADISTICA 1]
JOHN	[DISENO 4, SEG. INDUSTRIAL 1, TECNICAS DE IMP 2, DISENO 3, TECNICAS DE IMP 2, DISENO 3, GESTION 1, RELIGION 1]
ALVARO	[FISICA 3, FISICA 3, FISICA 3, FISICA 3, QUIMICA 4, QUIMICA 4]

Tabla 28 Carga docente arrojada por la heurística.

Carga total esperada	451
Asignadas	434
Faltantes	17
TAsignado forma automática	96.23%
Asignado forma semiautomática	3.77%

Tabla 29 Datos de asignación de heurística de bajo nivel

Es decir, que una vez realizado este proceso el modelo abordó parte del algoritmo necesario para resolver el problema, puesto que cumplió con el propósito de asociar a cada grado y a cada grupo una asignatura con un docente.

El punto de partida para la metaheurística será el siguiente:

Asignar para cada grupo de cada grado de cada día de cada semana un docente, ya que la heurística de bajo nivel ha generado para cada grupo una lista de horas con docente y asignatura asociada pero que carece de hora y día definido. En la tabla 30 puede observarse el código y la relación clave valor de la colección docente limite almacenada en la base de datos Soporte.uql y en la tabla 31 puede observarse que para cada grupo existe una programación horaria ligada a un docente y en la tabla.

```
0 {'ant': 'ALBERT M', 'con': 22}
1 {'ant': 'GIOVANNI O', 'con': 26}
2 {'ant': 'SONIA ', 'con': 22}
3 {'ant': 'DIANA MILENA R ', 'con': 6}
4 {'ant': 'PAOLA ', 'con': 22}
5 {'ant': 'RICARDO P', 'con': 22}
6 {'ant': 'AMANDA D', 'con': 22}
7 {'ant': 'ANDRES ', 'con': 22}
8 {'ant': 'STELLA T ', 'con': 24}
9 {'ant': 'MANUEL V ', 'con': 26}
10 {'ant': 'JOAQUIN ', 'con': 24}
11 {'ant': 'SERGIO M ', 'con': 22}
12 {'ant': 'DIEGO A ', 'con': 26}
```

```

13 {ant:'MIGUEL ANGEL G ', con': 22}
14 {ant:'ANA CRISTINA B', con': 22}
15 {ant:'ROGER ', con': 22}
16 {ant:'MIRYAM T', con': 10}
17 {ant:'LISETH ', con': 22}
18 {ant:'CARLOS T ', con': 22}
19 {ant:'JOHN ', con': 22}
20 {ant:'ALVARO ', con': 20}

```

Tabla 30 Coleccion en bse de datos Soporte.uql con Descripcion de id, nombre y carga esperada

```

6A:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 7 - 1', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 7 - 4', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 3', 'RELIGION - 1 - 6', 'INGLES - 5 - 0', 'BIOLOGIA - 5 - 8', 'SOCIALES - 6 - 5'], 39]
6B:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 7 - 1', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 7 - 4', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'RELIGION - 1 - 6', 'INGLES - 5 - 0', 'BIOLOGIA - 5 - 8', 'SOCIALES - 6 - 5'], 39]
6C:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 7 - 1', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 7 - 4', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'RELIGION - 1 - 6', 'INGLES - 5 - 0', 'BIOLOGIA - 5 - 8', 'SOCIALES - 6 - 5'], 39]
11A:[['ARTES - 2 - 14', 'MATEMATICAS - 4 - 11', 'ESTADISTICA - 1 - 11', 'EDUCACION FISICA - 2 - 16', 'ESPAÑOL - 4 - 18', 'ANALISIS DE PREP - 1 - 8', 'TECNOLOGIA - 2 - 16', 'FISICA - 3 - 20', 'ETICA - 1 - 6', 'CATEDRA DE PAZ - 1 - 6', 'INGLES - 3 - 12', 'TECNICAS DE IMP - 2 - 19', 'DISEÑO - 3 - 19', 'LAB. QUIMICA - 1 - 8', 'FILOSOFIA - 2 - 9', 'GESTION - 1 - 15', 'RELIGION - 1 - 16', 'QUIMICA - 4 - 17'], 38]
7B:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 5 - 10', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 15', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'RELIGION - 1 - 6', 'INGLES - 3 - 0', 'BIOLOGIA - 4 - 17', 'SOCIALES - 4 - 9'], 30]
7A:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 5 - 1', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 3', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'RELIGION - 1 - 6', 'INGLES - 3 - 0', 'BIOLOGIA - 4 - 8', 'SOCIALES - 4 - 5'], 30]
11B:[['ARTES - 2 - 14', 'MATEMATICAS - 4 - 11', 'EDUCACION FISICA - 2 - 16', 'ESPAÑOL - 4 - 18', 'ANALISIS DE PREP - 1 - 8', 'TECNOLOGIA - 2 - 16', 'FISICA - 3 - 20', 'ETICA - 1 - 6', 'CATEDRA DE PAZ - 1 - 6', 'INGLES - 3 - 12', 'TECNICAS DE IMP - 2 - 19', 'DISEÑO - 3 - 19', 'LAB. QUIMICA - 1 - 17', 'FILOSOFIA - 2 - 16', 'ESTADISTICA - 1 - 10', 'GESTION - 1 - 18', 'RELIGION - 1 - 19', 'QUIMICA - 4 - 17'], 38]
8B:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 5 - 10', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 15', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'INGLES - 3 - 12', 'BIOLOGIA - 4 - 17', 'SOCIALES - 4 - 9', 'RELIGION - 1 - 0'], 30]
8A:[['DIBUJO - 1 - 13', 'ARTES - 2 - 13', 'MATEMATICAS - 5 - 10', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 15', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'RELIGION - 1 - 6', 'INGLES - 3 - 12', 'BIOLOGIA - 4 - 17', 'SOCIALES - 4 - 9'], 30]
9A:[['DIBUJO - 1 - 13', 'ARTES - 2 - 14', 'MATEMATICAS - 5 - 10', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 15', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'INGLES - 3 - 12', 'BIOLOGIA - 4 - 17', 'SOCIALES - 4 - 9', 'RELIGION - 1 - 4'], 30]
10A:[['ARTES - 2 - 14', 'MATEMATICAS - 4 - 10', 'CONTABILIDAD - 1 - 11', 'ESTADISTICA - 1 - 11', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 4 - 18', 'TECNOLOGIA - 2 - 2', 'FISICA - 3 - 20', 'ETICA - 1 - 6', 'CATEDRA DE PAZ - 1 - 6', 'INGLES - 3 - 12', 'TECNICAS DE IMP - 2 - 14', 'DISEÑO - 4 - 14', 'SEG. INDUSTRIAL - 1 - 14', 'LAB. QUIMICA - 1 - 8', 'QUIMICA - 4 - 20', 'FILOSOFIA - 2 - 9', 'RELIGION - 1 - 12'], 39]
9B:[['DIBUJO - 1 - 14', 'ARTES - 2 - 14', 'MATEMATICAS - 5 - 11', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 5 - 18', 'TECNOLOGIA - 2 - 2', 'ETICA - 1 - 6', 'INGLES - 3 - 12', 'BIOLOGIA - 4 - 17', 'SOCIALES - 4 - 9', 'RELIGION - 1 - 12'], 30]
10B:[['ARTES - 2 - 14', 'MATEMATICAS - 4 - 11', 'CONTABILIDAD - 1 - 11', 'ESTADISTICA - 1 - 11', 'EDUCACION FISICA - 2 - 7', 'ESPAÑOL - 4 - 18', 'TECNOLOGIA - 2 - 2', 'FISICA - 3 - 20', 'ETICA - 1 - 6', 'CATEDRA DE PAZ - 1 - 6', 'INGLES - 3 - 12', 'TECNICAS DE IMP - 2 - 14', 'DISEÑO - 4 - 19', 'SEG. INDUSTRIAL - 1 - 19', 'LAB. QUIMICA - 1 - 8', 'QUIMICA - 4 - 20', 'FILOSOFIA - 2 - 9', 'RELIGION - 1 - 15'], 39]

```

Tabla 31 Descripción de carga horaria arrojada por el sistema para cada grupo, registro de la forma grupo, lista de la forma asignatura-carga horaria -id docente asignado

4.3.4 Hiperheurística

La metaheurística empleada es un algoritmo genético para el cual la hiperheurística establece valores de número de individuos en la población, cantidad de generaciones, posición inicial y final de cruzamiento y la longitud del genoma.

4.3.4.1 Definición del genoma y del algoritmo

En esta instancia del proceso es muy importante construir un modelo de genoma que permita representar las necesidades del problema, se ha modelado el gen como la cantidad de grupos disponibles en la institución, vale la pena aclarar que este valor es flexible y puede adaptarse a otras instituciones. La tabla 9 presenta un esquema del genoma y la representación a la que hace referencia.

0	1	2	3	4	5	6	7	8	9	10	11	12
6 ^a	6B	6C	7A	7B	8A	8B	9A	9B	10A	10B	11A	11B
id0	id1	id2	...									id12

Tabla 32 Genoma del algoritmo genético adaptado al problema

En el genoma se asignará el número de identificación interno de cada docente en cada una de las posiciones. Para el ejercicio se cuenta con 21 docentes, con identificación de 0 a 20, por lo que un genoma valido será aquel de 13 posiciones y un valor entero entre 0 y 20 en cada una de ellas.

Puesto que para cada docente y para cada grupo ya se ha asignado la carga horaria, será trabajo del algoritmo genético encontrar una combinación adecuada de asignación horaria de tal forma que no existan cruces entre docentes.

La hiperheurística se comporta acorde a la tabla 33.

Algoritmo 6: Hiperheurística para asignación de horarios
1. Para cada día de la semana tomar día
1.1 Para cada hora del día tomar hora
1.1.1 Hiperheurística establece parámetros generales de algoritmo genético

(numero de individuos, cantidad de generaciones, longitud del genoma)

1.1.2 Hiperheurística define espacio de búsqueda del algoritmo genético basado en carga docente por grupo

1.1.3 Hiperheurística inicia algoritmo genético

1.1.4 Hiperheurística toma del algoritmo genético la mejor solución encontrada para una franja horaria de un día en particular

1.1.5 Hiperheurística registra solo los campos validos encontrados en la mejor solución

1.1.6 Hiperheurística modifica la información de carga docente por grupo.

1.1.7 Si existe próxima hora ir a 1.1.1 en caso contrario ir a 1.2

1.2 Si existe próximo día ir a 1 en caso contrario terminar

Tabla 33 Algoritmo Hiperheurístico para la asignación de horarios

Por definición una hiperheurística es un heurística que actúa sobre el espacio de búsqueda de la metaheurística, es decir que cada vez que se ejecuta el algoritmo genético su modelo de población es modificado, redefiniendo el espacio de búsqueda.

En la implementación la hiperheurística establece una la variable numérica para representar el primer día de la semana, de esta manera la variable día tendrá un valor de 1 para el lunes, 2 para el martes, etc. Luego definirá una variable para las horas del día, en este trabajo se buscará la asignación horaria para las 6 horas reglamentarias en una institución educativa publica, teniendo en cuenta que este horario podría ser extendido si se tiene jornada única o alguna variación atípica.

Luego la hiperheurística establecerá valores predeterminados para el algoritmo genético, en este caso se definen valores siguientes:

- ✓ Tamaño de población: 100 individuos.
- ✓ Máximo número de generaciones: 20
- ✓ Diversidad: 21
- ✓ Grupos por día: 13

La diversidad es la cantidad de docentes que se tiene disponible, al definir este valor la clase que gestiona el algoritmo genético utiliza este número para crear los genes con valores entre 0 y el número de diversidad, de esta manera un gen es una lista de números donde cada número corresponde al código que identifica un docente.

La variable de grupos por día le indica al algoritmo la cantidad de espacios con que debe disponer un gen para representar todos los grupos de la institución, se tienen trece (13) grupos, tres de grado 6, y 2 de cada grado entre grado séptimo y once.

Con la anterior implementación se cumple que la hiperheurística debe permitir adaptar la solución a otros espacios similares para el mismo problema, ya que si cambian las condiciones al cambiar de institución, es posible adaptar los parámetros del algoritmo genético sin tener que intervenir el código.

Por último, la manera como la hiperheurística resuelve el problema es ejecutando el algoritmo genético para encontrar una solución factible para la primera hora del primer día, la calidad de esta solución está determinada por la función de adaptación que se encuentra contenida en la clase soporte.

La clase soporte recibe la información asociada a los grupos y genera un diccionario, que no es más que una lista que funciona con estructura clave valor, de tal manera que la clave es el identificador del registro y el valor es el dato o conjunto de datos que se le asocian.

En la implementación la identificación del docente es la clave, misma que coincide con los dígitos que componen el genoma, y el valor es la cantidad de horas que el docente tiene asignadas. En la tabla 10 se puede observar un ejemplo del diccionario arrojados por la clase soporte el cual resume la carga docente por grupo.

La hiperheurística interpreta estos registros de la siguiente manera:

- ✓ Todos los diccionarios están en una lista
- ✓ La lista, que en Python se comporta como un arreglo, a cada diccionario corresponde una posición, almacenada en el diccionario de información de grupos (Ver tabla 34, parte inferior).
- ✓ En el diccionario de información docente se muestra la relación docente - carga horaria.

{0: 3, 1: 1, 2: 0, 3: 0, 4: 3, 5: 1, 6: 0, 7: 1, 8: 3, 13: 2}
{0: 0, 1: 3, 2: 2, 4: 2, 5: 1, 6: 1, 7: 0, 8: 0, 13: 2}
{0: 0, 1: 6, 2: 0, 4: 0, 5: 3, 6: 1, 7: 0, 8: 2, 13: 2}
{0: 0, 1: 1, 2: 0, 3: 0, 5: 0, 6: 1, 7: 0, 8: 0, 13: 2}
{0: 1, 2: 0, 6: 1, 7: 0, 9: 0, 10: 3, 13: 2, 15: 2, 17: 0}
{2: 0, 6: 1, 7: 0, 9: 0, 10: 3, 12: 0, 13: 2, 15: 3, 17: 0}
{0: 0, 2: 0, 6: 0, 7: 0, 9: 2, 10: 2, 12: 0, 13: 2, 15: 0, 17: 0}
{2: 0, 4: 0, 6: 0, 7: 0, 9: 1, 10: 1, 12: 0, 13: 0, 14: 0, 15: 0, 17: 1}
{2: 0, 6: 0, 7: 0, 9: 3, 11: 0, 12: 1, 14: 2, 17: 1, 18: 0}
{2: 1, 6: 1, 7: 0, 8: 0, 9: 0, 10: 1, 11: 1, 12: 1, 14: 1, 18: 0, 20: 4}
{2: 0, 6: 1, 7: 2, 8: 0, 9: 0, 11: 1, 12: 3, 14: 2, 15: 0, 18: 1, 19: 1, 20: 4}
{6: 1, 8: 1, 9: 0, 11: 1, 12: 0, 14: 2, 15: 0, 16: 4, 18: 3, 19: 3, 20: 0}
{6: 1, 8: 0, 11: 0, 12: 2, 14: 0, 16: 2, 17: 2, 18: 1, 19: 4, 20: 0}
{0: ['6', 'A'], 1: ['6', 'B'], 2: ['6', 'C'], 3: ['7', 'A'], 4: ['7', 'B'], 5: ['8', 'A'], 6: ['8', 'B'], 7: ['9', 'A'], 8: ['9', 'B'], 9: ['10', 'A'], 10: ['10', 'B'], 11: ['11', 'A'], 12: ['11', 'B']}

Tabla 34 Valores arrojados por la clase soporte. En el parte superior listado de carga docente por grupo. En la parte inferior diccionario de informacion de grupos

De esta forma en {0: 3, 1: 1, 2: 0, 3: 0, 4: 3, 5: 1, 6: 0, 7: 1, 8: 3, 13: 2} para 0: 3 (id de docente 0) se tienen 3 horas que dictar en la semana, para el registro 13: 2 el docente (con id 13) tiene programado 2 horas a la semana. En la tabla 7 se encuentra la relación id- docente.

En el diccionario de información de grupos se observa que la posición 0 de la lista, de donde se ha sacado el anterior diccionario, en el diccionario de información de grupos a la posición 0 (registro 0: ['6', 'A']) corresponde 6A.

Una vez el algoritmo genético ha encontrado un valor óptimo, entendiendo este como aquel que representa la asignación de la primera hora del primer día sin docentes que se repitan y que cada docente este en una columna que representa un grupo al que previamente se la ha asignado una carga docente, se puede decir que se ha hallado la programación para todos los cursos en una franja horaria en particular.

El paso siguiente es almacenar la información de la franja horaria encontrada en una lista de programación de horario final que para cada hora se ha asociado grado grupo, asignatura, docente, día y hora, teniendo una entidad atómica: la hora de clase.

Finalmente, para cada grupo se realiza una actualización de horas disponibles por cada asignatura, con lo que el ciclo termina y se incrementa la hora de búsqueda.

Es así como iniciará un nuevo ciclo para el mismo día, pero para la segunda hora, la hiperheurística generará un nuevo diccionario con los datos actualizados de carga docente y lanzará de nuevo el algoritmo genético para encontrar la próxima franja horaria. Esto se repetirá hasta alcanzar el límite de horas diarias de búsqueda y cuando esto suceda, incrementará la variable de día, yendo de lunes a martes y así hasta completar los días que se espere realizar una programación.

Por lo anterior se puede afirmar que se ha implementado una hiperheurística de tipo constructiva, ya que no interfiere en la metaheurística en sus mecanismos o composición, sino que toma los resultados que esta va arrojando y cambia el espacio

de búsqueda acotándolo, lo que en la práctica significa que reduce el tiempo necesario para encontrar una solución factible.

En cuanto a la función de adaptación esta ópera de la siguiente manera:

- ✓ No se encuentra en la clase del algoritmo genético
- ✓ La clase algoritmo genético permite crear individuos, agruparlos en una población y , realizar el proceso de cruce.
- ✓ Castiga con $-n$ puntos (donde n es el número de espacios en el gen) en caso de que un docente no tenga carga asociada al grupo o el gen tenga docentes repetidos.
- ✓ Premia con $+1$ si el docente se encuentra en una posición que representa un grupo donde tiene carga horaria asignada y además tiene horas pendientes por programar.

Cuando el proceso completo de programación ha terminado se generan archivos de texto donde se registra el horario asignado por grupo que contiene día, hora, asignatura y docente.

También se generan archivos de texto de asignación horaria por días, es decir que se tiene un archivo por cada día con información de la carga horaria. En el anexo 1 se encuentra el registro de cada archivo generado por el sistema.

4.4 Análisis de Datos

Los datos mostrados a continuación fueron encontrados con los siguientes parámetros:

Configuración algoritmo genético	
Individuos	100
Diversidad	21 (cantidad de docentes)
Generaciones	20
Longitud gen	13 (cantidad de grupos)

Tabla 35 Configuración Algoritmo Genético

Configuración de hiperheurística	
Limite de días	5 (lunes a viernes)
Limite de horas diarias	6
Horario esperado	30 horas semanales
Cantidad de docentes	21
Cantidad de grados	6 (de 6 a 11)
Cantidad de grupos	13 (3 de 6, 2 de cada uno de los restantes)
Cantidad de áreas	12
Cantidad de asignaturas	23

Tabla 36 Configuración de la Hiperheurística

Datos arrojados por heurística de bajo nivel	
Carga total esperada	451
Asignadas	434
Faltantes	17
Asignado forma automática	96.23%
Asignado forma semiautomática	3.77%
Tiempo de ejecución (segundos)	3.1607

Tabla 37 Datos Arrojados de Heurística de bajo nivel

Datos arrojados por Hiperheurística sobre metaheurística	
Grupos programados	13
Docentes programados	21
Horas asignadas	388
Horas esperadas	390
Porcentaje de asignación	99.48%
Cruces docentes	0
Tiempo de ejecución (segundos)	2.2768

Tabla 38 Datos Arrojados Hiperheurística

Docente	Dia					Total
	1	2	3	4	5	
ALBERT	3	3	5	3	2	16
GIOVANNI	4	5	5	2	4	20
SONIA	4	5	3	4	4	20
DIANA MILENA R	1	0	2	3	0	6
PAOLA	4	3	2	1	3	13
RICARDO P	4	4	5	3	2	18
AMANDA D	4	5	4	6	3	22
ANDRES	4	2	5	3	5	19
STELLA T	6	3	4	4	2	19
MANUEL V	4	6	5	4	5	24
JOAQUIN	2	5	5	4	6	22
SERGIO M	3	3	4	4	6	20
DIEGO A	4	5	5	6	5	25
MIGUEL ANGEL G	6	5	4	6	6	27
ANA CRISTINA B	3	4	5	3	3	18
ROGER	3	5	5	4	3	20
MIRYAM T	4	4	3	3	1	15
LISETH	5	5	4	3	5	22
CARLOS T	5	4	1	3	4	17
JOHN	2	2	1	5	3	13
ALVARO	3	0	1	3	5	12
TOTALES	78	78	78	77	77	388

Tabla 39 Muestra de horario de programación docente, carga asignada por docente

<p>grupo 6A</p> <p>1 1 ETICA DIANA MILENA RAMIREZ</p> <p>1 2 ESPANOL PAOLA</p> <p>1 3 BIOLOGIA STELLA TEJADA</p> <p>1 4 ESPANOL PAOLA</p> <p>1 5 BIOLOGIA STELLA TEJADA</p> <p>1 6 ESPANOL PAOLA</p>	<p>grupo 6B</p> <p>1 1 DIBUJO MIGUEL ANGEL GOMEZ</p> <p>1 2 SOCIALES RICARDO PIEDRAHITA</p> <p>1 3 SOCIALES RICARDO PIEDRAHITA</p> <p>1 4 SOCIALES RICARDO PIEDRAHITA</p> <p>1 5 DIBUJO MIGUEL ANGEL GOMEZ</p> <p>1 6 INGLES ALBERT MENA</p>
<p>grupo 6C</p> <p>1 1 ESPANOL PAOLA</p> <p>1 2 MATEMATICAS GIOVANNI OSORIO</p> <p>1 3 MATEMATICAS GIOVANNI OSORIO</p> <p>1 4 DIBUJO MIGUEL ANGEL GOMEZ</p> <p>1 5 SOCIALES RICARDO PIEDRAHITA</p> <p>1 6 MATEMATICAS GIOVANNI OSORIO</p>	
<p>grupo 7A</p> <p>1 1 BIOLOGIA STELLA TEJADA</p> <p>1 2 BIOLOGIA STELLA TEJADA</p> <p>1 3 INGLES ALBERT MENA</p> <p>1 4 BIOLOGIA STELLA TEJADA</p> <p>1 5 MATEMATICAS GIOVANNI OSORIO</p> <p>1 6 ETICA AMANDA DELGADO</p>	<p>grupo 7B</p> <p>1 1 BIOLOGIA LISETH</p> <p>1 2 TECNOLOGIA SONIA</p> <p>1 3 BIOLOGIA LISETH</p> <p>1 4 ETICA AMANDA DELGADO</p> <p>1 5 EDUCACION FISICA ANDRES</p> <p>1 6 TECNOLOGIA SONIA</p>
<p>grupo 8A</p> <p>1 1 TECNOLOGIA SONIA</p> <p>1 2 BIOLOGIA LISETH</p> <p>1 3 DIBUJO MIGUEL ANGEL GOMEZ</p> <p>1 4 INGLES DIEGO ALVAREZ</p> <p>1 5 BIOLOGIA LISETH</p> <p>1 6 EDUCACION FISICA ANDRES</p>	<p>grupo 8B</p> <p>1 1 EDUCACION FISICA ANDRES</p> <p>1 2 DIBUJO MIGUEL ANGEL GOMEZ</p> <p>1 3 TECNOLOGIA SONIA</p> <p>1 4 EDUCACION FISICA ANDRES</p> <p>1 5 RELIGION ALBERT MENA</p> <p>1 6 DIBUJO MIGUEL ANGEL GOMEZ</p>
<p>grupo 9A</p> <p>1 1 SOCIALES MANUEL VELASQUEZ</p> <p>1 2 ESPANOL ROGER</p> <p>1 3 SOCIALES MANUEL VELASQUEZ</p> <p>1 4 ESPANOL ROGER</p> <p>1 5 ETICA AMANDA DELGADO</p> <p>1 6 SOCIALES MANUEL VELASQUEZ</p>	<p>grupo 9B</p> <p>1 1 DIBUJO ANA CRISTINA BEDOYA</p> <p>1 2 DIBUJO ANA CRISTINA BEDOYA</p> <p>1 3 ESPANOL CARLOS TABARES</p> <p>1 4 SOCIALES MANUEL VELASQUEZ</p> <p>1 5 INGLES DIEGO ALVAREZ</p> <p>1 6 ESPANOL CARLOS TABARES</p>
<p>grupo 10A</p> <p>1 1 ESPANOL CARLOS TABARES</p> <p>1 2 MATEMATICAS JOAQUIN</p> <p>1 3 FISICA ALVARO</p> <p>1 4 ESPANOL CARLOS TABARES</p> <p>1 5 MATEMATICAS JOAQUIN</p> <p>1 6 ARTES ANA CRISTINA BEDOYA</p>	<p>grupo 10B</p> <p>1 1 MATEMATICAS SERGIO MONTENEGRO</p> <p>1 2 MATEMATICAS SERGIO MONTENEGRO</p> <p>1 3 RELIGION ROGER</p> <p>1 4 FISICA ALVARO</p> <p>1 5 MATEMATICAS SERGIO MONTENEGRO</p> <p>1 6 DISENO JOHN</p>
<p>grupo 11^a</p> <p>1 1 TECNICAS DE IMP JOHN</p> <p>1 2 INGLES DIEGO ALVAREZ</p> <p>1 3 ETICA AMANDA DELGADO</p> <p>1 4 EDUCACION FISICA MIRYAM TABARES</p> <p>1 5 EDUCACION FISICA MIRYAM TABARES</p> <p>1 6 EDUCACION FISICA MIRYAM TABARES</p>	<p>grupo 11B</p> <p>1 1 EDUCACION FISICA MIRYAM TABARES</p> <p>1 2 FISICA ALVARO</p> <p>1 3 INGLES DIEGO ALVAREZ</p> <p>1 4 LAB. QUIMICA LISETH</p> <p>1 5 ESPANOL CARLOS TABARES</p> <p>1 6 ANALISIS DE PREP STELLA TEJADA</p>

Tabla 40 Muestra de horario generado para el día lunes de todos los grupos

CAPITULO 5

CONCLUSIONES Y BIBLIOGRAFIA

5.1 CONCLUSIONES

✓ Las heurísticas de bajo nivel pueden aportar mucho a la solución de problemas de difícil tratamiento siempre y cuando cuenten con la ayuda humana para guiarlas en la búsqueda de soluciones mejores.

✓ Las metaheurísticas como los algoritmos genéticos presentan un problema similar a las heurísticas de bajo nivel, pues es necesario ejecutarlas varias veces para garantizar el hallazgo de buenos resultados y al igual que las heurísticas de bajo nivel funcionan mejor con supervisión humana.

✓ El mayor aporte de la hiperheurística es poder sustituir la supervisión humana durante el uso de las heurísticas de bajo nivel.

✓ Al dividir el problema en dos partes se ataca el problema de la complejidad al reducir las variables y por ende el espacio de búsqueda.

✓ La heurística de bajo nivel seleccionado (programar horas primero) cumple con suficiencia el papel de asignar la carga docente a los distintos grupos en un tiempo relativamente corto.

✓ La hiperheurística al modificar el espacio de búsqueda antes de lanzar el algoritmo genético permite encontrar resultados puntuales de forma eficiente.

✓ El uso que da la hiperheurística al algoritmo genético al ejecutarlo exclusivamente para encontrar franjas horarias es una buena estrategia para

agilizar el proceso de asignación horario siempre que se tenga la carga docente asociada a cada grupo.

- ✓ Las bases de datos no relacionales como unqlite y los diccionarios en el lenguaje Python permiten acoplar las bases de datos y los registros empleados en el modelo de forma natural.

5.2 Trabajo futuro

Como trabajo futuro es posible investigar y encontrar resultados en:

- ✓ Un horario para jornada extendida, puesto que en este ejercicio solo se trabajó la jornada escolar general para todos los colegios.

- ✓ Ajustar la hiperheurística para que admita más restricciones como definir franjas horarias especiales para casos como permisos por lactancia o estudio.

- ✓ Afinar el modelo de tal forma que se pueda alcanzar un 100% con asistencia humana.

- ✓ Construir una aplicación que permita llevar a las instituciones educativas la implementación para resolver sus problemas de asignación horaria.

- ✓ Investigar con otras heurísticas de bajo nivel y probar resultados.

- ✓ Investigar con otras metaheurísticas y probar resultados

5.3 Bibliografía

[1]Alfonso, M. (2001). *Un Modelo de Integración y Técnicas de Clausura y CSP de Restricciones Temporales Aplicada a Problemas de Scheduling Tesis Doctoral (Tesis Doctoral)*. 20.

- [2]Almutairi , A., Ozcan, E., Jackson, W. G., & Kheiri, A. (2016). Performance Of Selection Hyper-heuristics On the Extended Hyflex Domains. *Computer and Information Sciences 31st International Symposium*, 154-162.
- [3]Burke, E. K., Gendreau, M., Hyde, M., Kedall, G., Ochoa, G., Ozcan, E., & Qu, R. (2013). Hyper-heuristics a Survey of the State of the Art. *Journal of the Operational Research Society Advance Publication*, 1-22.
- [4]Castrillon, O. D., Sarache, W. A., & Giraldo, J. A. (2010). Design of Hyperheuristic for Production Scheduling In Job Shop Environments. *Ingeniare Revista Chilena de Ingeniería*, 18(2), 203-214.
- [5]Cichowicz, T., Drozdowski, M., Frankiewicz, M., Pawlak, G., Rytwinski, F., & Wasilewski, J. (2012). Hyper-heuristics For Cross-domain search. *Bulletin Of The Polish Academy Of Sciences*, 60(4), 801-808.
- [6]Eliasay, A. (2014). Evolving Hyper Heuristic Using Genetic Programming. *Ben-Gurion University of the Negev* , 1-5.
- [7]Elyasaf, A. (2014). Evolving Hyper-heuristics Using Genetic Programming (Tesis Doctoral). Universidad Ben-Gurión del Néguev, Beersheba, Israel.
- [8]Escamilla Fuster, J. (2016). Eficacia Energética y Robustez en Problemas de Scheduling(Tesis Doctoral). Universidad Politecnica de Valencia, Valencia, España.
- [9]Esquivel T, L. L. (2014). Modelo Matemático Para La Programación De un Horario Escolar con Multi-localización De Docentes(Tesis Maestria). Universidad del Valle ,Santiago de Cali, Colombia.
- [10]Gallego Rendón, R. A., Escobar Zuluaga, A. H., & Toro Ocampo, E. M. (2008). *Técnicas Metaheurísticas de Optimización*. Pereira, Colombia.
- [11]Glover, F., & Kochenberger, G. (2003). *Handbook Of Metaheuristics*. (K. A. Publishers, Ed.) New Yourk, Estados Unidos.
- [12]Graessle , P., Baumann, H., & Baumann, P. (2005). *UML.2.0 in Action A Project based Tutorial*. (P. P. Ltd., Ed.)
- [13]Herranz Gómez, R. (2014). Bases de Datos NoSQL: Arquitectura y Ejemplos de Aplicación(Proyecto de Grado). Universidad Carlos III de Madrid, Madrid, España.
- [14]Kheiri, A. (2014). Multi-stage Hyper-heuristics For Optimisation Problems. University Of Noringram, Mlaysia, China.
- [15]Lazo Eche, E., Gutierrez Segura, F., & Vergara Moreno, E. (2016). Un Algoritmo Eficiente para Problemas de Single Machine con Tiempos de

Procesamiento Difusos. *Revista Cubana de Ciencias Informáticas*, 10(4), 139-153.

[16]Rumbaugh, J., Jacobson, I., & Booch, G. (2005). *The Unified Modeling Language Reference Manual* (2nd ed.). Boston, Estados Unidos: Pearson.

[17]Ryser Welch, P., & F Miller, J. (2014). A Review Of Hyper-heuristic Frameworks. *Researchgate*. Obtenido de <http://www.researchgate.net/publication/20161361800>

[18]Sastry K., Goldberg D., Kendall G. (2005) Genetic Algorithms. In: Burke E.K., Kendall G. (eds) *Search Methodologies*. Springer, Boston, MA.

[19]Salazar Hornig, E., & Torres Pérez, D. (2016). Aplicación de un Algoritmo ACO al Problemas de Flowshop Flexible con tiempos de Preparación dependientes de la Secuencias y Minimización de la Tardanza Total. *Ingeniare Revista Chilena de Ingeniería*, 24(3), 502-509.

5.4 Webgrafia

[20] *Lindon System INC*. (s.f.). Recuperado el 27 de 11 de 2017, de <http://www.lindo.com/index.php/products/lingo-and-optimization-modeling>

[21] *Lucidchart*. (s.f.). Recuperado el 27 de 11 de 2017, de <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>

Anexo 1

Datos generados por la aplicación.

A1. Configuración inicial.

A continuación se muestran los archivos empleados como restricciones iniciales.

area; asignaturas

ARTES ; ['ARTES ', 'DIBUJO']
 MATEMATICAS ; ['MATEMATICAS', 'CONTABILIDAD', 'ESTADISTICA']
 EDUCACION FISICA ; ['EDUCACION FISICA']
 HUMANIDADES ; ['ESPAÑOL']
 TECNOLOGIA E INF ; ['TECNOLOGIA']
 C.POLITICAS ; ['GESTION ', 'ANALISIS DE PREP']
 EDUCACION ETICA ; ['ETICA', 'CATEDRA DE PAZ']
 EDUCACION RELIGIOSA ; ['RELIGION']
 L. EXTRANJERAS ; ['INGLES']
 MODALIDAD ; ['DISEÑO', 'TECNICAS DE IMP ', 'SEG. INDUSTRIAL']
 C. NATURALES ; ['BIOLOGIA', 'QUIMICA', 'LAB. QUIMICA', 'FISICA']
 C SOCIALES ; ['SOCIALES', 'FILOSOFIA']

Archivo: area_asignaturas.txt

area; docentes

ARTES ; ['JOHN', 'MIGUEL ANGEL GOMEZ', 'ANA CRISTINA BEDOYA', 'AMANDA DELGADO']
 HUMANIDADES ; ['PAOLA', 'CARLOS TABARES', 'ROGER ', 'DIANA MILENA RAMIREZ']
 EDUCACION FISICA ; ['ANDRES', 'MIRYAM TABARES']
 MATEMATICAS ; ['GIOVANNI OSORIO', 'JOAQUIN', 'SERGIO MONTENEGRO']
 C. POLITICAS ; ['SERGIO MONTENEGRO', 'STELLA TEJADA']
 TECNOLOGIA E INF ; ['SONIA', 'MIRYAM TABARES']
 EDUCACION ETICA ; ['CARLOS TABARES', 'ROGER ', 'MIGUEL ANGEL GOMEZ', 'AMANDA DELGADO',
 'MIRYAM TABARES', 'DIANA MILENA RAMIREZ', 'JOHN']
 EDUCACION RELIGIOSA ; ['AMANDA DELGADO', 'JOHN']
 L. EXTRANJERAS ; ['DIEGO ALVAREZ', 'ALBERT MENA']
 MODALIDAD ; ['ANA CRISTINA BEDOYA', 'JOHN']
 C. NATURALES ; ['ALVARO ', 'LISETH ', 'STELLA TEJADA']
 C SOCIALES ; ['MANUEL VELASQUEZ', 'RICARDO PIEDRAHITA']

Archivo: area_docentes

docente; limite

ALBERT MENA ; 22
 GIOVANNI OSORIO ; 26
 SONIA ; 22
 DIANA MILENA RAMIREZ ; 6
 PAOLA ; 22
 RICARDO PIEDRAHITA ; 22
 AMANDA DELGADO ; 22
 ANDRES ; 22

STELLA TEJADA ; 26
 MANUEL VELASQUEZ ; 26
 JOAQUIN ; 24
 SERGIO MONTENEGRO ; 22
 DIEGO ALVAREZ ; 26
 MIGUEL ANGEL GOMEZ ; 22
 ANA CRISTINA BEDOYA ; 22
 ROGER ; 22
 MIRYAM TABARES ; 10
 LISETH ; 24
 CARLOS TABARES ; 22
 JOHN ; 22
 ALVARO ; 22

Archivo: docente_limite.txt

grado; asignaturas

11 ; [['TECNICAS DE IMP ', 'ARTES ', 'TECNOLOGIA', 'FILOSOFIA', 'INGLES', 'EDUCACION FISICA', 'ESTADISTICA', 'ANALISIS DE PREP', 'ESPAÑOL ', 'LAB. QUIMICA', 'GESTION', 'CATEDRA DE PAZ', 'RELIGION', 'DISEÑO', 'MATEMATICAS', 'FISICA', 'QUIMICA', 'ETICA'], [2, 2, 2, 2, 3, 2, 1, 1, 4, 1, 1, 1, 1, 3, 4, 3, 4, 1]]
 10 ; [['TECNICAS DE IMP ', 'DISEÑO', 'TECNOLOGIA', 'FILOSOFIA', 'INGLES', 'EDUCACION FISICA', 'ESTADISTICA', 'CONTABILIDAD', 'LAB. QUIMICA', 'CATEDRA DE PAZ', 'RELIGION', 'ETICA', 'ARTES ', 'MATEMATICAS', 'FISICA', 'QUIMICA', 'SEG. INDUSTRIAL', 'ESPAÑOL '], [2, 4, 2, 2, 3, 2, 1, 1, 1, 1, 1, 1, 2, 4, 3, 4, 1, 4]]
 7 ; [['BIOLOGIA', 'DIBUJO', 'ETICA', 'TECNOLOGIA', 'SOCIALES', 'INGLES', 'EDUCACION FISICA', 'MATEMATICAS', 'ESPAÑOL ', 'ARTES ', 'RELIGION'], [4, 1, 1, 2, 4, 3, 2, 5, 5, 2, 1]]
 6 ; [['SOCIALES', 'ETICA', 'DIBUJO', 'TECNOLOGIA', 'INGLES', 'EDUCACION FISICA', 'MATEMATICAS', 'ESPAÑOL ', 'BIOLOGIA', 'ARTES ', 'RELIGION'], [6, 1, 1, 2, 5, 2, 7, 7, 5, 2, 1]]
 9 ; [['TECNOLOGIA', 'EDUCACION FISICA', 'ESPAÑOL ', 'BIOLOGIA', 'DIBUJO', 'RELIGION', 'ARTES ', 'SOCIALES', 'INGLES', 'MATEMATICAS', 'ETICA'], [2, 2, 5, 4, 1, 1, 2, 4, 3, 5, 1]]
 8 ; [['SOCIALES', 'ETICA', 'BIOLOGIA', 'DIBUJO', 'TECNOLOGIA', 'INGLES', 'MATEMATICAS', 'ESPAÑOL ', 'ARTES ', 'RELIGION', 'EDUCACION FISICA'], [4, 1, 4, 1, 2, 3, 5, 5, 2, 1, 2]]

Archivo: grado_asignaturas.txt

grado:grupos

6:['A', 'B', 'C']
 7:['A', 'B']
 8:['A', 'B']
 9:['A', 'B']
 10:['A', 'B']
 11:['A', 'B']

Archivo: grado_grupos.txt

A2. Salida del sistema

La salida del sistema genera información almacenada en varios archivos y contiene la programación por grupo con descripción de asignatura y docente.

dia 1
 1 1 INGLES ALBERT MENA
 1 2 ETICA DIANA MILENA RAMIREZ

1 3 BIOLOGIA STELLA TEJADA
 1 4 MATEMATICAS GIOVANNI OSORIO
 1 5 INGLES ALBERT MENA
 1 6 ESPANOL PAOLA

dia 2

2 1 SOCIALES RICARDO PIEDRAHITA
 2 2 INGLES ALBERT MENA
 2 3 SOCIALES RICARDO PIEDRAHITA
 2 4 INGLES ALBERT MENA
 2 5 SOCIALES RICARDO PIEDRAHITA
 2 6 INGLES ALBERT MENA

dia 3

3 1 SOCIALES RICARDO PIEDRAHITA
 3 2 TECNOLOGIA SONIA
 3 3 ESPANOL PAOLA
 3 4 SOCIALES RICARDO PIEDRAHITA
 3 5 BIOLOGIA STELLA TEJADA
 3 6 ESPANOL PAOLA

dia 4

4 1 ESPANOL PAOLA
 4 2 SOCIALES RICARDO PIEDRAHITA
 4 3 EDUCACION FISICA ANDRES
 4 4 DIBUJO AMANDA DELGADO
 4 5 EDUCACION FISICA ANDRES
 4 6 RELIGION JOHN

dia 5

5 1 ESPANOL PAOLA
 5 2 MATEMATICAS GIOVANNI OSORIO
 5 3 TECNOLOGIA SONIA
 5 4 BIOLOGIA STELLA TEJADA
 5 5 MATEMATICAS GIOVANNI OSORIO
 5 6 MATEMATICAS GIOVANNI OSORIO

Archivo: salida_alt_6A.txt

dia 1

1 1 ESPANOL PAOLA
 1 2 SOCIALES RICARDO PIEDRAHITA
 1 3 EDUCACION FISICA ANDRES
 1 4 ESPANOL PAOLA
 1 5 BIOLOGIA STELLA TEJADA
 1 6 RELIGION JOHN

dia 2

2 1 TECNOLOGIA SONIA
 2 2 ETICA MIGUEL ANGEL GOMEZ
 2 3 BIOLOGIA STELLA TEJADA
 2 4 ESPANOL PAOLA
 2 5 BIOLOGIA STELLA TEJADA
 2 6 MATEMATICAS GIOVANNI OSORIO

dia 3

3 1 TECNOLOGIA SONIA
3 2 ESPANOL PAOLA
3 3 INGLES ALBERT MENA
3 4 INGLES ALBERT MENA
3 5 INGLES ALBERT MENA
3 6 EDUCACION FISICA ANDRES

dia 4

4 1 MATEMATICAS GIOVANNI OSORIO
4 2 BIOLOGIA STELLA TEJADA
4 3 SOCIALES RICARDO PIEDRAHITA
4 4 MATEMATICAS GIOVANNI OSORIO
4 5 INGLES ALBERT MENA
4 6 DIBUJO AMANDA DELGADO

dia 5

5 1 INGLES ALBERT MENA
5 2 SOCIALES RICARDO PIEDRAHITA
5 3 ESPANOL PAOLA
5 4 ESPANOL PAOLA
5 5 SOCIALES RICARDO PIEDRAHITA
5 6 ESPANOL PAOLA

Archivo: salida_alt_6B.txt

dia 1

1 1 MATEMATICAS GIOVANNI OSORIO
1 2 ETICA MIGUEL ANGEL GOMEZ
1 3 INGLES ALBERT MENA
1 4 INGLES ALBERT MENA
1 5 ESPANOL PAOLA
1 6 EDUCACION FISICA ANDRES

dia 2

2 1 MATEMATICAS GIOVANNI OSORIO
2 2 ESPANOL PAOLA
2 3 DIBUJO AMANDA DELGADO
2 4 BIOLOGIA STELLA TEJADA
2 5 ESPANOL PAOLA
2 6 ESPANOL PAOLA

dia 3

3 1 ESPANOL PAOLA
3 2 RELIGION JOHN
3 3 SOCIALES RICARDO PIEDRAHITA
3 4 DIBUJO AMANDA DELGADO
3 5 MATEMATICAS GIOVANNI OSORIO
3 6 INGLES ALBERT MENA

dia 4

4 1 TECNOLOGIA SONIA
4 2 DIBUJO AMANDA DELGADO
4 3 ESPANOL PAOLA
4 4 SOCIALES RICARDO PIEDRAHITA
4 5 BIOLOGIA STELLA TEJADA

4 6 ESPANOL PAOLA

dia 5

5 1 MATEMATICAS GIOVANNI OSORIO
 5 2 BIOLOGIA STELLA TEJADA
 5 3 SOCIALES RICARDO PIEDRAHITA
 5 4 INGLES ALBERT MENA
 5 5 INGLES ALBERT MENA
 5 6 EDUCACION FISICA ANDRES

Archivo: salida_alt_6C.txt

dia 1

1 1 DIBUJO AMANDA DELGADO
 1 2 BIOLOGIA STELLA TEJADA
 1 3 ETICA MIGUEL ANGEL GOMEZ
 1 4 DIBUJO AMANDA DELGADO
 1 5 MATEMATICAS GIOVANNI OSORIO
 1 6 TECNOLOGIA SONIA

dia 2

2 1 EDUCACION FISICA ANDRES
 2 2 DIBUJO AMANDA DELGADO
 2 3 ESPANOL DIANA MILENA RAMIREZ
 2 4 RELIGION JOHN
 2 5 ESPANOL DIANA MILENA RAMIREZ
 2 6 ESPANOL DIANA MILENA RAMIREZ

dia 3

3 1 ESPANOL DIANA MILENA RAMIREZ
 3 2 ESPANOL DIANA MILENA RAMIREZ
 3 3 MATEMATICAS GIOVANNI OSORIO
 3 4 TECNOLOGIA SONIA
 3 5 DIBUJO AMANDA DELGADO
 3 6 SOCIALES RICARDO PIEDRAHITA

dia 4

4 1 INGLES ALBERT MENA
 4 2 MATEMATICAS GIOVANNI OSORIO
 4 3 MATEMATICAS GIOVANNI OSORIO
 4 4 INGLES ALBERT MENA
 4 5 SOCIALES RICARDO PIEDRAHITA
 4 6 SOCIALES RICARDO PIEDRAHITA

dia 5

5 1 EDUCACION FISICA ANDRES
 5 2 INGLES ALBERT MENA
 5 3 BIOLOGIA STELLA TEJADA
 5 4 MATEMATICAS GIOVANNI OSORIO
 5 5 BIOLOGIA STELLA TEJADA
 5 6 SOCIALES RICARDO PIEDRAHITA

Archivo: salida_alt_7A.txt

dia 1

1 1 ETICA MIGUEL ANGEL GOMEZ

1 2 INGLES ALBERT MENA
1 3 RELIGION JOHN
1 4 ESPANOL ROGER
1 5 TECNOLOGIA SONIA
1 6 BIOLOGIA STELLA TEJADA

dia 2

2 1 INGLES ALBERT MENA
2 2 BIOLOGIA STELLA TEJADA
2 3 MATEMATICAS JOAQUIN
2 4 EDUCACION FISICA ANDRES
2 5 INGLES ALBERT MENA
2 6 DIBUJO AMANDA DELGADO

dia 3

3 1 MATEMATICAS JOAQUIN
3 2 BIOLOGIA STELLA TEJADA
3 3 SOCIALES MANUEL VELASQUEZ
3 4 BIOLOGIA STELLA TEJADA
3 5 MATEMATICAS JOAQUIN
3 6 TECNOLOGIA SONIA

dia 4

4 1 ESPANOL ROGER
4 2 MATEMATICAS JOAQUIN
4 3 ESPANOL ROGER
4 4 EDUCACION FISICA ANDRES
4 5 ESPANOL ROGER
4 6 ESPANOL ROGER

dia 5

5 1 DIBUJO AMANDA DELGADO
5 2 DIBUJO AMANDA DELGADO
5 3 MATEMATICAS JOAQUIN
5 4 SOCIALES MANUEL VELASQUEZ
5 6 SOCIALES MANUEL VELASQUEZ

Archivo: salida_alt_7B.txt

dia 1

1 1 TECNOLOGIA SONIA
1 2 TECNOLOGIA SONIA
1 3 ESPANOL ROGER
1 4 ETICA MIGUEL ANGEL GOMEZ
1 5 DIBUJO AMANDA DELGADO
1 6 DIBUJO AMANDA DELGADO

dia 2

2 1 BIOLOGIA LISETH
2 2 RELIGION JOHN
2 3 ESPANOL ROGER
2 4 BIOLOGIA LISETH
2 5 DIBUJO AMANDA DELGADO
2 6 INGLES DIEGO ALVAREZ

dia 3

3 1 EDUCACION FISICA ANDRES
 3 2 DIBUJO AMANDA DELGADO
 3 3 BIOLOGIA LISETH
 3 4 ESPANOL ROGER
 3 5 SOCIALES MANUEL VELASQUEZ
 3 6 MATEMATICAS JOAQUIN

dia 4

4 1 SOCIALES MANUEL VELASQUEZ
 4 2 EDUCACION FISICA ANDRES
 4 3 BIOLOGIA LISETH
 4 4 SOCIALES MANUEL VELASQUEZ
 4 5 INGLES DIEGO ALVAREZ

dia 5

5 1 SOCIALES MANUEL VELASQUEZ
 5 2 ESPANOL ROGER
 5 4 DIBUJO AMANDA DELGADO
 5 5 INGLES DIEGO ALVAREZ
 5 6 DIBUJO AMANDA DELGADO

Archivo: salida_alt_8A.txt

dia 1

1 1 EDUCACION FISICA ANDRES
 1 2 DIBUJO AMANDA DELGADO
 1 3 DIBUJO AMANDA DELGADO
 1 4 SOCIALES MANUEL VELASQUEZ
 1 5 MATEMATICAS JOAQUIN
 1 6 ETICA MIGUEL ANGEL GOMEZ

dia 2

2 1 DIBUJO AMANDA DELGADO
 2 2 BIOLOGIA LISETH
 2 3 RELIGION JOHN
 2 4 ESPANOL ROGER
 2 5 MATEMATICAS JOAQUIN
 2 6 TECNOLOGIA SONIA

dia 3

3 1 DIBUJO AMANDA DELGADO
 3 2 SOCIALES MANUEL VELASQUEZ
 3 3 ESPANOL ROGER
 3 4 INGLES DIEGO ALVAREZ
 3 5 ESPANOL ROGER
 3 6 DIBUJO AMANDA DELGADO

dia 4

4 1 EDUCACION FISICA ANDRES
 4 2 TECNOLOGIA SONIA
 4 3 INGLES DIEGO ALVAREZ
 4 4 INGLES DIEGO ALVAREZ
 4 5 BIOLOGIA LISETH
 4 6 BIOLOGIA LISETH

dia 5

5 1 MATEMATICAS JOAQUIN
5 2 BIOLOGIA LISETH
5 3 DIBUJO AMANDA DELGADO
5 4 ESPANOL ROGER
5 5 DIBUJO AMANDA DELGADO
5 6 ESPANOL ROGER

Archivo: salida_alt_8B.txt

dia 1
1 1 SOCIALES MANUEL VELASQUEZ
1 2 RELIGION JOHN
1 3 BIOLOGIA LISETH
1 4 TECNOLOGIA SONIA
1 5 ESPANOL ROGER
1 6 INGLES DIEGO ALVAREZ

dia 2
2 1 ARTES MIGUEL ANGEL GOMEZ
2 2 MATEMATICAS JOAQUIN
2 3 INGLES DIEGO ALVAREZ
2 4 MATEMATICAS JOAQUIN
2 5 SOCIALES MANUEL VELASQUEZ
2 6 ARTES MIGUEL ANGEL GOMEZ

dia 3
3 1 ESPANOL ROGER
3 2 BIOLOGIA LISETH
3 3 TECNOLOGIA SONIA
3 4 EDUCACION FISICA ANDRES
3 5 BIOLOGIA LISETH
3 6 ESPANOL ROGER

dia 4
4 1 BIOLOGIA LISETH
4 2 ARTES MIGUEL ANGEL GOMEZ
4 3 MATEMATICAS JOAQUIN
4 4 ESPANOL ROGER
4 5 DIBUJO AMANDA DELGADO
4 6 INGLES DIEGO ALVAREZ

dia 5
5 1 ARTES MIGUEL ANGEL GOMEZ
5 2 SOCIALES MANUEL VELASQUEZ
5 3 ESPANOL ROGER
5 4 EDUCACION FISICA ANDRES

Archivo: salida_alt_9A.txt

dia 1
1 1 ESPANOL CARLOS TABARES
1 2 MATEMATICAS SERGIO MONTENEGRO
1 3 MATEMATICAS SERGIO MONTENEGRO
1 4 INGLES DIEGO ALVAREZ
1 5 RELIGION JOHN
1 6 ESPANOL CARLOS TABARES

dia 2

2 1 ESPANOL CARLOS TABARES
2 2 ESPANOL CARLOS TABARES
2 3 BIOLOGIA LISETH
2 4 INGLES DIEGO ALVAREZ
2 5 EDUCACION FISICA ANDRES
2 6 MATEMATICAS SERGIO MONTENEGRO

dia 3

3 1 BIOLOGIA LISETH
3 2 ESPANOL CARLOS TABARES
3 3 INGLES DIEGO ALVAREZ
3 4 BIOLOGIA LISETH
3 5 EDUCACION FISICA ANDRES
3 6 MATEMATICAS SERGIO MONTENEGRO

dia 4

4 1 MATEMATICAS SERGIO MONTENEGRO
4 2 SOCIALES MANUEL VELASQUEZ
4 3 SOCIALES MANUEL VELASQUEZ
4 4 BIOLOGIA LISETH
4 5 TECNOLOGIA SONIA
4 6 SOCIALES MANUEL VELASQUEZ

dia 5

5 1 TECNOLOGIA SONIA
5 2 DIBUJO MIGUEL ANGEL GOMEZ
5 3 SOCIALES MANUEL VELASQUEZ
5 4 DIBUJO MIGUEL ANGEL GOMEZ
5 5 DIBUJO MIGUEL ANGEL GOMEZ
5 6 DIBUJO MIGUEL ANGEL GOMEZ

Archivo: salida_alt_9B.txt

dia 1

1 1 CONTABILIDAD SERGIO MONTENEGRO
1 2 QUIMICA LISETH
1 3 FILOSOFIA MANUEL VELASQUEZ
1 4 QUIMICA LISETH
1 5 QUIMICA LISETH
1 6 MATEMATICAS JOAQUIN

dia 2

2 1 CATEDRA DE PAZ ROGER
2 2 EDUCACION FISICA ANDRES
2 3 TECNICAS DE IMP ANA CRISTINA BEDOYA
2 4 TECNICAS DE IMP ANA CRISTINA BEDOYA
2 5 INGLES DIEGO ALVAREZ
2 6 QUIMICA LISETH

dia 3

3 1 TECNICAS DE IMP ANA CRISTINA BEDOYA
3 2 EDUCACION FISICA ANDRES
3 3 MATEMATICAS JOAQUIN
3 4 CONTABILIDAD SERGIO MONTENEGRO

3 5 TECNOLOGIA SONIA
3 6 INGLES DIEGO ALVAREZ

dia 4

4 1 LAB. QUIMICA STELLA TEJADA
4 2 TECNICAS DE IMP ANA CRISTINA BEDOYA
4 3 TECNOLOGIA SONIA
4 4 RELIGION JOHN
4 5 MATEMATICAS JOAQUIN
4 6 ARTES MIGUEL ANGEL GOMEZ

dia 5

5 1 CONTABILIDAD SERGIO MONTENEGRO
5 2 INGLES DIEGO ALVAREZ
5 3 ARTES MIGUEL ANGEL GOMEZ
5 4 ESPANOL CARLOS TABARES
5 5 TECNICAS DE IMP ANA CRISTINA BEDOYA
5 6 MATEMATICAS JOAQUIN

Archivo: salida_alt_10A.txt

dia 1

1 1 TECNICAS DE IMP ANA CRISTINA BEDOYA
1 2 FILOSOFIA MANUEL VELASQUEZ
1 3 ESPANOL CARLOS TABARES
1 4 EDUCACION FISICA ANDRES
1 5 INGLES DIEGO ALVAREZ
1 6 ETICA ROGER

dia 2

2 1 TECNICAS DE IMP ANA CRISTINA BEDOYA
2 2 MATEMATICAS SERGIO MONTENEGRO
2 3 MATEMATICAS SERGIO MONTENEGRO
2 4 CATEDRA DE PAZ MIRYAM TABARES
2 5 QUIMICA ALVARO
2 6 RELIGION JOHN

dia 3

3 1 MATEMATICAS SERGIO MONTENEGRO
3 2 ARTES MIGUEL ANGEL GOMEZ
3 3 MATEMATICAS SERGIO MONTENEGRO
3 4 ARTES MIGUEL ANGEL GOMEZ
3 5 INGLES DIEGO ALVAREZ
3 6 LAB. QUIMICA STELLA TEJADA

dia 4

4 1 TECNICAS DE IMP ANA CRISTINA BEDOYA
4 2 ESPANOL CARLOS TABARES
4 3 TECNICAS DE IMP ANA CRISTINA BEDOYA
4 4 TECNOLOGIA SONIA
4 5 MATEMATICAS SERGIO MONTENEGRO
4 6 EDUCACION FISICA ANDRES

dia 5

5 1 QUIMICA ALVARO
5 2 ESPANOL CARLOS TABARES

5 3 QUIMICA ALVARO
 5 4 TECNOLOGIA SONIA
 5 5 QUIMICA ALVARO
 5 6 MATEMATICAS SERGIO MONTENEGRO

Archivo: salida_alt_10B.txt

dia 1
 1 1 LAB. QUIMICA STELLA TEJADA
 1 2 QUIMICA ALVARO
 1 3 TECNICAS DE IMP ANA CRISTINA BEDOYA
 1 4 TECNICAS DE IMP ANA CRISTINA BEDOYA
 1 5 TECNICAS DE IMP ANA CRISTINA BEDOYA
 1 6 EDUCACION FISICA MIRYAM TABARES

dia 2
 2 1 RELIGION JOHN
 2 2 INGLES DIEGO ALVAREZ
 2 3 QUIMICA ALVARO
 2 4 ESPANOL CARLOS TABARES
 2 5 TECNICAS DE IMP ANA CRISTINA BEDOYA
 2 6 TECNICAS DE IMP ANA CRISTINA BEDOYA

dia 3
 3 1 ANALISIS DE PREP ALBERT MENA
 3 2 EDUCACION FISICA MIRYAM TABARES
 3 3 TECNICAS DE IMP ANA CRISTINA BEDOYA
 3 4 FILOSOFIA MANUEL VELASQUEZ
 3 5 TECNICAS DE IMP ANA CRISTINA BEDOYA
 3 6 ARTES MIGUEL ANGEL GOMEZ

dia 4
 4 1 QUIMICA ALVARO
 4 2 EDUCACION FISICA MIRYAM TABARES
 4 3 MATEMATICAS SERGIO MONTENEGRO
 4 4 QUIMICA ALVARO
 4 5 FILOSOFIA MANUEL VELASQUEZ
 4 6 ESPANOL CARLOS TABARES

dia 5
 5 1 ESPANOL CARLOS TABARES
 5 2 EDUCACION FISICA MIRYAM TABARES
 5 3 TECNICAS DE IMP ANA CRISTINA BEDOYA
 5 4 EDUCACION FISICA MIRYAM TABARES
 5 6 TECNICAS DE IMP ANA CRISTINA BEDOYA

Archivo: salida_alt_11A.txt

dia 1
 1 1 EDUCACION FISICA MIRYAM TABARES
 1 2 ESPANOL CARLOS TABARES
 1 3 QUIMICA ALVARO
 1 4 MATEMATICAS SERGIO MONTENEGRO
 1 5 ARTES MIGUEL ANGEL GOMEZ
 1 6 QUIMICA ALVARO

dia 2

2 1 QUIMICA ALVARO
2 2 EDUCACION FISICA MIRYAM TABARES
2 3 ESPANOL CARLOS TABARES
2 4 QUIMICA ALVARO
2 5 ETICA JOHN
2 6 EDUCACION FISICA MIRYAM TABARES

dia 3

3 1 ESPANOL CARLOS TABARES
3 2 MATEMATICAS SERGIO MONTENEGRO
3 3 EDUCACION FISICA MIRYAM TABARES
3 4 ESPANOL CARLOS TABARES
3 5 EDUCACION FISICA MIRYAM TABARES
3 6 ETICA JOHN

dia 4

4 1 ARTES MIGUEL ANGEL GOMEZ
4 2 QUIMICA ALVARO
4 3 EDUCACION FISICA MIRYAM TABARES
4 4 MATEMATICAS SERGIO MONTENEGRO
4 5 EDUCACION FISICA MIRYAM TABARES
4 6 QUIMICA ALVARO

dia 5

5 1 TECNICAS DE IMP ANA CRISTINA BEDOYA
5 2 QUIMICA ALVARO
5 3 EDUCACION FISICA MIRYAM TABARES
5 4 MATEMATICAS SERGIO MONTENEGRO
5 5 EDUCACION FISICA MIRYAM TABARES
5 6 QUIMICA ALVARO

Archivo: salida_alt_11B.txt