

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



**Trabajo de Diploma para Optar por el Título de Ingeniero en
Ciencias Informáticas.**



**Título: “Algoritmos para el módulo de Análisis del terreno de
GENESIG: Análisis de rutas.”**

AUTOR: Idalmis Palomino Arronte

TUTOR: Ing. Romanuel Ramón Antunez

Ciudad de La Habana, 29 de Junio de 2010

“Año 52 de la Revolución”

PENSAMIENTO

“No se vive celebrando victorias,
sino superando derrotas.”

Ernesto Che Guevara

DEDICATORIA

A mi mamá Nora y mi papá Héctor, por haber hecho de mí la mujer que soy.

*A mi abuelita illa, por todos sus consejos y que la quiero con la vida. A mi otra abuela Sara,
por comprenderme siempre y por su cariño.*

A mis tías Nancy y Norma, que me han dado lo mejor de ellas, a mi tío Jorge también.

A mis primos Yoslaine y Yosbel por ser como los hermanos que nunca tuve.

Idalmis Palomino Arronte

AGRADECIMIENTOS

Mis más sinceros agradecimientos a La Revolución Cubana y a la Universidad de las Ciencias

Informáticas por darme la oportunidad de formarme como joven profesional.

Agradezco además a mi mamá y a mi papá por consentirme tanto, por apoyarme en todo momento y

porque además hoy también se hacen realidad sus sueños.

A toda mi familia en general por darme lo mejor de ellos.

A mi tutor Romanuel, quien me supo guiar de la mejor manera en la realización de este trabajo. A

mis amistades de la Universidad, principalmente: Ariadna, Dayanelis, Suyen, por ser amigas

incondicionales.

A Antonio, por estar a mi lado cuando más lo necesité.

En fin a todas aquellas personas que de una forma u otra, hicieron posible la realización de este

*trabajo: **Muchas Gracias,***

Idalmis Palomino Arronte

Declaración de Autoría

Declaro que Idalmis Palomino Arronte es la única autora de este trabajo titulado: “**Algoritmos para el módulo de Análisis del terreno de GENESIG: Análisis de rutas.**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo el presente trabajo a los ____ días del mes de ____ del año.

Idalmis Palomino Arronte

Romanuel Ramón Antunez

DATOS DE CONTACTO

Síntesis del Tutor Ing. Romanuel Ramón Antunez:

Profesión: Ingeniero en Ciencias Informáticas

Años de graduado: 2

E-mail: rramon@uci.cu

RESUMEN

El avance y adelanto que han alcanzado las Tecnologías de la Informática y las Comunicaciones (TIC) en la actualidad ha sido bastante notorio. Paralelamente se han ido desarrollando los Sistemas de Información Geográfica (SIG), los cuales han dado lugar a la creación de herramientas y aplicaciones dedicadas al estudio de rutas mínimas. Por la necesidad de contar en la Universidad de las Ciencias Informáticas (UCI) con un producto que sirva como soporte al desarrollo de aplicaciones de estos Sistemas, surge la Plataforma GENESIG.

Esta plataforma constituye un proyecto realizado por Las Fuerzas Armadas Revolucionarias (FAR), un grupo de especialistas de GEOCUBA y UCI. La misma cuenta con algoritmos que muchas veces son pocos e ineficientes en el módulo de Análisis del terreno, específicamente la funcionalidad referente a Análisis de rutas. Esta situación provoca un escaso rendimiento del sistema y un consumo importante de recursos computacionales. Por lo que se hace indispensable el uso de nuevos algoritmos.

En el presente Trabajo de Diploma con título: “Algoritmos para el módulo de Análisis del terreno de GENESIG: Análisis de rutas”, se muestra la información resultante de la investigación y el estudio de algoritmos, que brindan soluciones para hallar las rutas óptimas; permitiendo a su vez, dar solución al proceso de trasladarse de un lugar a otro minimizando costes y recursos humanos. Esta solución propuesta de algoritmos aportará a la institución, grandes ahorros de recursos, como combustibles y tiempo, beneficiando así el desempeño de muchas de las actividades asociadas a esta, directa o indirectamente.

PALABRAS CLAVES

Algoritmos, herramientas, análisis de rutas, rutas mínimas.

Abstract

In actuality the progress and advancement that Informatics and Communications Technology (ICT) has reached have been very noticeable. The growth of the Geographic Information Systems (GIS) has been parallel, which has led to the creation of tools and applications dedicated to the study of minimum routes. Due to the need of being able to count on a product which can serve as a support to the development of the Geographic Information Systems applications in WEB based environments with free technologies, the GENESIG platform is born.

This platform is a Project by the Revolutionary Armed Forces (RAF), a group of GEOCUBA specialists and the University of Computer Sciences. The last mentioned has algorithms which often are few and inefficient in the ground route analysis, specifically in the functionality of Route Analysis. This situation causes poor system performance and significant consumption of computational resources. Therefore, the use of new algorithms becomes essential.

In this dissertation with title “Algorithms for the Analysis module GENESIG field: Analysis of routes” shows the information resulting from the research and the study of algorithms which provide solutions to find the optimal routes; allowing in turn, to solve the process of moving from one place to another. This solution proposed algorithms will provide the institution significant savings in resources such as fuel and time, therefore benefiting the performance of many of the activities associated with this, directly or indirectly.

Keys words

Algorithms, tools, pathways analysis, minimum paths.

Índice General

Introducción	1
CAPÍTULO I.....	6
Fundamentación Teórica	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema	6
1.2.1 Sistemas de Información Geográfica.....	6
1.2.2 Investigación de Operaciones	8
1.2.3 Red	9
1.2.4 Modelo matemático	11
1.2.5 Algoritmo.....	12
1.3 Objeto de Estudio.....	13
1.3.1 Análisis de rutas.....	13
1.3.2 Situación Problemática.....	14
1.4 Conclusiones parciales	14
CAPÍTULO II.....	15
Tendencias y tecnologías	15
2.1 Introducción	15
2.2 Descripción de algoritmos matemáticos	15
2.2.1 Algoritmo de Dijkstra	16
2.2.2 Algoritmo de Floyd-Warshall.....	18
2.2.3 Algoritmo de Prim.....	19
2.2.5 Algoritmo de Ford-Fulkerson	21
2.2.6 Algoritmo de Kruskal	23

2.2.7 Algoritmo de Bellman-Ford.....	24
2.2.8 Algoritmo de búsqueda en anchura (BFS).....	26
2.2.9 Algoritmo de búsqueda en profundidad (DFS).....	28
2.2.10 Problema del trasbordo	29
2.2.11 Problema del Viajante de Comercio	31
2.2.12 Problema de Rutas para Vehículos Capacitados.....	32
2.3 Algoritmos heurísticos: Algoritmo A*	34
2.4 Algoritmos Meta-heurísticos: Búsqueda Tabú	37
2.5 Algoritmos genéticos	39
2.6 Herramientas existentes para el cálculo de las distancias mínimas.....	42
2.6.1 Herramienta Rutas	43
2.6.2 Herramienta Grafos	43
2.6.3 Herramienta PgDijkstra	45
2.6.4 Herramienta PgRouting	45
2.7 Conclusiones parciales	46
CAPÍTULO III.....	47
Presentación de la Solución Propuesta.....	47
3.1 Introducción	47
3.2 Comparación entre las herramientas existentes para el cálculo de las distancias mínimas	47
3.3 Comparación entre algoritmos y modelos matemáticos	49
3.4 Solución propuesta	51
3.5 Validación	53
3.6 Conclusiones parciales	58
CONCLUSIONES GENERALES.....	59

RECOMENDACIONES	60
TRABAJOS CITADOS	61
BIBLIOGRAFÍA CONSULTADA.....	64
ANEXOS.....	¡Error! Marcador no definido.
GLOSARIO DE TÉRMINOS	66

Índice de Figuras

Fig. 1 Representación de un SIG [3]	7
Fig. 2 Representación de una red [11]	11
Fig. 3 Modelo del Problema del Transbordo.....	31
Fig. 4 Modelo del Problema del Viajante de Comercio.....	32
Fig. 5 Modelo matemático de tres subíndices para el problema básico CVRP [23]	34
Fig. 6 Operador de Selección [32].....	41
Fig. 7 Operador de Cruce [32].....	41
Fig. 8 Operador de Mutación [32].....	42
Fig. 9 Análisis de rutas.....	55
Fig. 10 Análisis de rutas.....	56
Fig. 11 Análisis isocrónicos.....	57

Índice de Tablas

Tabla 1. Comparación entre herramientas para el estudio de rutas mínimas	49
Tabla 2. Comparación entre algoritmos y modelos matemáticos	51

Introducción

Con el paso de los años el hombre se ha visto en la necesidad de minimizar el gasto de tiempo y de recursos para de esta forma lograr un mejor aprovechamiento de los mismos. A raíz de esta evolución, surgen las técnicas de programación y los modelos matemáticos, los cuales se han venido aplicando en aras de lograr mejoras en los procesos productivos y sociales.

Los excelentes resultados que las técnicas de programación han aportado hasta ahora, se han hecho muy visibles a través de los años, con el paso del tiempo. De esta forma se ha ido logrando la eficiencia en la asignación de los recursos y en particular donde el factor más importante lo constituye el trabajo del hombre y por consiguiente la productividad y el coste como variables más relevantes.

La matemática posee numerosas ramas, entre las que se encuentra la Investigación de Operaciones (IO), la cual permite ayudar en la toma de decisiones a la hora de planificar actividades futuras, a partir de la escasez de recursos disponibles para dichas actividades. Para ello se basa en el uso de los modelos matemáticos, estadísticas y algoritmos.

Gracias al desarrollo que han alcanzado las TIC, muchos de los problemas que antes no se podían resolver por falta de conocimientos, hoy por hoy, ya se les puede dar solución a través de la Investigación de Operaciones.

Un ejemplo del avance y evolución de las TIC, lo constituyen los SIG. Estos son herramientas que le permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y representar los resultados de todas estas operaciones. [1]

La tecnología de los SIG puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística por nombrar unos pocos. Por ejemplo, un SIG podría permitir a los grupos de emergencia calcular fácilmente los tiempos de respuesta en caso de un desastre natural, el SIG puede ser usado para encontrar los humedales que necesitan

protección contra la contaminación, o pueden ser utilizados por una empresa para ubicar un nuevo negocio y aprovechar las ventajas de una zona de mercado con escasa competencia. [2]

Las FAR junto con un grupo de especialistas de GEOCUBA y la UCI idearon una herramienta informática denominada: Plataforma GENESIG. La misma surge como necesidad de tener un producto que sirva como soporte al desarrollo de aplicaciones de Sistemas de Información Geográfica en entornos Web con tecnologías libres. Estas aplicaciones cumplen además con las especificaciones *OpenGIS* que establece el *Open Geospatial Consortium (OGC)*. A su vez, estas garantizan la interoperabilidad global entre los SIG y en consecuencia con la política de migración a software libre y de soberanía tecnológica que impulsa el país.

Su objetivo fundamental es realizar la representación geo-espacial de la información asociada a negocios específicos y el análisis de dicha información. Esta aplicación debe permitir la toma de decisiones en cualquier entidad de negocio, mediante los servicios de georeferenciación y localización de recursos.

Con el desarrollo de la aplicación GENESIG, los investigadores de la misma, han determinado la necesidad de implementar y utilizar algoritmos para lograr avances más rápidos y consistentes en su articulación. Los algoritmos con los que ya se cuenta, muchas veces son ineficientes en el módulo de Análisis del terreno, específicamente la funcionalidad referente a Análisis de rutas. Esta situación provoca un escaso rendimiento del sistema y un consumo importante de recursos computacionales. Por lo que se hace indispensable el uso de nuevos algoritmos.

Dada la situación planteada, el **problema a resolver** quedaría definido en la siguiente interrogante: ¿Cómo optimizar el proceso de desarrollo del módulo de Análisis del terreno, específicamente la funcionalidad referente a Análisis de rutas, en la plataforma GENESIG?

Partiendo del problema a resolver el **objeto de estudio** se encuentra compuesto por los algoritmos para el análisis de redes. De ahí entonces, que el **campo de acción** radica en la aplicación de algoritmos para el análisis de redes en los sistemas de información geográfica.

Como **idea a defender** se plantea que obtenida una propuesta de algoritmos para el análisis de redes (determinar rutas óptimas) se haría más sencillo y rápido el trabajo de los programadores para el desarrollo de una primera versión del módulo en cuestión.

El **objetivo general** de este trabajo consiste en: Obtener una propuesta de algoritmos para llevar a cabo el análisis de ruta en la plataforma GENESIG.

Para cumplir el objetivo general se realizaron las siguientes **tareas de la investigación**:

1. Definir principales Conceptos y Objeto de Estudio asociados a la investigación.
2. Describir modelos y algoritmos que existen para el cálculo de rutas mínimas.
3. Describir Problemas Tipos o Clásicos que existen para el cálculo de rutas mínimas.
4. Caracterizar las herramientas y tecnologías actuales.
5. Definir modelos y métodos a emplear en la plataforma GENESIG.
6. Definir herramientas a utilizar en el desarrollo de los módulos en la plataforma GENESIG.
7. Identificar posibles mejoras para las herramientas, complementándolas con los algoritmos y modelos matemáticos seleccionados.
8. Documentar la investigación con vistas a su socialización.

Luego de la realización de las tareas antes mencionadas se tuvo el siguiente **resultado posible**:

- Propuesta de algoritmos a implementar en la primera versión de los respectivos módulos de la plataforma GENESIG.

En el desarrollo de la investigación se pusieron de manifiesto algunos métodos tradicionales investigativos, a continuación se muestran los mismos:

Métodos teóricos:

Histórico-lógico: Este método se utilizó para conocer la evolución de los algoritmos de redes y problemas matemáticos de optimización y cubrimiento.

Analítico-Sintético: Este método se utilizó para centrarse en el análisis de las teorías, documentos, permitiendo la selección de los algoritmos más importantes que resuelvan los problemas tipos.

Inductivo-Deductivo. Este método se utilizó con el objetivo de llegar a conocimientos generalizadores que permitan determinar cuál es el algoritmo que permitirá obtener resultados eficientes.

Métodos empíricos:

Entrevistas: En este método se evidencian las entrevistas realizadas al tutor y otros especialistas de la matemática numérica, y matemática aplicada para un mayor conocimiento sobre el uso de los algoritmos de redes.

Capítulos de la investigación

El presente documento está estructurado por 3 capítulos, a continuación se expone brevemente una descripción de cada uno.

Capítulo I Fundamentación Teórica: En este primer capítulo se tratan los aspectos fundamentales para lograr una mejor comprensión del problema planteado. Para ello, se instauran los conceptos más importantes, así como una descripción más concisa de la situación problemática y el objeto de estudio, los cuales constituyen factores de vital importancia para el desarrollo de esta investigación.

Capítulo II Tendencias y Tecnologías: En este capítulo se realiza la descripción de los modelos matemáticos y herramientas existentes dedicadas al estudio de la optimización de rutas. Se caracterizan además los algoritmos enfocados a la obtención de dichos modelos, así como el análisis de problemas clásicos¹, los cuales son empleados en la funcionalidad referente a Análisis de rutas.

¹ **Problemas clásicos:** Son aquellos problemas que buscan la solución más óptima para llegar a un resultado final, cuando se está en presencia de ciertas y determinadas situaciones. Las siguientes constituyen algunas de dichas situaciones: 1-Un origen muchos destinos; 2-Un origen un destino; 3-Problema del viajante; 4-Asignación de recursos.

Capítulo III Descripción de la Solución Propuesta: En este último capítulo se identifican las herramientas que la plataforma GENESIG utilizará con el fin de lograr la optimización de rutas, perfeccionándolas con los algoritmos y modelos matemáticos seleccionados. Se exponen además las posibles mejoras para esta herramienta basada en la solución propuesta.

CAPÍTULO I

Fundamentación Teórica

1.1 Introducción

Este primer capítulo describe los conceptos asociados al dominio del problema con el objetivo de avalar dicha investigación. Se expone además una descripción más concisa de la situación problemática y del objeto de estudio, permitiendo de esta forma crear la base de los conocimientos para los capítulos siguientes.

1.2 Conceptos asociados al dominio del problema

Con el objetivo de lograr que el lector alcance una mayor comprensión y una visión general de los temas tratados en la presente investigación se muestran a continuación un grupo de conceptos. Estos conceptos de una forma u otra son utilizados a lo largo de la realización de este trabajo, por lo que se hace necesario concretar y puntualizar cada uno de ellos.

1.2.1 Sistemas de Información Geográfica

Un SIG es un sistema capaz de almacenar, desplegar y analizar información geo-referenciada. El SIG organiza la información espacial en una serie de capas. Cada capa de información contiene datos con similares características temáticas y estructura gráfica. Los datos de las diferentes capas (Fig. 1) comparten el mismo entorno geográfico, lo que permite la superposición de información. [3]

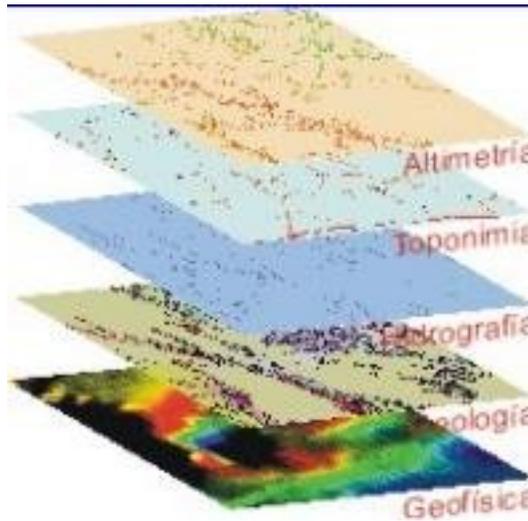


Fig. 1 Representación de un SIG [3]

Una aplicación de los SIG permite relacionar los datos textuales y descriptivos a la información geográfica, para manipular y analizar dichos datos y de esta forma mostrarlos en el territorio a través de mapas digitales. Estos mapas se encuentran constituidos por la superposición de capas de información que representan la región de los acontecimientos y las condiciones expresadas por los datos descriptivos utilizados. [4]

Una definición de SIG bastante admitida es la que presenta en 1990 el Centro Nacional de Información Geográfica y Análisis, el cual expone que un SIG constituye un sistema creado en hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos especialmente referenciados, para resolver problemas complejos de planificación y gestión. [5]

Expuestas las definiciones anteriores, las cuales fueron planteadas por especialistas con un vasto conocimiento en el tema, se puede llegar a la conclusión de que los SIG, constituyen sistemas que son utilizados en el estudio, análisis y representación de datos espaciales. Las funcionalidades que poseen los SIG se encuentran muy ligadas a la visualización, localización y recuperación de elementos u objetos en el espacio. A través de estos sistemas se puede almacenar también dicha información, permitiendo además realizar las diferentes consultas interactivas existentes.

El sistema de información, además está diseñado para trabajar con datos referenciados por coordenadas espaciales o geográficas. En otras palabras, un SIG es a la vez un sistema de base de datos con capacidades específicas para datos referenciados. A estos datos se les puede realizar un conjunto de operaciones matemáticas.

Gracias a las tecnologías que los SIG poseen, muchas son las aplicaciones que durante años se han utilizado en los diferentes campos de la industria. Un ejemplo de ello, lo constituyen las aplicaciones siguientes: [6]

- Cartografía de localizaciones
- Mapas cuantitativos
- Mapas de densidades
- Cálculo de distancias

1.2.2 Investigación de Operaciones

Definir el término investigación de operaciones no es tarea fácil, ya que su evolución permanente hace que sea difícil dar con precisión una definición. La investigación operativa se puede definir como la aplicación de métodos científicos en la mejora de la efectividad en las operaciones, decisiones y gestión. [7]

Con su orientación filosófica *Churchman, Ackoff y Arnoff* en su libro publicado en 1957 sobre la IO², sistematizan los conceptos y presentan de manera general las fases del método de la investigación de operaciones, derivado del método científico. [8] Ellos, definen la investigación de operaciones como la aplicación, por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (hombre-máquina), a fin de que se produzcan soluciones que sirvan mejor a los objetivos de la organización. [9]

Teniendo en cuenta las definiciones planteadas, se puede concluir que la investigación de operaciones consigue darle solución a los problemas de manera indirecta y no como normalmente se hace de forma

² IO: Acrónimo de Investigación de Operaciones.

directa, pasando del problema real a la solución real. Su objetivo es el de ayudar a las empresas a determinar científicamente sus políticas y acciones.

El desarrollo que ha alcanzado actualmente la disciplina de Investigación de Operaciones le ha permitido al hombre ayudarlo en el análisis de la toma de decisiones, a partir de la escasez de recursos. En el momento de optimizar un objetivo definido como por ejemplo la maximización de los beneficios o la minimización de costes, esta ayuda resulta muy importante.

Entre las características que posee la Investigación de Operaciones se pueden resaltar las siguientes:

- Usa el método científico para investigar el problema en cuestión. En particular, el proceso comienza por la observación cuidadosa y la formulación del problema incluyendo la recolección de datos pertinentes.
- Ha desarrollado una serie de técnicas y modelos muy útiles a la Ingeniería de Sistemas. Entre ellos tenemos: la Programación No Lineal, Teoría de Colas, Programación Entera, Programación Dinámica, entre otras.
- Intenta encontrar una mejor solución (llamada solución óptima), para el problema bajo consideración. En lugar de contentarse con mejorar el estado de las cosas, la meta es identificar el mejor curso de acción posible.

1.2.3 Red

No existe un sólo tipo de red, ya que históricamente han existido diferentes tipos de equipos que se han comunicado en varios lenguajes diferentes. La necesidad de contar con múltiples tipos de redes también surge de la heterogeneidad de los medios físicos de transmisión que las une, ya sea que los datos se transfieran de la misma manera (por pulsos eléctricos, haces de luz u ondas electromagnéticas) o que utilicen el mismo tipo de medio físico (como un cable coaxial, pares trenzados o líneas de fibra óptica).
[10]

Es por ello que la palabra red hace referencia a un conjunto de objetos que se encuentran interrelacionados entre sí. Estas conexiones permiten el envío de elementos a dichos objetos. Ejemplo de ello lo constituyen las redes de transporte, las cuales transportan personas de un lugar a otro. También se

encuentran las redes informáticas, que no son más que los equipos conectados unos con otros. Al conjunto de neuronas conectadas entre sí, se le llama redes neuronales.

A partir de lo analizado hasta aquí, se ha podido corroborar algunos de los significados que posee la palabra red. Partiendo entonces, del avance y desarrollo que posee en nuestros días la informática, todos estos tipos de redes mencionados y otros no mencionados pueden ser representados en un grafo³, con el fin de que se logre una buena visualización y un mejor entendimiento del problema en cuestión.

Muchos de los problemas de optimización que existen en la Matemática, pueden ser representados a través de grafos. Las redes se encuentran relacionadas con los grafos, los cuales poseen vértices que están unidos mediante caminos o redes (rutas como se les puede llamar además). Estos vértices tienen valores asignados en sus arcos, los cuales a su vez pueden o no cruzarse. En el caso que no se cruce se plantea que están unidos por una red plana.

Entonces este tipo de red se forma a partir de nudos conectados por hilos. Sobre los hilos se indica el sentido del flujo. Una propiedad fundamental de las redes es que el flujo total que entra en un nudo es igual al flujo total que sale, como se muestra en la Fig. 2. [11]

³ Los grafos son la representación natural de las redes, en las que estamos cada vez más incluidos. Juan C. Dürsteler.
<http://www.infovis.net/printMag.php?num=137&lang=1>

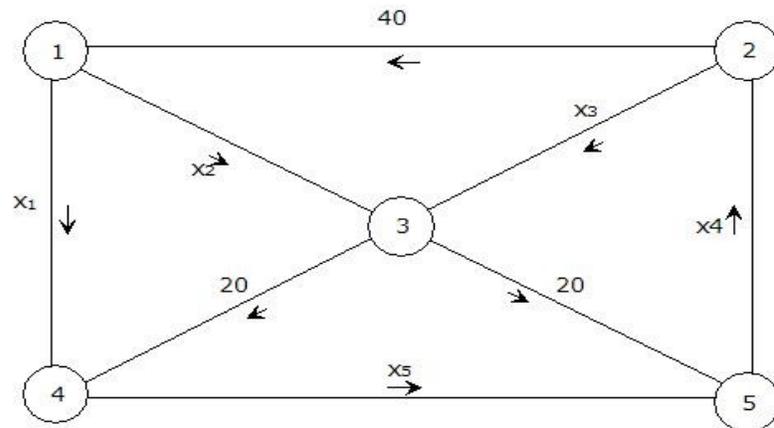


Fig. 2 Representación de una red [11]

1.2.4 Modelo matemático

El Diccionario de la Real Academia Española de la Lengua, en su contenido expone diferentes definiciones y conceptos. El modelo matemático lo define como un esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, que se utiliza para su comprensión y el estudio de su comportamiento.

Un modelado matemático es el proceso mediante el cual un problema tal como aparece en el mundo real se interpreta en términos de símbolos abstractos. La descripción abstracta que incluye una formulación matemática se denomina modelo matemático del problema original,... permitiendo entonces tratar el problema en términos exclusivamente matemáticos. [12]

También se puede afirmar que un modelo es la representación conceptual mediante una descripción cualitativa bien detallada y debido a su uso relacionado con aspectos numéricos puede ser matemático. Los modelos matemáticos se encuentran muy relacionados con los algoritmos ya que estos permiten darle solución a problemas que se nos presentan en la vida diaria.

Por tanto, analizando lo anteriormente expuesto aquí, un modelo matemático no es más que la forma que se utiliza para hacer la descripción matemática dada una situación de la vida cotidiana y que se puede expresar a través de funciones o de relaciones. Es decir la forma de llegar a un modelado simple dada

una descripción compleja. Se puede entonces concluir que un modelo constituye la representación matemática simplificada de cierta y determinada situación de la realidad compleja.

Los elementos constitutivos de los modelos matemáticos son:

- Variables
- Parámetros
- Relaciones Funcionales
- Zona de definición

1.2.5 Algoritmo

La palabra algoritmo proviene del latín *alkhowarizmi*, nombre este de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX. [13]

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problema. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. [14]

En la vida cotidiana muchos algoritmos son empleados para resolver diversos problemas. Por tanto un algoritmo no es más que una serie de pasos organizados de tal forma que se describe el proceso que se debe seguir para darle solución al problema en cuestión.

Los algoritmos según su función se caracterizan en dos tipos:

- **Algoritmos de búsquedas:** son aquellos que se diseñan para encontrar un elemento dentro de una estructura de datos. Estos algoritmos centran su solución en que dado un determinado elemento en un conjunto finito de elementos, puedan decidir si se encuentra o no y por supuesto mostrar la localización dentro de este.
- **Algoritmos de ordenamiento:** no son más que los algoritmos que dado una lista de elementos en una secuencia, permite que se puedan ordenar según el criterio de ordenamiento orientado.

Tipos de algoritmos de redes existentes.

- Algoritmo de Dijkstra (Soluciona el problema de los caminos más cortos de un vértice (origen) a los restantes.)
- Algoritmo de Floyd-Warshall (Soluciona el problema de los caminos más cortos entre todos los pares de vértices.)
- Algoritmo de Prim (Soluciona el problema de encontrar un árbol de expansión mínima.)
- Algoritmo de Kruskal (Soluciona el problema de encontrar un árbol de expansión mínima.)
- Algoritmo de Bellman-Ford
- Algoritmo de búsqueda en anchura (BFS)
- Algoritmo de búsqueda en profundidad (DFS)
- Algoritmo de búsqueda A*
- Algoritmo de Ford-Fulkerson

1.3 Objeto de Estudio

El objeto de estudio de la presente investigación está dado por los algoritmos matemáticos que existen para el Análisis de rutas. En aras de lograr un profundo análisis y que el mismo le facilite al lector un mayor entendimiento de lo que en realidad se plantea, se hace necesario estudiar el Análisis de rutas, como alternativa a tener en cuenta en el momento de visualizar en un mapa la distancia mínima entre unos o varios puntos.

1.3.1 Análisis de rutas

Las rutas constituyen el camino a seguir entre dos o más puntos. En matemática la palabra red es bastante similar a la palabra ruta. El análisis de ruta está relacionado con el análisis de redes, que este último no es más que la selección de rutas óptimas para llegar a un resultado final.

Entonces se puede decir que, el análisis de ruta es el estudio realizado a un grafo conexo, en el que los nodos representan conceptos o elementos relacionados mediante aristas, las cuales reflejan la intercomunicación conceptual de dichos elementos.

1.3.2 Situación Problemática

La plataforma soberana GENESIG, actualmente cuenta con diferentes funcionalidades, las cuales constituyen factores importantes en el desarrollo que deben tener cada uno de sus módulos así como en la representación de la información geográfica de los datos que se recogen en la base de datos. Esta plataforma cuenta con diferentes módulos y herramientas. El Análisis de rutas, es un ejemplo de funcionalidad que se encuentra dentro del módulo de Análisis del terreno. Esta, no se ha podido implementar debido a que los desarrolladores del equipo poseen poco conocimiento y además no cuentan con los algoritmos suficientes que permitan este tipo de modelado.

Esta situación trae consigo el impedimento de algunas funcionalidades que la Plataforma GENESIG posee, lo que entonces dificultaría el rendimiento en:

- Las funcionalidades matemáticas con los modelos, como es el ejemplo de la suma de un modelo con otro, aplicarle una función matemática a uno u otro modelo.
- La unión de modelos que se solapen creando una nueva área determinada.
- Encontrar la ruta óptima de llegar a un determinado lugar dados dos puntos en el mapa.
- Reducción de recursos materiales y costes, así como de esfuerzo humano.

1.4 Conclusiones parciales

En este primer capítulo se pudo definir una serie de conceptos asociados al problema científico, lo que posibilita en el futuro, un mejor entendimiento y una buena comprensión de la presente investigación. La situación problemática fue expuesta de la forma más concisa y centrada posible, en aras de lograr resultados positivos que mejoren la situación actual en la plataforma GENESIG.

CAPÍTULO II

Tendencias y tecnologías

2.1 Introducción

El principal objetivo de este capítulo, radica en la descripción de los modelos matemáticos y en las diferentes herramientas existentes, los cuales contribuyen al estudio de la optimización de rutas. Se caracterizan los diferentes algoritmos enfocados a la obtención de dichos modelos, de tal forma que se hagan corresponder con la solución propuesta.

2.2 Descripción de algoritmos matemáticos

Existen diferentes algoritmos que dan solución a la hora de hallar la ruta óptima de un punto a otro. Dentro de estos algoritmos se encuentran los llamados algoritmos genéticos⁴, los heurísticos⁵, los meta-heurísticos⁶ y otros más que hacen posible dicho objetivo. Más adelante se analizarán ejemplos de estos algoritmos, así como también las facilidades que los mismos proporcionan a quienes lo utilizan.

Todos estos algoritmos sean de un tipo u otro, se encuentran estrechamente relacionados con los modelos matemáticos. A continuación se expone una breve descripción de los algoritmos que comúnmente se utilizan para resolver los problemas de distancias mínimas. Muchos de estos algoritmos son estudiados en numerosas universidades, ejemplo de ello en la Universidad de las Ciencias Informáticas. Su uso garantiza el desarrollo y éxito en los proyectos en los que el cálculo de rutas mínimas sea su principal objetivo.

⁴ **Algoritmos Genéticos:** Son denominados además algoritmos evolutivos. Este tipo de algoritmo se inspira en el desarrollo biológico y en el origen genético-molecular. Realizan un proceso semejante al de la evolución biológica, evolucionando a su vez una determinada población de individuos.

⁵ **Algoritmos Heurísticos:** Constituyen algoritmos que al aplicarse se llega a la solución del problema en un tiempo determinado. En ocasiones encuentra buenas soluciones aunque no se pueda asegurar que sean óptimas.

⁶ **Algoritmos Meta-heurísticos:** Constituyen métodos aproximados para resolver problemas complejos de optimización combinatoria en los que la heurística no permite que se dé una solución satisfactoria.

2.2.1 Algoritmo de Dijkstra

En el año 1956, *Edsger Wybe Dijkstra*⁷ anunció un algoritmo que permitiría encontrar los caminos mínimos en un grafo, a este algoritmo lo nombró como su nombre: “Dijkstra”. El algoritmo de Dijkstra aunque fue diseñado para encontrar la ruta más corta se puede transformar fácilmente para encontrar inspira la ruta más larga (camino crítico), cambiando simplemente su función objetivo. Del mismo modo, se encuentra el árbol máximo desde un nodo origen. [15]

En otras palabras el algoritmo Dijkstra se utiliza para solucionar el problema de los caminos más cortos de un vértice, al que denominamos origen, a los restantes vértices del grafo. Es conocido como un algoritmo de tipo *Greedy*, que a su vez se le conoce como algoritmos ávidos, glotones o voraces. Estos algoritmos trabajan por etapas, tomando en cada una de ellas la solución mejor (óptimo local) sin considerar las consecuencias futuras. El óptimo encontrado en una etapa puede posteriormente ser modificado si surge una mejor solución.

Con el uso de este algoritmo se pueden garantizar excelentes resultados. Dentro de estos resultados se encuentran la disminución sensible del riesgo de que existan errores, la racionalización del esfuerzo intelectual y las habilidades y conocimientos al propiciar que se consideren caminos y rutas que de otra manera pudieran pasar desapercibidos; entre otros resultados. No obstante, su uso también puede provocar la lentitud en la repercusión del proceso recursivo y la dificultad de aplicar el método a situaciones en las que la solución al problema general no se deriva de la suma directa y simple de una parte del problema.

A continuación se muestra el **pseudocódigo** del Algoritmo: [15]

⁷ *Edsger Wybe Dijkstra* nació en Rotterdam, (Holanda) en 1930. Su padre era químico y su madre matemática. En 1972 ganó el Premio Turing ACM, y, en 1974, el *AFIPS Harry Good Memorial*. *Dijkstra* se trasladó a Austin, Texas a principio de los 80. En 1984, se le ofreció un puesto en Ciencias de la Computación en la Universidad de Texas, donde permaneció desde entonces hasta que recientemente el 6 Agosto del 2002, falleció en su hogar de *Nuenen, (Netherlands)*.

Dijkstra (G,s)

Inicializar

for cada v perteneciente a $V[G]$

do $d[v] = \text{infinito}$

$p[v] = \text{nulo}$

$d[s] = 0$

$S = \text{vacío}$

$Q = V[G]$

mientras Q no vacío

do $u = \text{nodo } v \text{ con min } d[v]$

$S = S \text{ unión } u$ 'se añade al conjunto de nodos finalizados

for cada v perteneciente Adyacente u

 Relajación

if $d[v] > d[u] + w(u,v)$ **then**

$d[v] = d[u] + w(u,v)$

$p(v) = u$

fin

Teorema: Para una entrada consistente en un grafo simple, conexo, con pesos y n vértices, el algoritmo de Dijkstra tiene un tiempo de ejecución $O(n^2)$ en el peor de los casos.

2.2.2 Algoritmo de Floyd-Warshall

Este algoritmo, se encuentra enmarcado a la solución de problemas de los caminos más cortos entre todos los pares de vértices del grafo. El nombre lo debe a su creador, un profesor de la Universidad de Stanford llamado *Robert W. Floyd*⁸.

El problema que intenta resolver este algoritmo es el de encontrar el camino más corto entre todos los pares de nodos o vértices de un grafo. Esto es semejante a construir una tabla con todas las distancias mínimas entre pares de ciudades de un mapa, indicando además la ruta a seguir para ir de la primera ciudad a la segunda. Este es uno de los problemas más interesantes que se pueden resolver con algoritmos de grafos. [16]

Es decir, su procedimiento es bastante similar a construir una tabla con todas las distancias mínimas entre pares de ciudades en un mapa. Es un algoritmo que cuenta con un bucle que examina cada vértice y lo toma como pivote.

A continuación se muestra el **pseudocódigo** del Algoritmo: [16]

Floyd-Warshall (G)

Inicializar

$D = A$ ' matriz de distancias = matriz de arcos

si $i=j$ o $D_{ij} = \infty$ entonces $P_{i,j} = \text{nulo}$ sino $P_{i,j} = i$ 'matriz de caminos

for $k = 1$ to V

for $i = 1$ to V

⁸ **Robert W. Floyd** nació el 8 de junio de 1936 en *Nueva York*, es profesor de la *Stanford University* (B.A. Chicago 1955 B.S. Chicago 1958), y en 1978 fue galardonado con el prestigioso premio A.M. Turing que otorga la ACM para reconocer las contribuciones de naturaleza técnica realizadas a la comunidad informática.

```
for j = 1 to V
  Di,j = min(Di,j , Di,k + Dk,j )
  si min = Di,k + Dk,j entonces
    Pi,j = Pk,j
fin
```

Este tipo de algoritmo puede emplearse a diferentes problemas, incluyendo el diseño de circuitos, el diseño de rutas de transporte, aproximaciones al problema del viajante de comercio, o como base de otros algoritmos más complejos.

Teorema: La complejidad temporal es $O(n^3)$, pues la función Floyd presenta un triple bucle “for” anidado, dentro del cual se realizan operaciones sencillas de asignaciones y sumas.

2.2.3 Algoritmo de Prim

Para el año 1957 *Robert Prim*⁹, descubre un algoritmo para la resolución del problema del Árbol de coste total mínimo, no obstante un año atrás en 1956 *Joseph B. Kruskal*, había resuelto este tipo de problema. El algoritmo de Prim tal vez es, el algoritmo de *Minimum Spanning Tree (MST)* o Árbol de Expansión Mínima, más sencillo de implementar. Este algoritmo puede encontrar el MST de cualquier grafo conexo ponderado.

Consiste en dividir los nodos de un grafo en dos conjuntos: procesados y no procesados. Al principio, hay un nodo en el conjunto procesado que corresponde al equipo central; en cada interacción se incrementa el grafo de procesados en un nodo (cuyo arco de conexión es mínimo) hasta llegar a establecer la conexión de todos los nodos del grafo a procesar. [17]

En otras palabras el algoritmo encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo,

⁹ **Robert Clay Prim** nació en 1921 *Sweetwater*, Texas, EEUU matemático y científico de la computación. Trabajó en la Universidad de Princeton desde 1948 hasta 1949 como investigador asociado. Trabajó además en los laboratorios Bell.

entonces el algoritmo encontrará el árbol de recubrimiento mínimo para cada uno de los componentes conexos que forman dicho grafo no conexo.

A continuación se muestra el **pseudocódigo** del Algoritmo: [17]

Prim (L [1..n , 1..n]) : 'conjunto de arcos

Inicialización: sólo el nodo 1 se encuentra en B

T =NULL 'T contendrá los arcos del árbol de extensión mínima Distmin[1]=-1

para i=2 hasta n **hacer**

 más_próximo [i]=1

 distmin [i]=L [i , 1]

para i=1 hasta n -1 **hacer**

 min=infinito

para j=2 hasta n **hacer**

 si $0 \leq \text{distmin} [j] < \text{min}$ **entonces**

 min=distmin [j]

 k=j

 T=T union {{mas_próximo [k], k }}

 distmin [k]= -1 'se añade k a B

para j=2 hasta n **hacer**

si $L [j , k] < \text{distmin} [j]$ entonces

$\text{distmin} [j] = L [j , k]$

más_próximo [j] = k

devolver T

Teorema: El algoritmo de Prim tiene un tiempo de ejecución $O(n \log n)$ en el peor de los casos.

2.2.5 Algoritmo de Ford-Fulkerson

Su calificativo esta dado por sus creadores, *L. R. Ford*¹⁰ y *Delbert Ray Fulkerson*¹¹. El objetivo de este algoritmo es encontrar caminos en los que se vaya incrementando el flujo, hasta lograr el flujo máximo. Su concepción principal es hallar la ruta de penetración con flujo positivo neto que una el nodo origen y el nodo destino. Este algoritmo es aplicable a los Flujos Maximales.

Sea (V, A, w) con V vértices, A aristas y w peso de las aristas, una red con una única fuente s y un único sumidero t ; $w(\alpha)$ es la capacidad de α perteneciente a la arista A . Un flujo f es viable si $f(\alpha) \leq w(\alpha)$ para todo α perteneciente a la arista A . Se trata de hallar un flujo viable con el valor máximo posible.

En una red con fuente j y sumidero k único el valor máximo que puede tomar un flujo variable es igual a la capacidad mínima que puede tomar un corte.

A continuación se muestra el **pseudocódigo** del algoritmo:

Ford-Fulkerson (G, j, k)

{

¹⁰ **Lester Randolph Ford, Sr.** (1886-1967). Matemático estadounidense. Su trabajo se reconoce por el establecimiento de los premios Lester R. Ford.

¹¹ **D. R. Fulkerson:** (1924-1976). Matemático que contribuyó a la creación del algoritmo Ford-Fulkerson.

for (cada arco (u, v) de E)

{

$f[u, v]= 0;$

$f[v, u]= 0;$

}

while (exista un camino p desde j a k en la red residual G_f)

{

$cf(p) = \min\{ cf(u, v): (u, v) \text{ está sobre } p\};$

for (cada arco (u, v) en p)

{

$f[u, v]= f[u, v] + cf(p);$

$f[v, u]= - f[u, v]; \quad \}$

}

}

Este algoritmo depende de tres conceptos principales: [18]

- Un **camino de aumento**, es una trayectoria desde el nodo fuente s al nodo sumidero t que puede conducir más flujo.
- La **capacidad residual** es la capacidad adicional de flujo que un arco puede llevar $cf(u,v) = c(u,v) - f(u,v)$

- **Teorema de Ford-Fulkerson (1962):** En cualquier red, el flujo máximo que fluye de la fuente al destino es igual a la capacidad del corte mínimo que separa a la fuente del destino.

En otras palabras el algoritmo Ford-Fulkerson, es un algoritmo iterativo, que comienza con $f(u,v)=0$ para cada par de nodos y en cada iteración se incrementa el valor del flujo buscando un camino de aumento. Este proceso se va a repetir hasta no encontrar un camino de aumento.

2.2.6 Algoritmo de Kruskal

El creador de este algoritmo como su nombre lo indica es Joseph B. Kruskal¹². Para el año 1956 ya había descubierto dicho algoritmo. Este algoritmo forma un árbol que incluye todos los vértices donde las aristas poseen menor peso. En el caso de que el grafo no sea conexo, entonces busca un bosque expandido mínimo.

Este algoritmo es de tipo *greedy*, ya que a cada paso, éste selecciona el arco más barato y lo añade al sub-grafo. Este tipo de algoritmos pueden no funcionar para resolver otro tipo de problemas, por ejemplo para encontrar la ruta más corta entre los nodos a y b. [19]

Entonces, se puede decir, que este algoritmo trabaja, creando dos conjuntos, uno con los árboles, formando un bosque, en el que los vértices del grafo constituyen un árbol separado, y otro que tenga a todas las aristas del grafo. Mientras el conjunto que contiene las aristas no es vacío, se van eliminando las aristas de peso mínimo de dicho conjunto y en el caso que esa arista conecta dos árboles diferentes se añade al bosque uniendo los dos árboles en un solo árbol y en caso contrario se desecha la arista. Como resultado final de este algoritmo se tiene un bosque con un solo componente, formando así un árbol de expansión mínimo en el grafo.

Por lo expuesto hasta aquí se puede apreciar que el principal objetivo del algoritmo de Kruskal es generar disimiles bosques, en los que se encuentran las distintas aristas que va seleccionando en su ejecución hasta formar el árbol de expansión mínima.

¹² **Joseph Bernard Kruskal, Jr.:** (29 de enero 1928). Matemático, estadístico e informático q ha colaborado en disimiles proyectos. Estudió en la Universidad de Chicago y la Universidad de Princeton, donde completó su doctorado en 1954.

A continuación se muestra el **pseudocódigo** del Algoritmo: [19]

Kruskal (G)

$E(1)=0$, $E(2)=$ todos los Arcos del grafo G

Mientras $E(1)$ contenga menos de $n-1$ arcos y $E(2) \neq 0$ **do**

De los arcos de $E(2)$ seleccionar el de menor coste $\rightarrow e(ij)$

$E(2) = E(2) - \{e(ij)\}$

Si $V(i)$, $V(j)$ no están en el mismo árbol entonces

juntar los árboles de $V(i)$ y de $V(j)$ en uno sólo

end Si

end do

Fin del algoritmo

Teorema: Debido a que las aristas se ordenan por su peso y después se eliminan las de menor peso en un tiempo de ejecución constante, este algoritmo posee complejidad $O(n \log n)$.

2.2.7 Algoritmo de Bellman-Ford

Dentro de los algoritmos del camino más corto también se encuentra el Bellman-Ford: versiones distribuidas y centralizadas. Personalidades como *Richard Bellman*¹³, *Samuel End*¹⁴ y *Lester Ford*,

¹³ **Richard Bellman:** (1920-1984). Su trabajo estuvo dedicado al estudio de las matemáticas y a la teoría de control. Célebre por su aporte en la programación dinámica en 1953. Hizo contribuciones importantes en muchos campos dentro de la Matemática.

¹⁴ **Samuel End:** Científico matemático que estuvo vinculado al estudio de algoritmos que facilitarían el cálculo de rutas óptimas para llegar de un punto a otro.

desarrollaron exitosamente este algoritmo. Para el año 1970, se mejora este algoritmo y es utilizado en grafos sin ciclos.

La eficiencia del mismo, radica en que se ejecute en un grafo dirigido ponderado, en el que algunas de sus aristas pueden llegar a tener peso negativo, a diferencia del algoritmo Dijkstra que no puede tener aristas con peso negativo. A pesar de que Bellman-Ford es bastante parecido al algoritmo Dijkstra, utiliza más cantidad de tiempo para resolver el mismo problema y se usa generalmente cuando existen arcos con peso negativo.

Soluciona el problema de la ruta más corta o camino mínimo desde un nodo origen, de un modo más general que el Algoritmo de Dijkstra, ya que permite valores negativos en los arcos. El algoritmo devuelve un valor booleano si encuentra un circuito o lazo de peso negativo. En caso contrario calcula y devuelve el camino mínimo con su coste. [20]

Este algoritmo posee dos variantes, las cuales le posibilita que sea usado para cuando existen arcos con pesos negativos. Una de estas variantes es la versión no optimizada utilizada en grafos con ciclos negativos. La otra versión es la optimizada, utilizada también en grafos, a diferencia de que en este grafo no existan ciclos de coste negativo.

Existen variantes del algoritmo que son utilizadas en el Protocolo de encaminamiento de información (RIP), el cual es de gran importancia en las redes, dispositivos router IP y en otros que permiten la comunicación a diario con los servicios de internet.

A continuación se muestra el **pseudocódigo** del Algoritmo: [20]

Bellman-Ford (G,s)

Inicializar

for cada v perteneciente a $V[G]$

do $d[v] = \text{infinito}$

$p[v] = \text{nulo}$

$p[s] = 0$

for $i=1$ to $V[G]-1$

do for cada arco (u,v) perteneciente a $A[G]$

Relajación

if $d[v] > d[u] + w(u,v)$ **then**

$d[v] = d[u] + w(u,v)$

$p(v) = u$

for cada arco (u,v) chequea lazo de peso negativo

do if $d[v] > d[u] + w(u,v)$ **then**

return FALSO 'el algoritmo no converge

return VERDADERO

2.2.8 Algoritmo de búsqueda en anchura (BFS)

Los algoritmos de búsqueda, generalmente se aplican a árboles. El algoritmo de Búsqueda en anchura o BFS (*Breadth First Search*, Amplitud en Primera Consulta), como lo indican sus siglas, transita de forma uniforme por todos los nodos de un árbol, expandiendo cada uno de los nodos de un nivel antes de pasar al próximo. Este algoritmo no utiliza para su desarrollo ninguna destreza heurística.

Generalmente, si estamos en presencia de un grafo, se comienza por el primer vértice o sea la raíz, seleccionándolo como elemento raíz y se prosigue a explorar todos los nodos adyacentes a dicho elemento raíz, que por consiguiente es un nodo también. Después se prosigue a realizar la misma

operación con cada uno de los nodos restantes, de forma tal que se recorra todo el árbol. Este recorrido siempre es de forma creciente, teniendo en cuenta el índice de los nodos.

Entre las ventajas que posee este algoritmo, es que debido a que su recorrido lo hace de forma uniforme, garantiza así que se pueda hallar la solución más óptima de forma rápida y eficaz. Su alto grado de complejidad computacional hacen que el nivel del árbol y la cantidad de hijos de un nodo aumenten velozmente los requerimientos y se vuelvan inaceptables, convirtiéndolo así en la desventaja primordial que tenga este algoritmo.

A continuación se muestra el **pseudocódigo** del Algoritmo:

```
halla_objetivo_anchura (A: espacio_de_búsquedas)
```

```
{ /* dev objetivo*/
```

```
Cola lista_nodos;
```

```
Boolean acabar = false;
```

```
lista_nodos.encolar (A.nodo_raiz);
```

```
while ((!lista_nodos.esVacia()) AND (!acabar)) do
```

```
{
```

```
    nodo_actual = lista_nodos.primer();
```

```
    lista_nodos.quitaPrimer();
```

```
    if (esObjetivo (nodo_actual) ) then
```

```
        { acabar = true; }
```

```
    else {
```

```
nodo_actual.expandir(); //se añaden los hijos del nodo actual

    lista_nodos.encolarMuchos (nodos_expandidos_de_nodo_actual); }

}

return nodo_actual;

}
```

2.2.9 Algoritmo de búsqueda en profundidad (DFS)

Este algoritmo es bastante parecido al BFS, sus siglas en inglés significan *Depth First Search*, o sea Profundidad en Primera Consulta. A diferencia de que cuando visita todos los nodos de un árbol lo hace de forma ordenada, pero no uniforme. Su función es comenzar siempre por el primer nodo hijo que no haya sido visitado e ir expandiendo cada uno de los nodos a medida que los encuentra. Cuando todos los hijos de ese nodo han sido visitados, regresa otra vez y el padre del nodo pasa a ser el nodo actual y se le va aplicando el mismo procedimiento, evitando siempre la expansión indefinida de los nodos.

Este algoritmo no siempre encuentra la solución más óptima, a diferencia del BFS, que siempre encuentra la solución. Tiene como ventaja que puede expandir los caminos hasta su máxima profundidad y a lo largo del proceso de ejecución puede eliminar nodos que le permiten reducir el recorrido de ramas en el árbol. Como desventaja, tiene que posee un requerimiento de memoria limitado.

A continuación se muestra el **pseudocódigo** del Algoritmo:

```
halla_objetivo_profundidad (A: espacio_de_búsquedas)
```

```
{ /* dev objetivo*/
```

```
    Pila lista_nodos;
```

```
    Boolean acabar = false;
```

```
lista_nodos.apilar (A.nodo_raiz);

while ((!lista_nodos.esVacia()) AND (!acabar)) do

{

    nodo_actual = lista_nodos.cima();

    lista_nodos.desapila();

    if (esObjetivo (nodo_actual) ) then

        { acabar = true; }

    else

        {   nodo_actual.expandir(); //se añaden los hijos del nodo actual

            lista_nodos.apilarMuchos (nodos_expandidos_de_nodo_actual);

        }

}

return nodo_actual;

}
```

2.2.10 Problema del trasbordo

El problema del trasbordo es un caso particular de los problemas de Flujos. Su objetivo es enviar un bien desde unos puntos origen a unos puntos destino pero pudiendo pasar por puntos intermedios. [21]

En ocasiones el envío de mercancías mediante la utilización de puntos intermedios resulta más viable que realizarlo directamente desde el origen hasta el destino.

Los vértices del grafo pueden ser de varios tipos: [21]

- **Orígenes puros:** Solo pueden enviar bienes. De ellos solamente pueden salir arcos.
- **Destinos puros:** Solo pueden recibir bienes. A ellos solamente pueden llegar arcos.
- **Transbordos:** Pueden enviar y/o recibir mercancías. A ellos pueden llegar arcos y/o de ellos pueden salir arcos.

Una red de flujo se puede representar usando un grafo dirigido, donde los vértices abastecedores originan una determinada cuantía de material que es empleada por los vértices clientes. Por lo que un arco puede pensarse como el canal que tiene un volumen máximo o mínimo de flujo y un valor de transporte por unidad de material.

Ahora bien, dado un conjunto de vértices orígenes X , que posean capacidad de suministro $b < 0$ y otro conjunto de vértices destinos Y con demanda $b > 0$, donde además existen nodos de transbordo, los cuales solamente realizan un transbordo ya que no tributan ni gastan las mercancías.

Se persigue además, que los nodos sean continuos, es decir, la suma de flujos que entran a un nodo restado a la suma de los flujos que salen, tiene que ser mayor o igual que la capacidad a del nodo. Por tanto el modelado del problema del Transbordo se muestra en la siguiente figura: [21]

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

sujeto a :

$$\sum_{k=1}^m x_{ki} - \sum_{j=1}^m x_{ij} = b_i, j = 1 \dots m \begin{cases} b_i > 0, \text{demanda} \\ b_i < 0, \text{oferta} \\ b_i = 0, \text{transbordo} \end{cases}$$

$$x_{ij} \geq 0, x_{ij} \in Z$$

Fig. 3 Modelo del Problema del Transbordo

Entonces, el problema en cuestión se expresa en la siguiente interrogante: ¿cuál es el flujo de transporte de cada arco con el cual se puede transportar el material desde los nodos proveedores a los nodos cliente, sin violar las restricciones de capacidad y continuidad y todo ello minimizando el coste total de transporte?. [22]

2.2.11 Problema del Viajante de Comercio

En la actualidad existen múltiples problemas relacionados con el cálculo de rutas mínimas. Uno de ellos es el problema del viajante (TSP¹⁵), el cual se encuentra dentro del campo de la combinatoria computacional. Su planteamiento en ocasiones, resulta simple, pero la práctica evidencia que suele ser uno de los más complicados a la hora de encontrar la solución. Esta complicación se debe a que requiere de un sinnúmero de recursos computacionales para llegar a un resultado final.

La representación de este problema se logra mediante un grafo dirigido completo $G = (V, A)$, donde V es el conjunto de vértices y A el conjunto de aristas, en el que existe una conexión entre cualquier par de

¹⁵ TSP: Acrónimo de *Traveling Salesman Problem*.

vértice. Estos vértices constituyen ciudades y las aristas representan las distancias que existen entre cada una de ellas, es por ello que las aristas tienen que ser obligatoriamente positivas.

Formulando el problema matemáticamente, se dice que el objetivo del mismo consiste en encontrar el camino C formado por la secuencia de $n+1$ ciudades, $C = \{1, 2, \dots, n, n+1\}$, donde $\forall i \in N, i \in C$, como se muestra en la siguiente figura: [22]

$$\text{Minimizar } \sum_{i=1}^n A[C_i, C_{i+1}] \text{ tal que } C_1 = C_{n+1} = 0$$

Fig. 4 Modelo del Problema del Viajante de Comercio

Por último y no menos importante, es bueno mencionar el problema de los m -viajantes del comercio. Este problema se trata de m vendedores o viajantes de comercio que debe visitar n ciudades para vender u ofertar sus productos. Cada viajante de comercio podrá visitar como máximo a p clientes o ciudades. Cada par de ciudades puede estar comunicado o no, su distancia se define mediante C_{ij} . El problema es por tanto, decidir el recorrido que comenzando por una determinada ciudad $\{0\}$ pase por todas las demás una sola vez y vuelva finalmente a la primera, de manera que se minimice la distancia total recorrida de todas las rutas. [22]

2.2.12 Problema de Rutas para Vehículos Capacitados

A grandes rasgos un problema de ruteo de vehículos consiste en, dado un conjunto de clientes y depósitos dispersos geográficamente y una flota de vehículos, determinar un conjunto de rutas de costo mínimo que comiencen y terminen en los depósitos, para que los vehículos visiten a los clientes máximo una vez. [23] Dentro de estos problemas dedicados al ruteo se encuentra el Problema de Rutas para Vehículos Capacitados (CVRP), o como sus siglas en inglés VRP^{16} . Figuras como *Dantzig y Ramser*¹⁷,

¹⁶ **VRP**: Acrónimo de Vehicle Routing Problem.

¹⁷ **Dantzig y Ramser**: G. B. Dantzig, J. H. Ramser ambos matemáticos. Dedicados al estudio de la combinatoria computacional, implantaron en 1959 el CVRP bajo el nombre original de "The truck dispatching problem".

*Clarke y Wright*¹⁸ hicieron posible que el CVRP, alcanzara en nuestros días un avance y desarrollo bastante notorio. El objetivo este tipo de problemas es establecer un conjunto de rutas dado una flota que puedan proporcionar asistencia a un grupo de clientes. Posee características que lo hacen importante y junto con el Problema del Viajante, pertenece al conjunto de problemas de optimización combinatoria.

El CVRP presenta dentro de sus funcionalidades la resolución de los problemas de optimización del cálculo del número total de vehículos que se necesitan para brindar apoyo o ayuda a los clientes así como también minimizar los costes fijos relacionados con los vehículos. Otra de las funcionalidades que brinda este problema es lograr que las rutas se encuentren de forma equitativa teniendo en cuenta factores como el tiempo de viaje o la carga que posee el vehículo.

El Problema CVRP básico trata de determinar los recorridos de cierta cantidad de vehículos de capacidad que partiendo de un origen común deben pasar por un conjunto de lugares de interés (clientes) para recoger o distribuir mercancías según una demanda, y volver de nuevo al origen de manera que la distancia total recorrida (el coste o el tiempo empleado) por el conjunto de vehículos sea mínima. En el tipo de problema más sencillo no se tiene en cuenta el horario de entrega o recogida en cada lugar de interés (ventanas horarias). [24]

Se puede decir entonces que este tipo de problemas constituye una de las actividades más difíciles y engorrosas dentro del campo de la optimización combinatoria. Este problema que ha sido definido hace más de 40 años radica en encontrar el conjunto de rutas para asistir a un determinado grupo de clientes. A pesar de ello su resolución posee una dificultad considerable. Se tiene un grafo, en el cual existe un conjunto de nodos representados por las variables i, j . Estas variables deberán minimizar el coste total de los arcos que han sido visitados para hallar la solución. Se tienen dos variables binarias X_{ijk} y Y_{ik} .

En la primera variable se muestra si el arco ij es recorrido por el vehículo k . En la segunda variable se muestra si el nodo i que tiene demanda d_i será atendido por el vehículo k que tiene capacidad C_k . Es por

¹⁸ **Clarke y Wright:** G. Clarke, J. W. Wright figuras que propusieron el primer algoritmo que resultó efectivo para resolver el CVRP.

ello que cada nodo será atendido solamente por un vehículo. La fig. 5 muestra una explicación más detallada de lo que sería el modelo matemático del CVRP.

$$\begin{aligned}
 & \min \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \\
 & \text{s.a.} \\
 & \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V \setminus \{0\} \\
 & \sum_{k=1}^K y_{0k} = K \\
 & \sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \quad \forall i \in V, k = 1 \dots K \\
 & \sum_{i \in V} d_i y_{ik} \leq c_k \quad \forall k = 1 \dots K \\
 & \sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk} \quad \forall S \subseteq V \setminus \{0\}, h \in S, k = 1 \dots K \\
 & x_{ijk} \in \{0,1\} \quad \forall i, j \in V, k = 1 \dots K \\
 & y_{ik} \in \{0,1\} \quad \forall i \in V, k = 1 \dots K
 \end{aligned}$$

Fig. 5 Modelo matemático de tres subíndices para el problema básico CVRP [23]

2.3 Algoritmos heurísticos: Algoritmo A*

La palabra heurística se encuentra estrechamente relacionada con la resolución de problemas matemáticos en un determinado tiempo. Fundamentalmente este término se utiliza en Inteligencia Artificial, rama ésta, dentro de las Matemáticas. Este tipo de algoritmos, brindan una solución como

todos los demás. A pesar de ser una excelente solución, no se puede demostrar que es la solución óptima para determinado problema.

Los algoritmos heurísticos son aquellos que cuando la solución no se determina de forma directa, se realizan una serie de ensayos, pruebas y reensayos. El método que ellos usan consiste en generar candidatos de soluciones posibles de acuerdo a un patrón dado; luego los candidatos son sometidos a pruebas de acuerdo a un criterio que caracteriza a la solución. Si un candidato no es aceptado, se genera otro; y los pasos dados con el candidato anterior no se consideran. [25]

El siguiente esquema trata de dar unas categorías amplias, no excluyentes, en donde ubicar a los métodos heurísticos más conocidos: [26]

- **Métodos de Descomposición:** El problema original se descompone en sub problemas más sencillos de resolver, teniendo en cuenta, aunque sea de manera general, que éstos pertenecen al mismo problema.
- **Métodos Inductivos:** La idea de estos métodos es generalizar de versiones pequeñas o más sencillas al caso completo.
- **Métodos de Reducción:** Consiste en identificar propiedades que se cumplen mayoritariamente en las buenas soluciones e introducirlas como restricciones del problema. El objeto es restringir el espacio de soluciones simplificando el problema. El riesgo obvio es dejar fuera las soluciones óptimas del problema original.
- **Métodos Constructivos:** Consisten en construir literalmente paso a paso una solución del problema. Usualmente son métodos deterministas y suelen estar basados en la mejor elección en cada iteración. Estos métodos han sido muy utilizados en problemas clásicos como el del agente viajero.
- **Métodos de Búsqueda Local:** A diferencia de los métodos anteriores, los procedimientos de búsqueda o mejora local, comienzan con una solución del problema y la mejoran progresivamente. El método finaliza cuando no existe ninguna solución accesible que mejore la anterior.

Este tipo de algoritmo se caracteriza por tener en cuenta la rapidez de resolución del algoritmo, así como la calidad que debe tener la solución. Es imprescindible utilizar los algoritmos heurísticos, en dos casos:

uno cuando se necesita obtener un buen resultado final y otro cuando se ha encontrado la solución y se requiere perfeccionarla. Dentro de los algoritmos heurísticos se encuentra el A*, algoritmo que es usado generalmente en la teoría de grafos.

El algoritmo de búsqueda A* se encuentra en el grupo de los algoritmos de búsqueda en grafos. Un algoritmo de búsqueda, no es más, que el algoritmo que dado una estructura de datos, permite encontrar un elemento dentro de la misma, y además decir, si el elemento pertenece o no a dicho conjunto.

Figuras como *Peter E. Hart*¹⁹, *Nils J. Nilsson*²⁰ y *Bertram Raphael*²¹ lo presentaron en el año 1998. El algoritmo de búsqueda A*, localiza el camino de menor coste entre dos nodos, es decir, entre el nodo origen y el nodo destino.

Este algoritmo comienza, a partir de la creación de una lista con el nodo raíz. Mientras que dicha lista no esté vacía y no se alcance la meta y el primer elemento de la lista es la meta, se expanden todos los nodos que tengan menor costo que el nodo meta; de lo contrario se elimina el nodo que se encuentra en la primera posición de la lista, luego se agregan sus hijos a la lista sumándole las funciones $g^{22} + h^{23}$ y se ordenan los elementos de acuerdo al costo y se eliminan los que estén repetidos de mayor costo. Por último se expanden todos los nodos que posean menor costo que el nodo meta.

Este algoritmo como muchos algoritmos de búsquedas posee un tiempo de ejecución exponencial.

A continuación se muestra el **pseudocódigo** del Algoritmo: [27]

¹⁹ **Peter E. Hart:** (1940). Científico de la computación. Creador y presidente de *Ricoh Innovations* en 1997.

²⁰ **Nils J. Nilsson:** *Kumagai* Profesor de Ingeniería del Departamento de Ciencias de la Computación en la Universidad de *Stanford*, recibe su doctorado en Ingeniería Eléctrica de *Stanford* en 1958. Fue presidente de dicho Departamento hasta agosto de 1990.

²¹ **Bertram Raphael:** (nacido en 1936 en *Nueva York*) es un científico de la computación estadounidense, conocido por sus contribuciones a la inteligencia artificial.

²² **g:** Es una medida del costo de llegar desde el nodo inicial al nodo actual

²³ **h:** Es una estimación del costo adicional para llegar desde el nodo actual al estado objetivo.

Algoritmo A*

Est_abiertos.insertar(EstadoInicial)

Actual= Est_abiertos.primer() ()

mientras no es_final?(Actual) **y no** Est_abiertos.vacia? () **hacer**

Est_abiertos.borrar_primer() ()

Est_cerrados.insertar(Actual)

hijos= generar_sucesores(Actual)

hijos= tratar_repetidos(Hijos, Est_cerrados , Est_abiertos)

Est_abiertos.insertar (Hijos)

Actual= Est_abiertos.primer() ()

fmientras

fAlgoritmo

El algoritmo A* facilita los problemas en los que encontrar la mejor solución es el objetivo fundamental. Su coste en espacio y tiempo en el caso medio es mejor que los algoritmos de búsquedas a ciegas en el caso de que la heurística sea la adecuada. A pesar de ello, el algoritmo A* en ocasiones no puede dar respuesta a algunos problemas de búsquedas.

2.4 Algoritmos Meta-heurísticos: Búsqueda Tabú

La matemática ha ido desarrollando una serie de métodos y procedimientos entre los que se destaca la meta-heurística. Este procedimiento es utilizado en la resolución de problemas de optimización combinatoria, específicamente en el caso que estos sean aún más complejos y no se les pueda dar

solución a través de los métodos heurísticos. La búsqueda tabú es un ejemplo de algoritmos meta-heurísticos que brinda la posibilidad de dar solución a problemas de optimización combinatoria, como es el caso del problema del viajante, de los problemas de asignación, por solo mencionar algunos ejemplos.

Búsqueda tabú o Tabu search (en inglés), tiene sus inicios por los años setenta cuando Fred Glover²⁴ plantea sus primera ideas. Según el diccionario Webster, la palabra tabú significa "una prohibición impuesta por factores sociales la costumbre como una medida de protección "o de algo" prohibido con riesgos. Una acepción más moderna la define como "Una prohibición impuesta por costumbres sociales como una medida de protección". [28]

En muchas ocasiones para resolver los problemas de optimización, se emplean un sinnúmero de métodos de búsquedas, los cuales llega el momento en que el algoritmo comienza a hacer ciclos o itera por un tiempo indefinido. Esta restricción hace que se empiece a utilizar la búsqueda tabú. Esta búsqueda posee una lista tabú²⁵ y unos elementos de selección, que hacen que marque la diferencia con las demás meta-heurísticas.

La búsqueda tabú permite moverse a una solución aunque no sea tan buena como la actual, de modo que se pueda escapar de óptimos locales y continuar estratégicamente la búsqueda de soluciones aún mejores. [29] La búsqueda tabú hace uso de las estructuras de memoria, las cuales las clasifica en dos tipos. Un tipo es la estructura de memoria explícita, la cual proporciona un óptimo local. El otro tipo de estructura es la de atributos, que su principal objetivo es orientar la búsqueda que se está realizando.

Dentro de las búsquedas tabús existen otros conceptos fundamentales para poder llevar a cabo la realización del mismo. Ejemplo de ello es entorno o vecindario, que es el conjunto que almacena las posibles soluciones asociadas. Otro ejemplo lo constituyen los criterios de aspiración, estos criterios admiten que un determinado movimiento en el algoritmo sea o no válido a pesar de estar catalogado como tabú.

²⁴ **Fred Glover:** Nació el 8 de Marzo de 1937; Ciudad de, *Kansas Missouri*. Ha trabajado como director en numerosas empresas, ejemplo: Director de Desarrollo Tecnológico, Robótica Management, Inc., 1981 – 1992, Director de Investigación, Centro para la Ciencia Hearin Empresa, desde 2000 hasta 2002, entre otros. Es profesor de Universidad de Colorado (USA).

²⁵ **Lista tabú:** Lista que va a almacenar todos los elementos o posibles soluciones que no deben ser el resultado final.

Muchas son las aplicaciones que la búsqueda tabú tiene. Esto se debe a que muchos de los problemas de la vida práctica que no pueden ser resueltos a través de los métodos analíticos se resuelven a través de heurísticas y meta-heurísticas. El problema del viajante (visto en epígrafes anteriores), constituye uno de los problemas de optimización que resuelve la búsqueda tabú, para ello se basa en el algoritmo del vecino más cercano. La búsqueda tabú va a llegar a su fin cuando recorridas todas las ciudades se encuentre la solución que tenga menor distancia recorrida.

2.5 Algoritmos genéticos

Muchos de los algoritmos vistos anteriormente, son utilizados por las grandes compañías que se encuentran dedicadas al estudio de optimización de rutas. En ocasiones estos algoritmos no actúan de forma eficiente. Esto se debe a que las soluciones de los problemas presentados requieren de algoritmos y métodos más complejos para llegar a una solución final. Es por ello que surgen los algoritmos genéticos, los cuales poseen un mayor grado de complejidad y brindan la posibilidad de resolver este tipo de problemas.

Dentro de los métodos de búsqueda dirigida se encuentran los llamados Algoritmos Genéticos (AG). Su surgimiento data desde la década de los años sesenta cuando *John. H. Holland* ²⁶ introduce esta técnica de programación que en sus primeros inicios se llamó Planes Reproductivos. Estos algoritmos están relacionados con la computación y la genética.

Existen diferentes terminologías acerca de los algoritmos genéticos. Según John Holland los Algoritmos Genéticos son algoritmos matemáticos de optimización de propósito general basados en mecanismos naturales de selección y genética, proporcionando excelentes soluciones en problemas complejos con gran número de parámetros. [30]

Según ACM los AG son una herramienta muy poderosa de optimización que puede ser usada para resolver un gran número de problemas difíciles con gran eficiencia y exactitud, basándose en la genética natural y la teoría de la evolución de Darwin. Por último, Gastón Crevillén y David Díaz menciona que los

²⁶ **John. H. Holland:** Profesor estadounidense de la Universidad de Michigan que introduce la idea de los Algoritmos Genéticos, y que años más tarde es desarrollada por otras personalidades.

algoritmos genéticos (AG) proporcionan un método de aprendizaje basado en la analogía con la evolución de las especies. [30]

Entonces, se puede concluir que los algoritmos genéticos, constituyen instrumentos importantes dentro de la Inteligencia Artificial. Estos son parte de la computación evolutiva. Su desarrollo se encuentra estrechamente relacionado con la reproducción, mutación, competición y selección que puedan tener lugar en una determinada población. Estos algoritmos se basan en la probabilidad²⁷, es decir que a medida que las iteraciones aumenten en el desarrollo del algoritmo, la probabilidad de obtener la solución óptima tiende a 1.

Un algoritmo genético tiene también una serie de parámetros que se tienen que fijar para cada ejecución, como los siguientes: [31]

- **Tamaño de la población:** Debe de ser suficiente para garantizar la diversidad de las soluciones, y, además, tiene que crecer más o menos con el número de bits del cromosoma, aunque nadie ha aclarado cómo tiene que hacerlo. Por supuesto, depende también del ordenador en el que se esté ejecutando.
- **Condición de terminación:** Lo más habitual es que la condición de terminación sea la convergencia del algoritmo genético o un número prefijado de generaciones.

Los algoritmos genéticos son muy conocidos por sus excelentes resultados. Es por ello que pueden ser utilizados en cualquier función matemática, no importa cuán compleja pueda ser su solución. Este tipo de algoritmo utiliza una serie de operadores genéticos, los cuales hacen posible la diversidad genética en una determinada población. La selección, el cruzamiento y la mutación, son ejemplos de este tipo de operadores genéticos. A continuación se hace una breve explicación de cada uno de ellos:

- **Selección:** Su principal objetivo es elegir los miembros de la población que más tarde serán utilizados para el cruzamiento. Los miembros que mejores se encuentren tiene

²⁷ **Probabilidad:** Medida entre 0 y 1 que se utiliza en diferentes ciencias y que determina la periodicidad con que se llega a un resultado al realizar fenómenos potenciales.

mayor posibilidad de quedar seleccionados. La siguiente figura muestra un ejemplo de operador de selección.

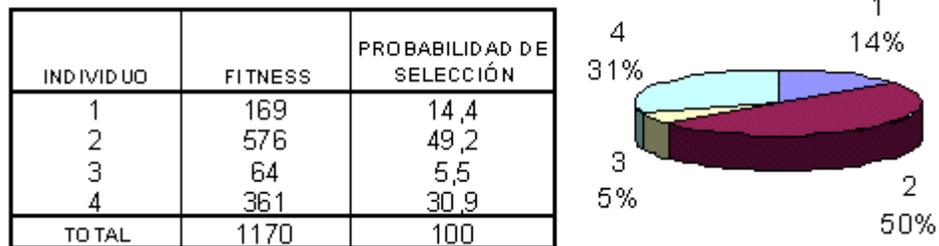


Fig. 6 Operador de Selección [32]

- **Cruzamiento (Crossover):** Este operador trabaja con los padres de los hijos. Siempre selecciona dos individuos de la población para trabajar. Su objetivo es formar dos nuevos hijos a través de la combinación de cromosomas de los padres, de forma tal que estos nuevos descendientes conserven características especiales de los padres. La siguiente figura muestra un ejemplo de operador de cruzamiento.

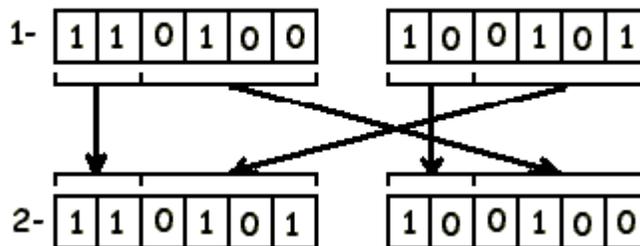


Fig. 7 Operador de Cruce [32]

- **Mutación:** Este operador trabaja sin seguir un orden lógico. Es decir transforma de forma aleatoria una parte del cromosoma de los miembros de la población, produciendo un nuevo descendiente. El principal objetivo de este operador es lograr que la población sea lo más heterogénea posible. La siguiente figura muestra un ejemplo de operador de mutación.

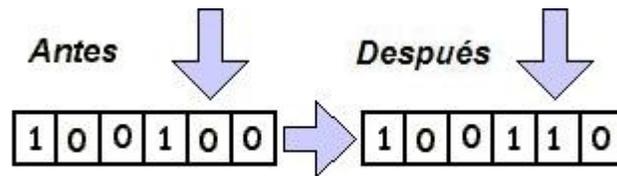


Fig. 8 Operador de Mutación [32]

En la actualidad existen diferentes problemas que son resueltos por los algoritmos genéticos, ejemplo de ello son los problemas de optimización combinatoria. Su solución puede ser aplicada en diferentes diseños, tal es el caso de los diseños de topologías de redes funcionales, los diseños automatizados de equipamiento industrial, y demás diseños para redes geodésicas, por solo mencionar algunos.

También se les suele encontrar en teorías de juego, en el análisis de expresión de genes, en ingeniería y en el problema del viajante que al usar este tipo de algoritmo hace eficaz su solución. El reconocimiento de rostros en la computadora, por sus características complejas, constituye otra de las áreas en que pueden operar los algoritmos genéticos.

2.6 Herramientas existentes para el cálculo de las distancias mínimas

Con la evolución de los Sistemas de Información Geográfica, actualmente existen diversas herramientas y tecnologías en aras de lograr un buen desarrollo y una mayor eficacia en las aplicaciones relacionadas a la localización geográfica. Entre estas herramientas y tecnologías, se encuentran las dedicadas al estudio del Análisis de rutas, las cuales proveen de un soporte que permite la utilización de algoritmos avanzados en la determinación de rutas óptimas.

Muchas son las compañías que desarrollan este tipo de herramientas y que más tarde las comercializan en el mercado de la industria. Algunas de estas herramientas debido a las características y particularidades que poseen, pueden ser privativas o libres. Esto provoca que sean utilizadas por una o varias empresas y que su distribución no sea la más equitativa posible.

2.6.1 Herramienta Rutas

El proyecto Rutas brinda la posibilidad a quienes lo utilizan de poder contar con las diferentes aplicaciones y funciones más atractivas que el mismo posee. Las siguientes, solo constituyen algunos ejemplos dentro de las tantas aplicaciones que Rutas proporciona: [33]

- Localización de clientes, centros de tránsito y almacenes (*geocoding, waypoints*)
- Cálculo y gestión de distancias, tiempos y costes de transporte
- Cálculo y optimización de rutas de transporte
- Definición de zonas de distribución
- Análisis de flujos logísticos. Diseño y análisis de una red de transporte.
- Planificación de rutas de reparto y aprovisionamiento

La herramienta Rutas se encuentra enmarcada en hallar la solución del cálculo de rutas, de la gestión de las mismas y de los problemas reales de flotas de vehículos capacitados. Rutas es el fruto de los conocimientos científicos que poseen los miembros de un grupo de I+D+I²⁸ y de la experiencia de ingenieros y empresas. Windows es el sistema operativo bajo el cual el software Rutas trabaja, y que además necesita de un ordenador bien sofisticado.

Rutas posee la característica de ser un software propietario. Actualmente Rutas se encuentra en desarrollo y a pesar de ello, cuenta con gran demanda entre los usuarios. Sus administradores continúan trabajando y tienen fe en que el producto salga al mercado de la industria con magníficos algoritmos y con la calidad requerida.

2.6.2 Herramienta Grafos

Otra de las herramientas que existen para el cálculo de distancias mínimas es la herramienta Grafos. Es un software para la construcción, edición y análisis de grafos. Grafos pretende ser de utilidad para la docencia y el aprendizaje de la Teoría de Grafos, y otras disciplinas relacionadas como la ingeniería de

²⁸ I+D+I significa Investigación + Desarrollo + Innovación tecnológica

organización industrial, la logística y el transporte, investigación operativa, diseño de redes, entre otras. Grafos se puede usar perfectamente para el modelado y resolución de problemas reales. [34]

El objetivo fundamental de esta herramienta es estudiar, diseñar y llegar a un resultado óptimo a partir de un grafo dado. Brinda además, la posibilidad a quienes la utilizan de trabajar cómodamente, es decir ir analizando el grafo en cuestión sin tener que preocuparse por el algoritmo que la misma usará. En cada caso se irá revisando que el algoritmo sea compatible con la herramienta. En el caso que se piense en un algoritmo que la misma no soporte, ésta le notificará que el algoritmo no es viable para ser usado.

A pesar de ser una herramienta que se distribuye bajo las condiciones: Reconocimiento-NoComercial-CompartirIgual3.0 (*Creative Commons License*)²⁹, no está aprobada su comercialización. Producto de que actualmente se encuentra en desarrollo y en procesos de investigación, la misma admite cualquier tipo de donativo o ayuda, para de esta forma aumentar el número de algoritmos de análisis y de funciones, logrando así un mejor uso para quienes se benefician de él. Algunos de los algoritmos y modelos mencionados en el epígrafe anterior son implementados por Grafos.

Dentro de las primiciales características funcionales que el programa brinda, se encuentran: [34]

- Posibilidad de arco con origen y destino sobre un mismo nodo.
- Valores en arcos (mínimo, máximo y coste).
- Deshacer y rehacer operaciones de edición gráfica.
- Exporta la imagen del grafo a diferentes formatos de gráficos (.gif, .tif, .png, .bmp, .svg)
- Personalización del aspecto gráfico de cada nodo y arco (incluye puntas de flecha)
- Definición de estilo gráfico.
- Visión preliminar e Impresión.

²⁹**Creative Commons License:** Licencia bajo la cual se está en la libertad de copiar, distribuir y comunicar públicamente la obra y hacer obras derivadas. Y todo ello bajo las condiciones de **Reconocimiento:** especificar autor; **No comercial:** No usar para fines comerciales; **Compartir bajo la misma licencia:** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

2.6.3 Herramienta PgDijkstra

Dentro de la Geografía existen numerosos mapas vectoriales los cuales a su vez poseen un sin número de puntos. Estos son muy exactos y específicos por lo que lleva milésimas de segundos para que salgan a la luz. PgDijkstra es una herramienta que mediante la eliminación de vértices en un grafo, reduce estos mapas permitiendo q los mismos se visualicen de una mejor forma.

PgDijkstra es una extensión de enrutamiento para PostgreSQL desarrollado por la camptocamp empresa Suiza / francesa. Se puede utilizar para resolver cualquier problema del camino más corto, especialmente los relacionados geográficamente y puede combinarse con PostGIS. El software está licenciado bajo la GNU GPL y es parte de la distribución Cartoweb. Lleva el nombre de Edsger Dijkstra, inventor del algoritmo subyacente. [35]

2.6.4 Herramienta PgRouting

La herramienta PgRouting tiene como primordial objetivo proporcionar la funcionalidad de enrutamiento a PostGIS/PostgreSQL. Esta es parte de PostLBS, que proporciona las herramientas básicas para los servicios basados en Localización (LBS). Esta herramienta permite realizar un Análisis de rutas así como visualizar la información utilizando el mapserver, openlayers y demás visores de información geográfica. Es una herramienta de fuente abierta (OSS). Actualmente esta herramienta soporta los siguientes algoritmos:

- Algoritmo de búsqueda A*.
- Algoritmo del TSP o del problema del viajante.
- Algoritmo de Dijkstra.

La mayoría de los *frameworks* de desarrollo de Sistemas de Información Geográfica utilizan la PgRouting como herramienta fundamental para llevar a cabo el cálculo de la ruta más corta entre dos nodos de un grafo. Uno de estos es el Cartoweb.

2.7 Conclusiones parciales

Durante el desarrollo de este segundo capítulo se evidencia la utilización en la actualidad de algoritmos para el cálculo de rutas mínimas en un grafo. Se muestran además ejemplos de herramientas dedicadas al estudio del Análisis de rutas. Expuesto todo ello se crea una base teórica en aras de realizar un profundo análisis que patentice la solución propuesta que se dará en el próximo capítulo.

CAPÍTULO III

Presentación de la Solución Propuesta

3.1 Introducción

En el siguiente capítulo se realiza una comparación de las diferentes herramientas existentes para el cálculo de rutas mínimas. Se comparan además los algoritmos y modelos matemáticos que fueron analizados anteriormente. Todas estas comparaciones son de vital importancia para dar sustento a la presentación de la solución propuesta. Con dicha propuesta se suministran los beneficios y mejoras que se obtienen con el uso de dichos algoritmos.

3.2 Comparación entre las herramientas existentes para el cálculo de las distancias mínimas

Al mismo tiempo que los algoritmos y modelos matemáticos constituyen aspectos substanciales a tener en cuenta y que son de vital importancia para el cálculo de rutas mínimas, existen además herramientas, como las que se describieron anteriormente. Estas herramientas a su vez, se encuentran relacionadas con dichos algoritmos y modelos, ya que para su desarrollo necesitan implementar algunos de los mismos. Es por ello, que se hace inevitable conocer a fondo cada una de las facilidades que brinda cada una de estas herramientas.

La herramienta Rutas por ejemplo, para lograr un mejor funcionamiento, se auxilia del problema de los vehículos capacitados para rutas. Con esto garantiza que su resultado en las diferentes funcionalidades que brinda sea lo más exitosa posible. A pesar de ello constituye una herramienta privativa. Esto trae como consecuencia que no tenga mucha aceptación entre los usuarios. Grafos es otra herramienta que principalmente se encuentra dedicada a la docencia, es decir brinda sus servicios a la enseñanza basados en la Teoría de Grafos. Asimismo cuenta con numerosos algoritmos los que utiliza con propósitos educativos. Esta herramienta a pesar de ser libre, tiene la peculiaridad de ser distribuida por una licencia especial, la cual no permite que se utilice con fines lucrativos ni comerciales.

PgRouting es otra herramienta que tiene la primordial característica de tener código abierto lo que hace que se convierta en una herramienta libre. La misma implementa para el cálculo de rutas mínimas algoritmos tales como: A*, Dijkstra y el Problema del Viajante. Otra de las herramientas utilizada hace varios años atrás es PgDijkstra. Con el desarrollo que tienen los sistemas de información geográfica y los descubrimientos realizados acerca del estudio del cálculo de la distancia mínima, en la actualidad, esta herramienta no es muy utilizada, en otras palabras con el paso del tiempo se ha vuelto obsoleta. De ahí entonces, que muchos prefieran utilizar PgRouting.

Es por ello que para seleccionar la herramienta a utilizar se hace obligatorio realizar una tabla de comparación. En la Tabla 1 se mencionan los algoritmos que cada una de ellas implementa y las licencias bajo las cuales trabajan. Factores estos importantes a la hora de hacer la selección de la herramienta.

Herramienta	Algoritmos que utiliza	Licencia que posee
Rutas	Problema de Rutas para Vehículos Capacitados (CVRP)	Privativa (A pesar de que aún se encuentra en desarrollo)
Grafos	Dijkstra, Bellman-Ford, Floyd- Warshall, Kruskal, Prim, Ford, Problema del Traspordo, Problema de Asignación, Problema del Viajante, Problema del M-Viajante, Algoritmo de Rutas, Problema de las M-Rutas, Problema de Rutas	Libre (Reconocimiento-NoComercial-CompartirIgual 3.0. (<i>Creative Commons License</i>))

	para Vehículos Capacitados	
PgDijkstra	Dijkstra	Libre (GNU GPL)
PgRouting	A*, Problema del Viajante, Dijkstra, <i>Driving Distance</i>	Libre (GNU GPL)

Tabla 1. Comparación entre herramientas para el estudio de rutas mínimas

3.3 Comparación entre algoritmos y modelos matemáticos

Valiosos son los algoritmos y modelos matemáticos que en el mundo existen. Automatizar y mejorar la resolución de problemas, se convierten sin duda alguna en importantes objetivos de estos algoritmos. Varios de estos algoritmos se describieron en capítulos anteriores. Algunos de ellos, por sus características y por su naturaleza pueden resolver problemas de optimización de rutas, otros no. Es por ello, que en el momento de seleccionar el más adecuado para ser utilizado en la resolución de un determinado problema, se deben tener en cuenta alguna de estas características.

Los algoritmos son implementados por diferentes herramientas, las cuales por supuesto también se encuentran dedicadas al estudio de rutas mínimas. Estos algoritmos se diferencian además por dar solución a diferentes problemas clásicos, es decir, los problemas referentes a: un origen y muchos destinos, un origen y un solo destino, entre otros ejemplos. Además se tiene la complejidad de cada uno de ellos, característica ésta, que también marca la diferencia de un algoritmo matemático con los restantes, ya que no es más que el tiempo que demora el algoritmo en dar su solución.

Capítulo III Presentación de la Solución Propuesta

Por tanto, el grado de complejidad de los algoritmos, las herramientas que lo implementan y los problemas clásicos que resuelven, constituyen los parámetros principales que han sido seleccionados para establecer la comparación que sustentará la solución propuesta. Esta información será recogida en una tabla de comparaciones, la cual sirve de apoyo para la selección de la solución propuesta. La Tabla 2, corrobora lo expuesto hasta el momento.

Algoritmo	Grado de Complejidad	Herramientas que lo implementan	Problemas clásicos que resuelven
Dijkstra	$O(n^2)$	Grafos PgRouting	Va de uno a los restantes vértices del grafo
Floyd	$O(n^3)$	Grafos	Problema de los caminos más cortos entre los pares de vértices
Prim	$O(n \log n)$	Grafos	Problema del Árbol de Expansión Mínima
A*	$O(n \log n)$	PgRouting	Encuentra un elemento dado una estructura de datos y localiza el menor camino entre dos nodos

Kruskal	$O(n \log n)$	Grafos	Problema del Árbol de Expansión Mínima
Problema del Traslado	Exponencial (Solución para casos particulares)	Grafos	Va desde unos puntos orígenes a unos puntos destinos, pasando por el camino más corto
Problema del Viajante del Comercio	Exponencial (Solución para casos particulares)	Grafos PgRouting	Encuentra la distancia total mínima en un grafo de un vértice a los restantes
Problema de Rutas para Vehículos Capacitados (CVRP)	Exponencial (Solución para casos particulares)	Grafos Rutas	Encontrar un conjunto de rutas dado una flota que puedan proporcionar asistencia a un grupo de clientes

Tabla 2. Comparación entre algoritmos y modelos matemáticos

3.4 Solución propuesta

Con el objetivo de proporcionarle un trabajo más factible a los desarrolladores de la UCI, especialmente para aquellos que trabajan dentro del módulo Análisis del terreno en la Plataforma GENESIG, se realiza una propuesta de herramientas y algoritmos matemáticos. Esta propuesta agilizará además el trabajo

referente a la implementación del módulo en cuestión y el gasto de recursos computacionales será mínimo. Como parte de este proceso de selección de propuesta se realizaron diferentes tablas de comparación, las cuales fueron reflejadas anteriormente.

Realizados los estudios necesarios y las pruebas pertinentes, se decidió seleccionar la herramienta PgRouting. Esta selección se basó principalmente en sus ventajas. Una de estas ventajas es que PostGIS se encuentra vinculado con PgRouting. PostGIS es un módulo con diferentes funcionalidades para la localización geográfica, al cual se le añade PgRouting y la mezcla de ambos aportan valiosos objetos geográficos a las bases de datos que se utilizan en el módulo de Análisis del terreno, en la funcionalidad Análisis de Rutas, dentro de la Plataforma GENESIG.

Otra de las ventajas es que es una herramienta totalmente libre. Esto garantiza una buena aceptación en el mercado de la industria, debido a que muchos usuarios no tienen sustentos económicos para pagar licencias privativas. Partiendo entonces de aquí, se tiene además que esta herramienta implementa los algoritmos de menor tiempo de ejecución para llegar a un resultado final, como fue mostrado en la tabla anterior. De ahí entonces, la selección de los algoritmos. Un ejemplo de ello, es el algoritmo Dijkstra, que resuelve el problema de las distancias mínimas entre dos puntos y posee un tiempo de ejecución $O(n^2)$.

No obstante se tiene el algoritmo de búsqueda A^* , el cual posee un tiempo de ejecución mucho menor que Dijkstra ($O(n \log n)$). También trabaja con estructuras de datos y es muy utilizado donde se hace necesario mejorar y perfeccionar la solución que ya se tiene. Además soluciona algunos casos particulares del problema del viajante mediante algoritmos genéticos. Otro de los algoritmos que implementa PgRouting es el *Driving Distance*. Este algoritmo no es muy conocido, pero su esencia radica en buscar las distancias mínimas a partir de obstáculos e inconvenientes que deban ser tomados en cuenta para llegar al destino final.

3.5 Incluir mejoras para Herramienta PgRouting

La herramienta PgRouting, como se pudo comprobar hasta el momento, posee las características que cualquier otra herramienta pudiese pretender. Posee además la atención de cualquier cliente. Esto se debe a que cuenta con código fuente abierto y brinda además, magníficos servicios basados en la

localización geográfica, así como la información geográfica que permite visualizar a través de los visores de información geográfica.

Sin embargo, actualmente con el uso de esta herramienta no se pueden resolver algunos problemas de distancias mínimas. El motivo de ello, es que a pesar de los excelentes servicios que brinda, no tiene los suficientes algoritmos matemáticos implementados. Debido a esto, se hace necesario incluir nuevos algoritmos y modelos matemáticos. Ejemplo de ello, lo constituyen los algoritmos heurísticos y meta-heurísticos.

Por la gran importancia que estos algoritmos poseen en el cálculo de rutas mínimas, es necesario que en próximas versiones de PgRouting se tenga en cuenta la inclusión de los mismos. De esta forma los algoritmos meta-heurísticos, mediante la búsqueda tabú solucionarán los casos particulares del problema del trasbordo, el cual garantiza el envío de mercancías desde unos puntos orígenes a unos puntos destinos pasando por unos intermedios, sería un servicio más que brindase la herramienta PgRouting.

3.5 Validación

Actividades como la planificación de tiempo o la revisión de las tareas, por solo mencionar algunas, constituyen varias de las normativas que se siguen periódicamente en la plataforma GENSIG. Cada trabajador tiene un determinado rol, ejemplo de ello es el planificador, el cual mediante un cronograma de fechas, está a cargo de establecer el tiempo que durará una tarea en arribar a su fin.

Por su parte el planificador de la plataforma GENESIG, estimó que la funcionalidad Análisis de rutas dentro del módulo Análisis del terreno, quedase lista en un período de tres meses como mínimo debido al poco conocimiento de los algoritmos que utilizarían los desarrolladores del equipo. Con la ayuda de esta propuesta de solución, estos tres meses se redujeron a un mes. El trabajo les fue más fácil y problemas que existían sin tener respuestas en el módulo y que no habían sido resueltos hasta el momento encontraron su solución y sus resultados fueron satisfactorios.

Encontrar la ruta mínima, resolver los problemas del viajante, de análisis isocrónicos³⁰, y de circulación fueron resultados esperados por todos en la plataforma GENESIG. Para tener una mayor visión de alguno de estos resultados, se muestran tres figuras, las cuales dan fe de lo que se ha venido explicando hasta el momento. La figura 9 y 10 muestra el avance que tuvo la funcionalidad Análisis de rutas, mientras que la figura 11 muestra el desarrollo que alcanzó el Análisis isocrónicos, ambas funcionalidades se encuentran dentro del módulo Análisis del terreno.

³⁰ **Análisis Isocrónicos:** Conjunto de vías o alternativas que se tienen para saber las distancias que existen desde un determinado lugar hasta otros puntos destinos.

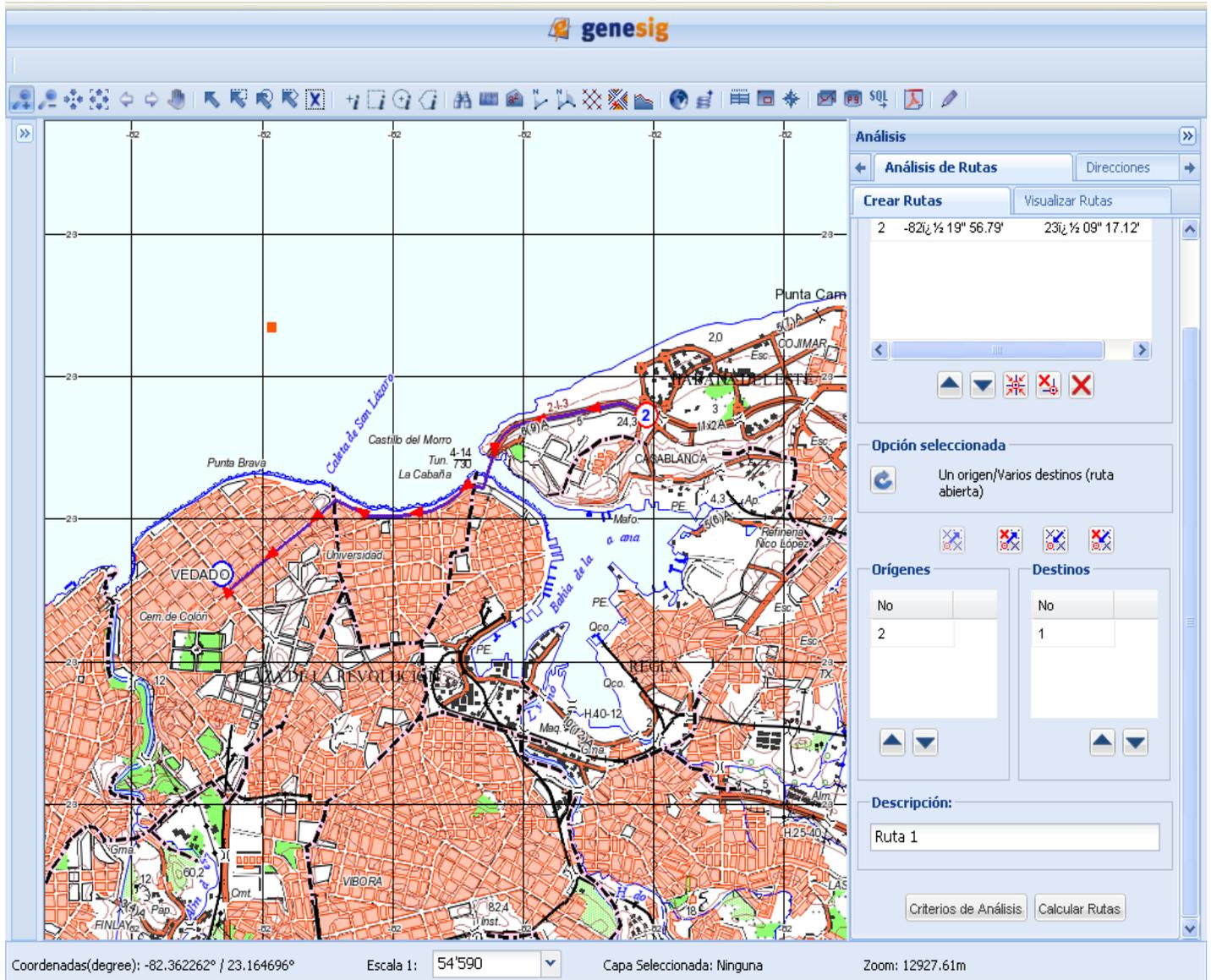


Fig. 9 Análisis de rutas

Capítulo III Presentación de la Solución Propuesta

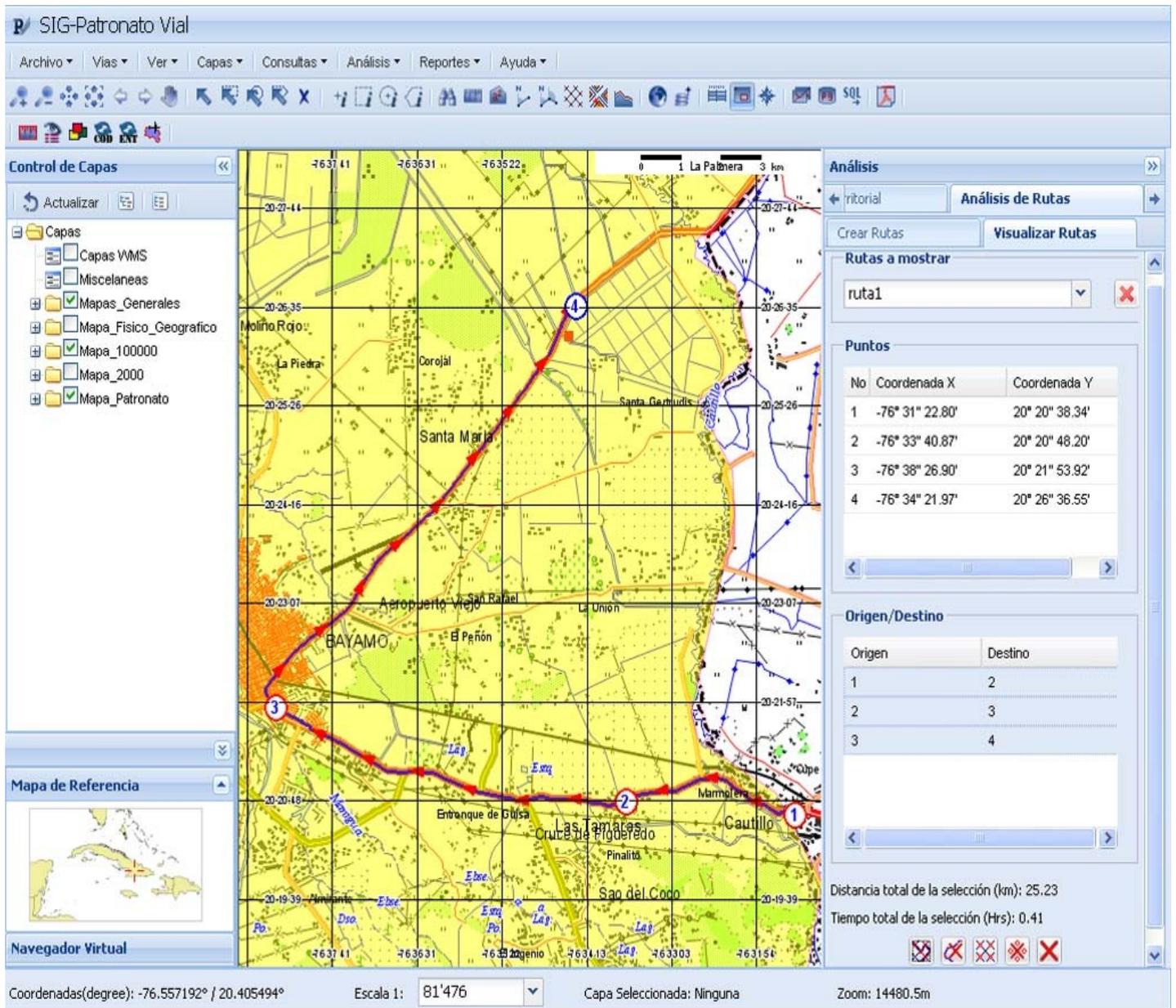


Fig. 10 Análisis de rutas

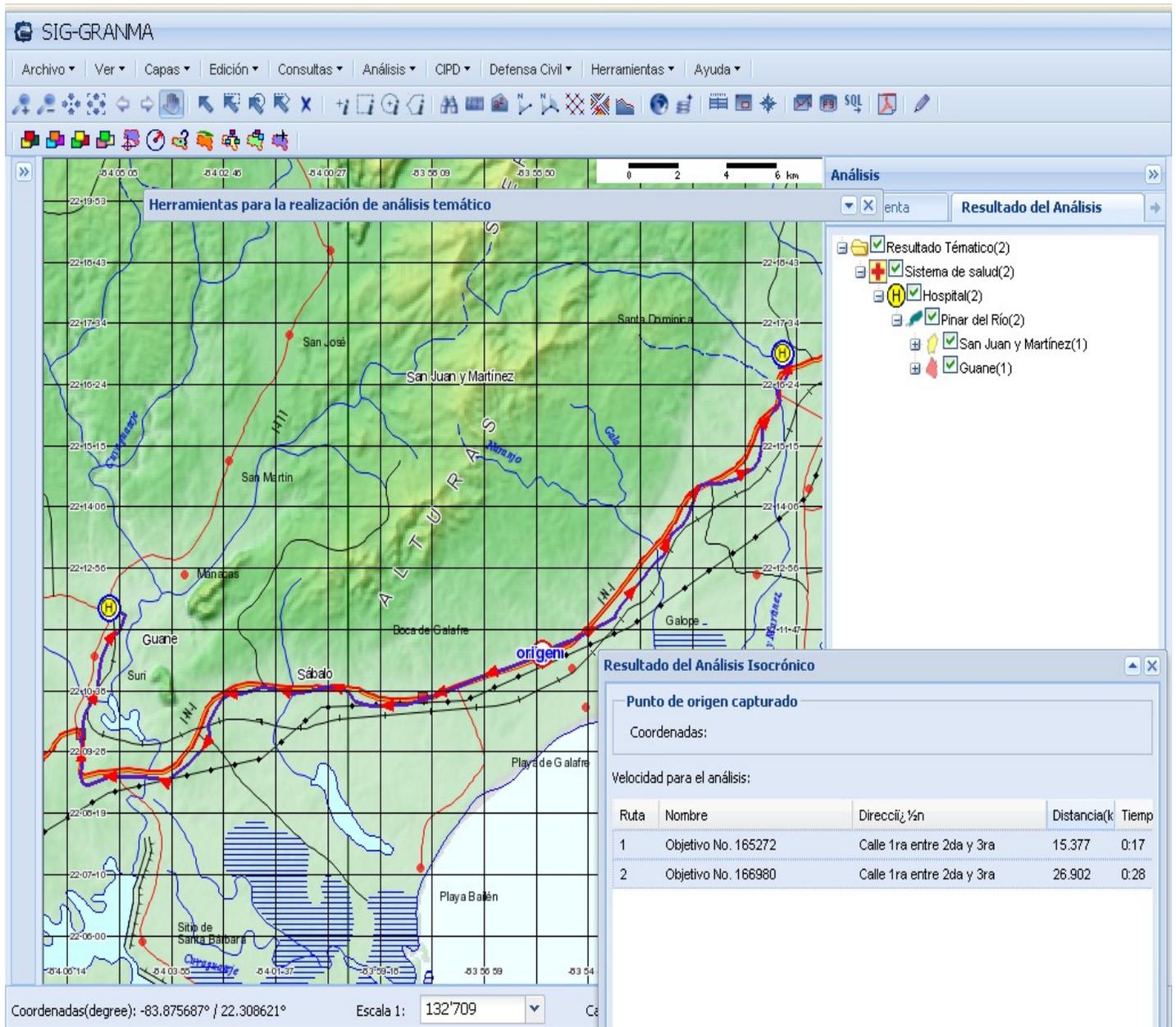


Fig. 11 Análisis isocrónicos

3.6 Conclusiones parciales

El presente capítulo durante todo su desarrollo tuvo como principal desafío formar la solución propuesta. Para ello, como se pudo confirmar se analizaron a fondo las principales características de los algoritmos y modelos matemáticos, así como de las herramientas, a través de tablas de comparación. Se vieron además las facilidades y ventajas del uso de estos algoritmos, logrando realizar la solución propuesta.

CONCLUSIONES GENERALES

Por las difíciles situaciones económicas que actualmente está atravesando el país se hace necesario contar con herramientas o aplicaciones en las que reducir el coste o minimizar el gasto de recursos materiales y esfuerzo humano sean los objetivos fundamentales, sin alterar la eficiencia de las actividades.

El objetivo de este Trabajo de Diploma es obtener una propuesta de algoritmos para llevar a cabo el Análisis de rutas en la Plataforma GENESIG. Los algoritmos, propuestos en este trabajo son de vital importancia para el desarrollo y avance del módulo de Análisis del terreno en la Plataforma GENESIG. Luego de terminada la investigación y realizada la propuesta de solución, se concluye que con el uso de la misma:

- Se garantizará que los implementadores pertenecientes a la Plataforma GENESIG puedan contar con el conocimiento y las herramientas necesarias que permitan realizar la implementación del módulo de Análisis del terreno, específicamente la funcionalidad Análisis de rutas.
- Se agiliza el trabajo de implementación referente a la funcionalidad Análisis de rutas, dentro del módulo Análisis del terreno.
- Se realizan búsquedas del camino más corto entre dos o más puntos visualizados en un mapa en un menor tiempo.

De esta forma, obstáculos e impedimento que existían en la Plataforma GENESIG, dentro de la Universidad de las Ciencias Informáticas, fueron resueltos. La propuesta definida le facilitó al equipo de desarrolladores pertenecientes al módulo Análisis del terreno en la Plataforma GENESIG, obtener en un tiempo considerable una mejor implementación de las diferentes funcionalidades que la misma brinda.

RECOMENDACIONES

Se recomienda en este trabajo, dar cumplimiento a los siguientes puntos:

- Poner en práctica la solución propuesta en este trabajo, la cual facilitará y ahorrará tiempo en la implementación de estas herramientas de análisis en el módulo de análisis del terreno.
- Continuar el estudio de nuevos algoritmos y modelos matemáticos que permiten el cálculo de distancias mínimas.
- Añadir nuevos algoritmos y modelos matemáticos a la herramienta PgRouting.
- Valorar la posibilidad de hacer extensiva la utilización de esta propuesta a otras instituciones con el fin de proporcionar ayuda a otros usuarios.
- Utilizar la documentación generada sobre algoritmos matemáticos como material de consulta en la realización de trabajos u otros documentos.

TRABAJOS CITADOS

1. Salinas, J., Información Geográfica, Software Libre e Infraestructuras de. Cuba. 2007.
2. Dieguez, N., Impacto de los SIG en la sociedad. Cuba. 2003.
3. SEGEMAR, Cartografía. [cited 2009 9 de septiembre];
]. Available from: <http://www.segemar.gov.ar/cartografia/cartografiadigital.htm>.
4. Venuda, F., GIS (Geographic Information System) in libraries: first part. (English)
5. Autoaprendizaje SIG. 12 March 2001 [cited 2010 15 de Enero]; November 1999:[Available from: <http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISTheory.htm>
6. Aplicaciones SIG. [cited; Available from: <http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISTheory.htm#apps>.
7. R., R., "Welcome to OR Territory" OR/MS 1999: p. Today pp.40-43 August.
8. Alvarado, G.S.M. METODOLOGÍA Y MÉTODOS EN SISTEMAS. [cited 2010 25 de Enero]; Available from: http://www.centrogeo.org.mx/curriculum/GermanMonroy/pdf/metodologia_y_metodos_en_sistemas_1981.pdf.
9. M., D.I.F.B., INVESTIGACIÓN DE OPERACIONES
CURSO DE LA ESCUELA DE ADMINISTRACIÓN Y CONTADURÍA. Semestre Marzo - Julio 2004 p. 10.
10. Kioskea. El concepto de red [cited 2010 19 de marzo]; Available from: <http://es.kioskea.net/contents/initiation/concept.php3>.
11. Análisis de Redes. 2003-2006 [cited 2010 19 de marzo]; Available from: <http://www.omerique.net/calculmat/redes.htm#Vamos>.
12. MURTHY, D.N.P.P., N.W.; RODIN, E. T. , Mathematical Modelling. 1992.
13. MURUA, L.P., CAPITULO1 CONCEPTO DE ALGORITMO. INSTITUTO TECNOLÓGICO. ANALISIS DE LENGUAJE p. 19.
14. ALGORITMOS.DEFINICION. p. 6.
15. Villalobos, A.R. Grafos. Algoritmo de Dijkstra. [cited 2010 18 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/Dijkstra.htm>.

16. Villalobos, A.R. Grafos. Algoritmo de Floyd-Warshall [cited 2010 18 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/FloydWarshall.htm>.
17. Villalobos, A.R. Grafos. Algoritmo de Prim. [cited 2010 18 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/Prim.htm>
18. Villalobos, A.R. Grafos. Algoritmo de Ford-Fulkerson. [cited 2010 18 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/FordFulkerson.htm>
19. Villalobos, A.R. Grafos. Algoritmo de Kruskal. [cited 2010 18 de Marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/Kruskal.htm>.
20. Villalobos, A.R. Grafos. Algoritmo de Bellman-Ford. [cited 2010 18 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/Ford.htm>.
21. Tema 6: Problemas Especiales de Programación Lineal. [cited 2010 19 de marzo]; Available from: <http://www.uv.es/martinek/material/Tema6.pdf>
22. Egea, C.P., et al., Análisis del software Grafos Métodos Cuantitativos III. 2007/2008.
23. Francesco Narducci, J.M.D., Jairo R. Montoya, RESOLUCIÓN DEL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS CON LIMITACIONES DE CAPACIDAD UTILIZANDO UN PROCEDIMIENTO METAHEURÍSTICO DE DOS FASES, in Revista EIA. Diciembre 2009. p. p. 23-38.
24. Villalobos, A.R., GRAFOS, Problema de Rutas para Vehículos Capacitados.
25. Bijit, P.L.S., UNIVERSIDAD TECNICA FEDERICO SANTA MARIA DEPARTAMENTO DE ELECTRONICA Programación en Pascal Capítulo 25. Algoritmos heurísticos. 2003.
26. Martí, R., Procedimientos Metaheurísticos en Optimización Combinatoria. 2009.
27. Búsqueda Heurística. 2006/2007 [cited 2010 18 de Marzo]; Available from: http://www.lsi.upc.edu/~bejar/ia/transpas/teoria/2-BH2-Busqueda_heuristica.pdf.
28. Glover, F. and M. Laguna, TABU SEARCH. p. 18.
29. Díaz, A., F. Glover, and H.M. Ghaziri, Optimización Heurística y Redes Neuronales. . 1996.
30. Rocío, R., Rossana, Rafael, Algoritmos Genéticos. 2002. Versión 0.91a7: p. 19.
31. Guervós, J.J.M., Informática evolutiva: Algoritmos genéticos.
32. López, J.A., Computación Evolutiva: Introducción y Conceptos básicos. 2000.
33. Villalobos, A.R., Rutas. 2006 - 2009

34. Villalobos, A.R. Grafos [cited 2010 19 de marzo]; Available from: <http://personales.upv.es/arodrigu/grafos/index.htm>
35. Mapbender. 9 de Julio de 2006 [cited 2010 18 de marzo]; Available from: <http://www.mapbender.org/Pgdijkstra>.

BIBLIOGRAFÍA CONSULTADA

1. Dijkstra, E. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* 1: 269-271.
2. Edmonds, J. a. K., R. M. (1972). "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems." 248-264.
3. Floyd, R. W. (1962). Algorithm 97 (SHORTEST PATH)". *Communications of the ACM*, 5(6):345.
4. Ford, L. R. J. (1956). "Network Flow Theory." P-923.
5. Gabow, G., Spencer, and Tarjan (1986). "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, vol. 6." pp. 109-122.
6. GIMÉNEZ Cánovas, D. (2001). "Apuntes y Problemas de Algorítmica."
7. Glover, F. (Summer 1989). "Tabu Search – Part I." v. 1, n. 3: pp. 190-206.
8. Glover, F. y. M., Belén (2003). Búsqueda tabú. *Revista Iberoamericana de Inteligencia Artificial*. N.19: pp. 29-48.
9. Heidelberg, S. B. "Trabajos de Estadística y de Investigación Operativa." (Volume 12, Numbers 1-2 / February, 1961): 47-69.
10. Ian., P. (1995). "Problems on Algorithms."
11. Karger, K., and Tarjan (1995). "A randomized linear-time algorithm to find minimum spanning trees." pp. 321-328.
12. KNUTH, D. E. (1973). "Sorting and Searching." volume 3 de "The Art of Computer Programming", segunda edición, Addison-Wesley Publishing Company, Reading, Massachusetts.
13. Laguna, M. (Abril 1994). "A Guide to implementing Tabu search." v. 4, n. 1: pp. 5-25
14. Melián, B. P., Jose A (2003). Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial*. N.19 pp. 7-28
15. Olivera, A. (Agosto 2004). "Heurísticas para Problemas de Ruteo de Vehículo."
16. Prim, R. C. (1957). "Shortest connection networks and some generalizations." pp. 1389-1401.
17. Rodríguez-Villalobos, A. (2008). "Integración de un SIG con modelos de cálculo y optimización de rutas de vehículos CVRP y software de gestión de flotas."
18. Warshall, S. (1962). "A theorem on boolean matrices." *Journal of the ACM*, 9(1):11-12."

19. Willard, F. a. (1990). ""Trans-dichotomous algorithms for minimum spanning trees and shortest paths", 31st IEEE Symp. Foundations of Comp. Sci." pp. 719-725.
20. Winston, W. L. (2005). "Investigación de Operaciones. Aplicaciones y Algoritmos."
21. Yu, Z., X. Ma, et al. (May2010). "Mathematical modeling of combustion in a grate-fired boiler burning straw and effect of operating conditions under air- and oxygen-enriched atmospheres." Vol. 35 Issue 5, p895-903, 9p.

GLOSARIO DE TÉRMINOS

Algoritmo: Proveniente del latín dicitur *algorithmus*. Un algoritmo constituye una lista bien definida, ordenada y finita de operaciones que se ejecutan en un tiempo determinado empleando cierta cantidad de recursos computacionales para hallar la solución a un problema.

Árbol: Se llama árbol al grafo que no tiene ciclos y que enlaza todos sus puntos. Su importancia radica en que los arboles constituyen grafos que conectan todos los vértices de forma tal que se utilice el menor número de aristas posibles.

Arista: Es la unión que existe entre nodos o vértices, es decir, la línea que los une hasta formar un árbol o grafo. Estas aristas pueden estar dirigidas o no. En caso de estar dirigidas, se estaría en presencia de un grafo dirigido, en caso contrario, sería un grafo no dirigido.

Camino o Ruta: Es la forma de llegar de un punto inicio a un punto final. En un grafo es el arco que va desde un nodo hasta otro, formando un camino. Se llama ciclo al camino que une a un nodo consigo mismo.

GENESIG: Herramienta informática utilizada para el desarrollo de Sistemas de Información Geográfica, implementada con herramientas y tecnologías libres. Es un proyecto que surge gracias a los esfuerzos de los especialistas de la Universidad de las Ciencias Informáticas, Las Fuerzas Armadas Revolucionarias y la Empresa de GEOCUBA. Presenta varios módulos, que se han ido implementando poco a poco, logrando resultados en los que se evidencian grandes proyecciones de expansión y madurez técnica.

GNU/GPL: (Licencia Pública General de GNU). La licencia libre GNU/GPL posibilita la modificación y redistribución del software únicamente bajo dicha licencia. Partiendo del punto de vista del software libre, existen diferentes variantes del concepto o grupos de licencias, ejemplo de ello son: AGPL, BSD, Licencias estilo MPL y derivadas, entre otras.

Grafo: Un grafo es la representación de una cantidad determinada de vértices interconectados por aristas. Estas líneas (aristas) pueden o no tener sentido, de ahí que existan grafos dirigidos y grafos no dirigidos.

Muchas redes de uso cotidiano pueden ser modeladas con un grafo: una red de carreteras que conecta ciudades, una red eléctrica o la red de drenaje de una ciudad.

Inteligencia Artificial: Ciencia de la computación encargada de la creación de máquinas dedicadas a la implementación de tareas relacionadas con el comportamiento humano, logrando gran similitud al pensamiento del hombre.

Investigación de Operaciones: Ciencia dentro de la Matemática encargada de realizar el procedimiento científico destinado a diseñar y operar problemas complejos de la dirección y administración de grandes sistemas que forman una organización compleja en las cuales las decisiones son muy importantes y difíciles de elegir. Es por ello que la investigación de operaciones constituye un factor importantísimo en la toma de decisiones para lograr la optimización de los recursos.

Open Source: (*Open Source* o Código abierto). Término por el cual es conocido el software, una vez que es distribuido y desarrollado libremente. Tiene un punto de vista más orientado a los beneficios prácticos, es decir, tiene la peculiaridad de compartir el código en cuestiones morales y/o filosóficas, las cuales se destacan en el llamado software libre.

Rol: Papel que juega una persona dentro de una estructura, en este caso un proyecto. Es decir, desarrolladores, analistas, jefe de proyecto, arquitectos, entre otros, constituyen roles en un proyecto.

Vértice o Nodo: Los vértices constituyen uno de los dos elementos que conforman un grafo. Es considerada la unidad fundamental, ya que son tratados como objetos indivisibles y sin propiedades; ejemplo de ello es la representación en un grafo de una red carretera en la que las ciudades o construcciones representan los vértices.