

Universidad de los Andes
Facultad de Ingeniería

Departamento de Ingeniería
de Sistemas y Computación



Universidad
de los Andes

Tesis de grado
MAGÍSTER EN INGENIERÍA DE SISTEMAS

GENERACIÓN DE RUTAS DE COLEGIO APOYADO EN EL
USO DE HERRAMIENTAS SIG

PRESENTADO POR:
CÉSAR ALBERTO NIÑO MÉNDEZ

ASESOR DE TESIS:
Ing. GERMÁN ENRIQUE BRAVO CÓRDOBA

Bogotá D.C., Agosto de 2006

A mis Padres, Hermano y Tía,
por ser fuente de inspiración
y motivación para
buscar un mejor mañana.

AGRADECIMIENTOS

Deseo expresar mis agradecimientos:

Al profesor Luís Alejandro Camacho Botero, por su apoyo y orientación durante la maestría.

A Robert Antonio Barraza Gamero y Julián Orlando Díaz Rivera, compañeros de incondicional apoyo.

Tabla de Contenido

1	INTRODUCCIÓN.....	1
2	OBJETIVOS.....	3
2.1	OBJETIVO GENERAL.....	3
2.2	OBJETIVOS ESPECÍFICOS	3
3	ALCANCE	5
4	MARCO TEÓRICO.....	6
4.1	TRANSPORTE URBANO.....	6
4.1.1	<i>Transporte público.....</i>	6
4.1.2	<i>Transporte escolar.....</i>	7
4.1.3	<i>Entidades reguladoras.....</i>	8
4.1.3.1	<i>Secretaría de tránsito y transporte de Bogotá D.C.....</i>	8
4.1.3.2	<i>Secretaria de educación de Bogotá D.C.....</i>	8
4.2	ALGORITMOS.....	8
4.2.1	<i>Algoritmos deterministas.....</i>	9
4.2.2	<i>Algoritmos no deterministas.....</i>	9
4.3	COMPLEJIDAD.....	9
4.3.1	<i>Complejidad L.....</i>	10
4.3.2	<i>Complejidad P.....</i>	10
4.3.3	<i>Complejidad NP.....</i>	10
4.3.4	<i>Complejidad NP-Completo.....</i>	10
4.3.5	<i>Complejidad NP-Hard.....</i>	10
4.4	ANÁLISIS DEL PROBLEMA VRP.....	11
4.4.1	<i>Descripción general del problema VRP.....</i>	11
4.4.2	<i>Elementos que conforman el problema VRP.....</i>	12
4.4.2.1	<i>Clientes</i>	12
4.4.2.2	<i>Depósitos.....</i>	12
4.4.2.3	<i>Vehículos</i>	13
4.4.3	<i>Parámetros del problema VRP.....</i>	13
4.4.4	<i>Variantes del problema VRP.....</i>	14
4.5	TÉCNICAS PARA SOLUCIONAR VRP.....	16
4.5.1	<i>Métodos exactos</i>	16
4.5.2	<i>Heurísticas.....</i>	16
4.5.3	<i>Metaheurísticas.....</i>	17
4.6	ANÁLISIS DE ALTERNATIVAS.....	18
4.6.1	<i>Clarke & Wright.....</i>	18
4.6.2	<i>(*)Tabu search.....</i>	20
4.6.2.1	<i>Simulated annealing.....</i>	21
4.6.2.2	<i>Criterios de modelación.....</i>	23
4.6.2.3	<i>Algoritmo.....</i>	26
5	DESCRIPCIÓN DEL PROBLEMA.....	27
5.1	PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS DE TRANSPORTE ESCOLAR.....	27
5.2	SOLUCIONES CONOCIDAS AL PROBLEMA DE TRANSPORTE ESCOLAR.....	28
5.2.1	<i>Proyecto SIGTEBAL</i>	29
5.3	DEFINICIÓN DE TÉRMINOS.....	29
5.4	RUTAS DE COLEGIO COMO VRP Y NP-HARD.....	30
5.5	<i>(**)DEFINICIÓN MATEMÁTICA DEL PROBLEMA.....</i>	31
6	SOLUCIÓN PROPUESTA.....	37
6.1	MÓDULO DE GEOCODIFICACIÓN	37

6.2	MÓDULO DE SEGMENTACIÓN.....	38
6.3	MÓDULO MATRIZ ORIGEN-DESTINO.....	39
6.4	MÓDULO GENERADOR DE RUTA	40
6.5	MÓDULO DE POST-OPTIMIZACIÓN MEDIANTE IMPLOSIÓN-EXPLOSIÓN	40
6.6	MÓDULO DESPLIEGUE.....	41
7	IMPLEMENTACIÓN.....	42
7.1	IMPLEMENTACIÓN MÓDULO DE GEOCODIFICACIÓN.....	43
7.2	(*) IMPLEMENTACIÓN DEL MÓDULO DE SEGMENTACIÓN.....	45
7.3	IMPLEMENTACIÓN MÓDULO MATRIZ ORIGEN-DESTINO	48
7.4	IMPLEMENTACIÓN MÓDULO GENERADOR DE RUTA.....	49
	7.4.1 <i>Clarke & Wright</i>	49
	7.4.2 <i>Simulated annealing</i>	51
7.5	(*)IMPLEMENTACIÓN MÓDULO DE POST-OPTIMIZACIÓN MEDIANTE IMPLOSIÓN-EXPLOSIÓN.....	55
7.6	IMPLEMENTACIÓN MÓDULO DESPLIEGUE.....	57
8	RESULTADOS.....	58
8.1	RESULTADOS SEGMENTACIÓN.....	58
8.2	RESULTADOS SIMULATED ANNEALING.....	60
	8.2.1 <i>Resultados preliminares esquema de enfriamiento</i>	61
	8.2.2 <i>Resultados esquema de enfriamiento</i>	63
	8.2.3 <i>Resultados temperatura inicial</i>	65
	8.2.4 <i>Resultados número de iteraciones</i>	65
	8.2.5 <i>Resultados finales</i>	66
8.3	COMPARACIÓN TABU SEARCH – SIMULATED ANNEALING.....	66
8.4	COMPARACIÓN ENTRE CLARKE & WRIGHT Y LAS METAHEURÍSTICAS.....	68
9	CONCLUSIONES.....	70
10	RECOMENDACIONES.....	72
11	REFERENCIAS.....	73

Lista de Tablas

<i>Tabla 4-1 Notación complejidad.....</i>	<i>9</i>
<i>Tabla 7-1 Formato inicial dirección.....</i>	<i>44</i>
<i>Tabla 7-2 Formato final dirección.....</i>	<i>45</i>
<i>Tabla 7-3 Ejemplo archivo entrada matriz origen destino.....</i>	<i>48</i>
<i>Tabla 7-4 Ejemplo resultado matriz origen destino.....</i>	<i>49</i>
<i>Tabla 8-1 Resultados validación segmentación.....</i>	<i>59</i>
<i>Tabla 8-2 Parámetros de Simulated Annealing.....</i>	<i>60</i>
<i>Tabla 8-3 Esquema nomenclatura pruebas.....</i>	<i>61</i>
<i>Tabla 8-4 Resultados preliminares.....</i>	<i>61</i>
<i>Tabla 8-5 Mejores resultados.....</i>	<i>63</i>
<i>Tabla 8-6 Escenarios.....</i>	<i>64</i>
<i>Tabla 8-7 Resultados.....</i>	<i>65</i>
<i>Tabla 8-8 Resultados iteraciones.....</i>	<i>65</i>
<i>Tabla 8-9 Criterios escogidos.....</i>	<i>66</i>
<i>Tabla 8-10 Resultados para Simulated annealing.....</i>	<i>67</i>
<i>Tabla 8-11 Resultados para tabu search.....</i>	<i>67</i>
<i>Tabla 8-12 Mejoras en los resultados.....</i>	<i>69</i>

Lista de Figuras

Figura 4-1 Esquema típico de datos de entrada a VRP.....	11
Figura 4-2 Esquema típico de datos de salida de VRP.....	11
Figura 4-3 Esquema Clarke & Wright.....	19
Figura 4-4 Algoritmo Búsqueda Tabú.....	21
Figura 4-5 Esquemas de enfriamiento.....	24
Figura 4-6 Algoritmo Simulated Annealing.....	26
Figura 6-1 Esquema solución propuesta.....	37
Figura 6-2 Esquema geocodificación.....	38
Figura 6-3 Esquema segmentación.....	38
Figura 6-4 Matriz de costos.....	39
Figura 6-5 Esquema matriz origen-destino.....	39
Figura 6-6 Esquema generador ruta.....	40
Figura 6-7 Esquema implosión - explosión.....	41
Figura 6-8 Esquema despliegue.....	41
Figura 7-1 Segmentación con KMeans (1000 estudiantes – 30 segmentos).....	47
Figura 7-2 Centroides después del proceso de KMeans.....	47
Figura 7-3 Segmentación con KMeans con capacidad.....	48
Figura 7-4 Esquema Clarke & Wright.....	50
Figura 7-5 Esquema RE-INSET.....	53
Figura 7-6 Esquema SWAP.....	54
Figura 7-7 Puntos alrededor de un estudiante, 4 segmentos, 200 metros.....	56
Figura 7-8 Obtención puntos alrededor de un estudiante.....	56
Figura 8-1 Modelo segmentación.....	58
Figura 8-2 Comportamiento del modelo.....	59
Figura 8-3 Escenarios de prueba preliminar.....	63
Figura 8-4 Resultados pruebas.....	64
Figura 8-5 Resultados en costo y tiempo de las dos metaheurísticas.....	68

1 Introducción

El diseño de rutas de transporte escolar es un tema de interés para algunos sectores de la sociedad, entre ellos, las compañías transportadoras, los planteles educativos, las autoridades nacionales y regionales, los padres de familia y por supuesto los mismos estudiantes.

En diferentes partes del mundo han considerado este tema como de vital importancia, por ejemplo en las Islas Baleares [i] el objetivo principal en este tema es mejorar el sistema de gestión y planificación de las rutas. Para eso, como primera medida analizan la situación real del sistema, luego proponen nuevas rutas en función de distintos criterios y finalmente desarrollan un aplicativo SIG (Sistema de información geográfica) para que gestione de manera adecuada la demanda educativa del servicio y se logre una disminución en los costos que esta actividad involucra.

El costo que se busca minimizar en la presente investigación es la distancia recorrida por cada vehículo transportador. Al disminuir la distancia total recorrida se está bajando a favor del transportador el combustible consumido, con su evidente impacto en la economía del negocio. En el caso de los usuarios del servicio -padres de familia y estudiantes- se reduce el tiempo de viaje en el sistema y se ofrece un conjunto de rutas que pasa lo más cercanamente posible al lugar de residencia.

Es interesante ver como para resolver un problema de enrutamiento de vehículos de transporte escolar, es necesario involucrar diferentes áreas del conocimiento como: optimización, herramientas informáticas, sistemas de información geográfica, complejidad computacional, entre otras.

El sistema de transporte escolar consiste en el desplazamiento de estudiantes desde su lugar de residencia hasta el plantel educativo y en sentido contrario, desde el plantel educativo hasta su lugar de residencia al terminar la jornada escolar.

El sistema en la vida real está compuesto por vías, cruces, sentidos en las vías, direcciones, vehículos transportadores, etc. Los cuales son necesarios para cumplir con la tarea de movilización.

Es importante tener un buen diseño de rutas escolares para garantizar el cubrimiento de todos los estudiantes, de tal manera que los costos que esta actividad involucra se minimicen. Sin embargo, la tarea de diseñar un sistema óptimo de rutas de transporte

escolar no es sencilla, debido a que para obtener la solución, actualmente es necesario explorar todo el universo de soluciones, requiriendo una capacidad y tiempo de cómputo muy elevados.

No existe un procedimiento que haga que la respuesta se obtenga de manera fácil, por tal razón este tipo de problemas entra en la categoría de problemas de optimización combinatoria, en donde se explora un gran número de soluciones posibles a partir de un algoritmo definido con tal de obtener una muy buena respuesta. La rapidez y la calidad con que se de la respuesta está dado por la eficiencia del algoritmo planteado.

El objetivo del diseño de rutas escolares es dejar de lado su elaboración empírica basada en el vago conocimiento de algunas vías de la ciudad, para que a partir de una malla vial bien definida se pueda aplicar un algoritmo que determine los recorridos más cortos.

Este tipo de problemas se ha definido en la literatura como NP-Duro, a los cuales se les considera como problemas intratables por no existir un algoritmo determinístico que permita obtener una solución exacta en un tiempo razonable.

Los procedimientos que se utilizan actualmente para aproximarse a una buena solución son las heurísticas y las metaheurísticas, las cuales hacen una búsqueda especializada en el espacio de soluciones, sin entrar en una búsqueda exhaustiva. Dentro de las heurísticas y metaheurísticas más usadas están: algoritmo de ahorros, búsqueda local, algoritmo de pétalos, algoritmos genéticos, colonia de hormigas, *simulated annealing* y *tabu search*.

En la presente investigación se plantea realizar inicialmente el tratamiento de la información necesaria para luego implementar un algoritmo que permita diseñar las rutas de transporte escolar. Este trabajo se desarrolla en paralelo con la investigación de Robert Antonio Barraza Gamero [ii], quien a partir de la misma información, implementa un algoritmo diferente (*tabu search*), al final se espera determinar cuál algoritmo arroja la mejor solución en un tiempo razonable. Dentro del documento se encuentran fragmentos que son tomados del trabajo de Robert Barraza con su previa autorización, dichos fragmentos se identifican con un asterisco entre paréntesis (*) en la parte inicial del título. También es preciso resaltar que los fragmentos que fueron redactados en conjunto aparecen señalados con dos asteriscos entre paréntesis (**).

2 Objetivos

2.1 *Objetivo general*

Analizar, diseñar e implementar un procedimiento que permita dar solución al problema de enrutamiento de vehículos de transporte escolar, el cual se considera como un VRP (vehicle routing problem) con restricción de capacidad y distancia (DCVRP). La solución se basa en heurísticas como *simulated annealing*, para optimizar el proceso de búsqueda de una buena solución.

Se busca de igual manera desarrollar un marco de trabajo en el que el manejo de la información no esté condicionado a un caso específico y pueda ser aplicado para todo conjunto de estudiantes con una ubicación clara del colegio y del garaje desde donde salen y llegan los vehículos de transporte. Al final se comparan los resultados obtenidos con el trabajo de Robert Barraza quien implementa la metaheurística *tabú search*.

2.2 *Objetivos específicos*

- Implementar el proceso de geocodificación para localizar geográficamente los estudiantes en la malla vial de Bogotá D.C.
- Implementar un algoritmo de segmentación de estudiantes para afrontar el problema de enrutamiento en conjuntos más pequeños.
- Determinar e implementar los parámetros de entrada de la metaheurística *Simulated annealing* para obtener una buena solución.
- Determinar e implementar un proceso de post-optimización que mejore las rutas obtenidas por la metaheurística.
- Diseñar e implementar un aplicativo que integre los diferentes módulos que dan solución al problema de enrutamiento de vehículos de transporte escolar.

- Hacer una comparación objetiva entre esta investigación y la que se desarrolla en paralelo con la implementación de la metaheurística *tabu search*, implementación que hace Robert Barraza en su tesis de grado.
- Realizar recomendaciones para los trabajos posteriores sobre diseño de rutas de transporte escolar y posible paralelización del aplicativo.

3 Alcance

El problema de enrutamiento de vehículos especialmente de transporte escolar se ha tratado en diferentes investigaciones alrededor del mundo. Lo que se busca en el presente trabajo es abordar el problema de una manera diferente con la implementación de una herramienta que en diferentes partes del proceso optimice la respuesta en busca de una buena solución en un tiempo de cómputo razonable.

El trabajo comprende el problema de enrutamiento de vehículos con capacidad limitada en donde se busca minimizar la distancia total recorrida. Para afrontar dicho problema que por lo general consta de un gran número de estudiantes, se plantea una segmentación inicial con el uso de *K-means* más un acondicionamiento particular para tener en cuenta las limitaciones de capacidad.

Una vez se obtienen los diferentes segmentos se implementa la metaheurística *Simulated Annealing*, en la cual es necesario determinar los mejores parámetros para lograr un buen resultado. En el presente trabajo se muestra el análisis y las conclusiones de mencionado estudio para ser aplicado en cualquier problema de características similares.

Debido a que en la vida real los estudiantes se pueden desplazar cierta distancia desde su lugar de residencia, se incorpora en el presente trabajo un proceso de post-optimización para contemplar el posible desplazamiento de los paraderos de la ruta y así lograr rutas más cortas.

La propuesta implementada está enfocada a trabajar sobre soluciones informáticas de casas desarrolladoras que han generado sistemas de información geográfica. Es el caso de ESRI con su producto ArcGIS, herramienta que permite manipular y obtener información vital para el proyecto, pero que no hace parte del objeto de estudio del presente trabajo.

La presente investigación se desarrolla en paralelo con la de Robert Antonio Barraza Gamero, los dos proyectos parten de los mismos datos usando las mismas metodologías en el tratamiento de la información, pero en la parte intermedia se implementan diferentes metaheurísticas obteniendo resultados de similares características que permiten evaluar de manera objetiva que solución es mejor.

Cada uno de los módulos es completo en términos funcionales, para que se tenga un punto de partida para posteriores investigaciones en el mismo tema o afines.

4 Marco Teórico

En el presente capítulo se exponen los diferentes conceptos sobre los cuales subyace la investigación. Dentro de los temas a tratar está la definición del transporte urbano, la explicación de complejidad computacional contextualizando la investigación en ella. Se presenta también la definición de VRP (*Vehicle routing problem*), diferentes formas de solucionar este tipo de problemas, el estado del arte para el diseño de rutas de transporte escolar, el análisis de las alternativas implementadas en la solución y por último la descripción del problema con su correspondiente definición matemática.

4.1 Transporte urbano

El transporte nace de la necesidad de desplazar mercancías o personas de un lugar a otro, ya sea por seguridad, comodidad o porque las distancias que se necesitan recorrer son demasiado largas. En la sociedad actual, la cual se ha organizado en comunidades bastante grandes como lo son las ciudades, las personas cuentan con dos modalidades de transporte urbano, el privado y el público de pasajeros.

El transporte privado está destinado a servir solamente a sus propietarios. Mientras que el servicio público se enfoca en desplazar personas dentro de la ciudad basándose en el diseño de rutas para cubrir la mayor parte de la malla vial disponible.

4.1.1 Transporte público

El transporte público juega un papel importante dentro de las ciudades, genera un ambiente de permanente convivencia entre la malla vial, vehículos y usuarios, los cuales a lo largo y ancho de las urbes ven como permanentemente se está congestionando, renovando, actualizando y modernizando el sistema. Además del papel tan relevante que tiene el transporte público por estar dentro de las ciudades modernas, tiene la responsabilidad de garantizar una buena movilidad, brindar seguridad, ofrecer costos favorables para el usuario, ser cómodo y sobre todo tener un servicio eficiente.

En nuestro país el transporte público ha estado en las últimas décadas en constante evolución y transformación. Por ejemplo, en la capital de la república pasamos de un transporte público con poco control a mediados del siglo veinte, a la incorporación de una troncal en una de las principales vías de la ciudad. Es el caso de la Avenida Caracas, en donde se buscaba organizar a partir de una serie de paraderos bien definidos las diferentes

rutas que por esta vía transitaban, pero al final por falta de control no fue una solución adecuada.

El siguiente sistema que se implementó a comienzos del siglo veintiuno fue el basado en una serie de paraderos o estaciones con mayor control humano y tecnológico, paradas y rutas bien definidas y una flota de buses articulados que recorren todo el sistema bajo la supervisión y regulación de una sola empresa, a diferencia del modo anterior, el cual se componía de un gran número de empresas separadas sin ninguna interrelación entre ellas.

En otras partes del país el transporte urbano también ha demostrado un constante avance, es el caso de la ciudad de Medellín en donde en los años ochenta se construyó un metro ligero que recorre la ciudad por su eje principal de norte a sur y en la parte central de oriente a occidente en una línea integrada a la principal. Además, recientemente se incorporó a este sistema otra modalidad de transporte, el Metrocable, el cual consta de una serie de teleféricos que desde una estación del metro, cubren varios barrios en las montañas nororientales de la ciudad.

4.1.2 *Transporte escolar*

Un aspecto que influye en el desarrollo de una ciudad y un país es el nivel educativo que alcanzan sus habitantes, debido a esto el número de planteles educativos en una ciudad es considerable y por ende la cantidad de estudiantes que a ellos asisten. Lo anterior hizo que con el tiempo a los estudiantes se les dedicara un tipo de transporte especializado para recogerlos y dejarlos en el lugar de residencia o muy cerca de ella, evitando que recorran caminos muy largos y que se expongan a peligros por la vulnerabilidad que sus tempranas edades representan.

Como en la actualidad las rutas se diseñan de una manera “artesanal” sin tener en cuenta diferentes variables que influyen en el medio, se pretende tener una herramienta que optimice la ruta final para que la empresa transportadora pueda prestar un mejor servicio basándose en ciertas restricciones que emulan mejor el comportamiento vial en un computador, es difícil tener presente todas las variables pero se logra una muy buena respuesta.

El transporte escolar se convierte entonces en un actor más dentro del transporte urbano, llamando la atención de las entidades reguladoras, las cuales mediante normas, leyes y reglamentos buscan controlar la movilidad de estos vehículos por las calles, para así lograr que se brinde un servicio seguro, cómodo y eficiente a los estudiantes.

4.1.3 Entidades reguladoras

4.1.3.1 Secretaría de tránsito y transporte de Bogotá D.C.

Esta corporación se encarga de regular y controlar el estado del servicio de tránsito y transporte terrestre automotor en la ciudad, diseña y ejecuta políticas para mantener la seguridad y correcto funcionamiento de la red vial, se ocupa de los trámites del parque automotor, tales como: licencia de conducción, certificado de tradición, traspaso, y radicación de cuenta, entre otros.

En cuanto al transporte escolar, la Secretaría de Tránsito y Transporte se encarga de fijar los lineamientos para ser cumplidos por las compañías prestadoras de este servicio y velar para que la normatividad nacional también sea acogida en el Distrito. Dentro de la documentación y los requerimientos mecánicos exigidos para la movilidad escolar, se encuentran: portar los documentos de ley, tener vigente la revisión técnico mecánica, tener pintados los distintivos y portar algún sistema de comunicación, entre otros.

4.1.3.2 Secretaria de educación de Bogotá D.C.

La secretaría de educación del Distrito Capital es la entidad encargada de velar por que en su jurisdicción se brinde una buena calidad de educación preescolar, básica y media. Se ocupa de establecer políticas, planes y programas de acuerdo a la regulación establecida por el Ministerio de Educación Nacional, organizar y controlar la evaluación de calidad de la educación.

En lo que respecta al transporte escolar, se encarga de velar porque los estudiantes de planteles públicos de escasos recursos puedan acceder a un subsidio de transporte, dando un incentivo económico para que los estudiantes de estratos socioeconómicos bajos no deserten de la educación. Además, la Secretaría cuenta con un sistema de rutas para aquellos estudiantes que posean alguna discapacidad o se encuentren a una distancia considerable (igual o superior a 2Km) del plantel educativo.

4.2 Algoritmos

En el mundo computacional se conoce como algoritmo a la serie de pasos que se siguen para resolver un problema, esta serie de pasos deben conducir en cualquier instancia del algoritmo a una solución correcta, completa y precisa. De igual manera en informática un algoritmo debe ser eficiente, lo cual implica hacer buen uso de la memoria en donde se almacenan los datos y el tiempo que toma finalizar con éxito el proceso.

4.2.1 Algoritmos deterministas

Los algoritmos deterministas son aquellos en los que cada paso está determinado o definido de manera única. Lo anterior se basa en que al estar en una fase determinada del sistema, se conocen las variables del entorno y su comportamiento frente a cambios en el ambiente, por tal razón, se puede predecir sin ningún error la siguiente fase del sistema.

4.2.2 Algoritmos no deterministas

Este tipo de algoritmos son aquellos en los que para decidir el siguiente estado del sistema hay que explorar primero un conjunto de posibilidades para ver cómo reacciona el entorno. Una vez agotadas se escoge una solución correcta siempre y cuando ésta exista.

4.3 Complejidad

En computación, la complejidad de un algoritmo se mide en espacio y en tiempo, siendo esta última la más común de las métricas, pero el tiempo no es cuantos segundos el algoritmo consume, sino que hace referencia al tiempo computacional o número de operaciones que se deben hacer dentro del algoritmo, siendo una función que acota el tiempo máximo de ejecución en función del número de datos del problema, esto hace que el cálculo no sea dependiente de la máquina que se utiliza. La Tabla 4-1 muestra la notación utilizada en diferentes órdenes de complejidad computacional.

Tabla 4-1 Notación complejidad

$O(1)$	Orden constante
$O(\log n)$	Orden logarítmico
$O(n)$	Orden lineal
$O(n \log n)$	Orden $N \log N$
$O(n^2)$	Orden cuadrático
$O(n^a)$	Orden polinomial ($a > 2$)
$O(a^n)$	Orden exponencial ($a > 2$)
$O(n!)$	Orden factorial

En los siguientes numerales se presentan algunas definiciones claves sobre las diferentes clases de complejidad computacional.

4.3.1 Complejidad L

La clase de complejidad L es el conjunto de problemas que se encuentran dentro del espacio logarítmico determinista. Este tipo de problemas puede resolverse en un periodo de tiempo $\log(n)$, en donde n indica el tamaño de la entrada.

4.3.2 Complejidad P

Se denomina así a los algoritmos deterministas cuya complejidad se representa por un polinomio $p(n)$, este tipo de problemas se consideran tratables, ya que en la práctica pueden ser solucionados.

4.3.3 Complejidad NP

Dentro de esta categoría de complejidad se encuentran los algoritmos intratables. Su método de solución es no determinista, en el cual se utilizan heurísticas para encontrar posibles soluciones que van siendo evaluadas para su posterior aprobación o rechazo a un ritmo polinómico.

4.3.4 Complejidad NP-Completo

Este tipo de complejidad se aplica a problemas de tipo NP con una extrema complejidad, tienen una característica entre sí, y es que son problemas equivalentes, de tal forma que al encontrar una solución polinomial para alguno de esta clase, se está encontrando una solución aplicable a todos ellos.

Como ejemplo de un problema NP-completo está el problema de la suma de subconjuntos, el cual dice que a partir de un conjunto S de números enteros, ¿existe un subconjunto no vacío de S cuyos elementos sumen cero? La forma de solucionar este problema es explorando todos los $2^n - 1$ subconjuntos posibles hasta encontrar que se cumple con la condición.

4.3.5 Complejidad NP-Hard

Se agrupan así los problemas de decisión, los cuales son por lo menos tan difíciles como un problema de complejidad NP-Completo. Tienen como característica que la respuesta no será encontrada en tiempo polinomial, a menos que $P=NP$.

4.4 Análisis del problema VRP

4.4.1 Descripción general del problema VRP

Se conoce con el nombre genérico de VRP (*Vehicle Routing Problem*) a una gran variedad de problemas que buscan determinar un conjunto de rutas preferiblemente eficientes, las cuales son recorridas por una flota de vehículos que tienen como tarea, repartir o recoger mercancía o personas que están distribuidos geográficamente. Este tipo de problemas se puede presentar a diferentes escalas, por ejemplo: en vehículos industriales dentro de una fábrica, distribución de mercancía dentro de una misma ciudad, entre varias ciudades o hasta negocios que involucran enrutamiento a nivel mundial.

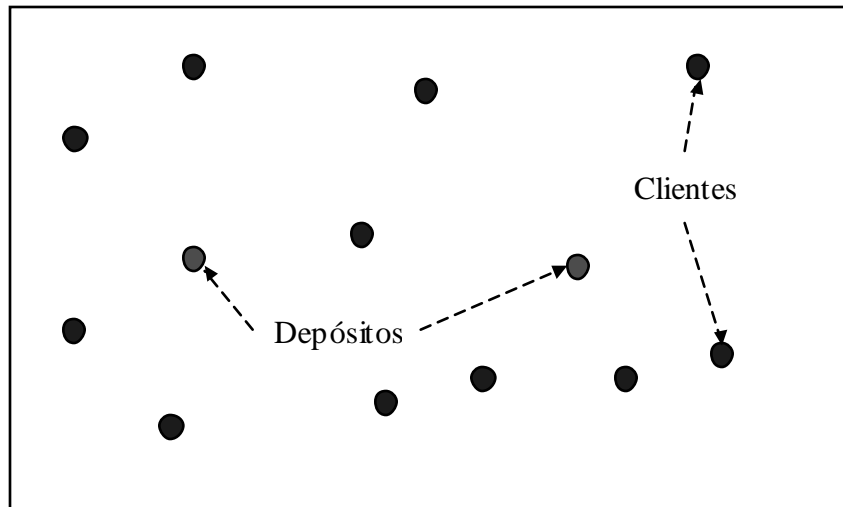


Figura 4-1 Esquema típico de datos de entrada a VRP

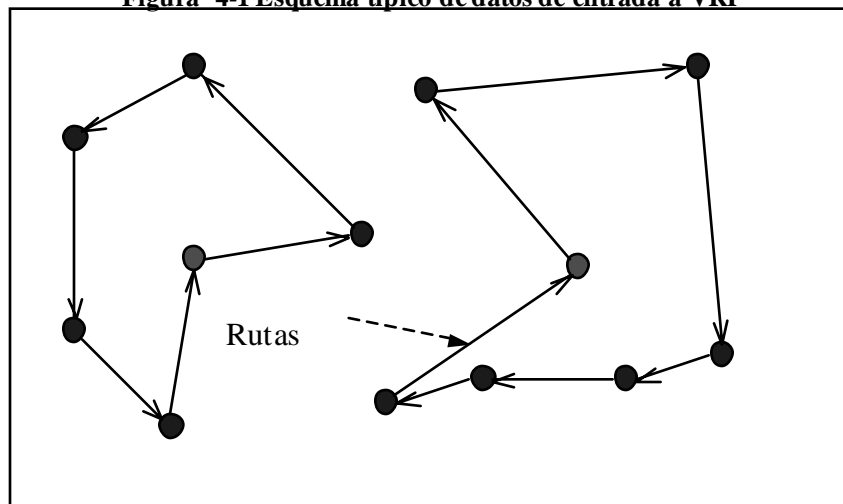


Figura 4-2 Esquema típico de datos de salida de VRP

En la figura Figura 4-1 se aprecian los datos que servirían de entrada a un problema VRP, en este caso dos depósitos y un conjunto de clientes esparcidos sobre un área determinada, por lo general se conocen las coordenadas de dichos puntos. De igual forma en la Figura 4-2 se puede ver la salida de estos problemas, el ejemplo muestra dos rutas que salen de cada depósito, visita a los clientes y regresa al punto de partida.

Los problemas de enrutamiento de vehículos pertenecen a la gama de los problemas NP-Hard en complejidad computacional. La razón es el gran esfuerzo de cómputo que se necesita para conseguir una solución óptima o cercana a ella: el esfuerzo se va incrementando de manera exponencial a medida que el número de clientes involucrados aumenta. Actualmente se utilizan métodos de aproximación para encontrar una buena respuesta al problema y que no involucre una gran cantidad de tiempo.

En el mundo moderno la necesidad de analizar, estudiar y solucionar este tipo de problemas radica en que cada vez se busca recorrer menores distancias, gastar menos tiempo o circular en rutas de manera más eficiente. El objetivo final es reducir costos y así brindar al cliente productos a menor costo y en lapsos de tiempo apropiados.

4.4.2 Elementos que conforman el problema VRP

4.4.2.1 Clientes

Los clientes son los que van a ser atendidos por los vehículos que participan en el problema de enrutamiento, dando o recibiendo, bienes o servicios. Los bienes pueden ser entregados por el vehículo transportador al cliente, por el cliente al vehículo o un intercambio de mercancías entre ambos. El objetivo es satisfacer la demanda del cliente, en donde los bienes entregados por lo general ocupan un espacio determinado dentro del vehículo, generando la necesidad de diseñar bien la distribución y recaudo de mercancía para que se cumpla con la finalidad del servicio, recurriendo en algunas ocasiones a varias rutas para satisfacer el mismo cliente.

4.4.2.2 Depósitos

Los depósitos en el problema de enrutamiento se pueden encontrar al comienzo de la ruta, al final o en algunos casos en lugares intermedios de las rutas dependiendo la naturaleza del problema. Con base en las características del problema, el depósito puede tener su propia flota de vehículos, puede tener una capacidad de producción limitada, prestar el servicio de carga o descarga en determinados momentos de tiempo o tener una capacidad determinada para el parqueo de los vehículos.

4.4.2.3 Vehículos

Los vehículos tienen una capacidad limitada para transportar objetos, la cual puede tener cómo medida el volumen o el peso. Se asocia también un costo por el hecho de utilizar el vehículo y otro por la longitud que se recorre a través de la ruta. En la mayoría de los problemas se busca minimizar la cantidad de vehículos utilizados, pero hay problemas en donde lo importante es optimizar sus recorridos, siendo el número de vehículos un parámetro de entrada ya conocido. Cuando la capacidad de todos los vehículos es conocida y son iguales, se dice que la flota es homogénea, mientras que en el caso contrario, la flota es heterogénea.

4.4.3 Parámetros del problema VRP

Los problemas de enrutamiento tienen varios parámetros que deben ser tenidos en cuenta en el momento de representar un problema de la vida real en un modelo para ser optimizado. A continuación se mencionan los parámetros más relevantes.

- Flota de vehículos: Cuando la capacidad de todos los vehículos es igual, se dice que la flota es homogénea, de lo contrario es heterogénea.
- Forma del bien transportado: El objeto que se transporta puede tener dimensiones conocidas y constantes, para catalogarse como bien homogéneo o heterogéneo.
- Depósito: Las características del depósito también deben ser tenidas en cuenta, la funcionalidad del depósito puede ser para almacenar la mercancía que llega, para distribuir productos generados en él y que solo salen del depósito o puede ser una combinación de ambas.
- Número de depósitos: El número de depósitos puede ser desde uno solo en todo el problema hasta n . En el caso del presente trabajo, se cuenta con dos depósitos, uno inicial desde donde salen los vehículos y otro el de llegada.
- Tipo de ruta: La ruta puede ser circular, el punto de partida es el mismo de llegada, o abierta, ambos puntos son diferentes.
- Función objetivo: Al diseñarse un problema de enrutamiento, se debe tener claro qué se busca minimizar. Dentro de ellos está el tiempo, la distancia, el dinero y el número de rutas entre otros.

- Número de vehículos: Otro factor que en algunos problemas es relevante minimizar es el número de vehículos.
- Ventanas de tiempo: Hay problemas en los que el cliente debe ser atendido durante un periodo de tiempo específico, a esto se le considera como ventana de tiempo, mientras que en otros tipos de problemas, es suficiente solo con llegar a la parada detenerse un momento y luego continuar.
- Clientes: Los usuarios del servicio es necesario catalogarlos dentro del modelo para ver si son nodos dentro de la ruta que están estáticos para cualquier instante, o van a variar en ubicación y en número en diferentes estados.

4.4.4 Variantes del problema VRP

Con base en las diferentes variables del problema, en la literatura [iii] se han definido algunas variantes clásicas al problema VRP, dependiendo el enfoque del problema que se quiera analizar.

VRP con capacidad definida (CVRP)

Son aquellos problemas en los que se cuenta con una flota homogénea de vehículos con la misma capacidad, la cual se encarga de satisfacer las demandas de todos los clientes partiendo de un depósito.

El objetivo es minimizar el tiempo total de viaje y la flota de vehículos sin exceder su capacidad.

VRP con restricción de distancia y capacidad definida (DCVRP)

Son aquellos problemas en los que se cuenta con una flota de vehículos, la cual se encarga de satisfacer las demandas de todos los clientes partiendo de un depósito.

El objetivo es minimizar el tiempo total de viaje y la flota de vehículos sin exceder su capacidad y la distancia total recorrida.

VRP con múltiples depósitos (MDVRP)

Se parte de un conjunto de usuarios y varios depósitos desde los cuales salen los vehículos. Cada vehículo asignado a un depósito, sale de él, recorre a los clientes y luego regresa al mismo depósito.

El objetivo es minimizar la flota de vehículos, el tiempo total de viaje y satisfacer la demanda de todos los clientes.

VRP periódico (PVRP)

Este problema extiende la concepción del VRP normal, en el cual se supone que todos los clientes son atendidos dentro de un solo día de recorrido, mientras que en el PVRP, la duración del recorrido puede durar más de un día, lo cual implica que el vehículo no regrese el mismo día al depósito.

El objetivo de esta variante es minimizar el número total de vehículos y el tiempo total de viaje.

VRP con entrega dividida (SDVRP)

Hay casos en que la capacidad que solicita el cliente no puede ser atendida por un solo vehículo, es por eso que él debe ser atendido por diferentes vehículos para completar la demanda pedida.

El objetivo es minimizar el tiempo de viaje y la cantidad de vehículos usados para satisfacer la demanda de todos los clientes.

VRP recogiendo y entregando (VRPPD)

Este problema involucra que los vehículos transportadores dejan mercancía pero también recogen de los usuarios. Esta restricción incrementa la complejidad del problema debido a que se tiene que mantener siempre capacidad suficiente para satisfacer tanto la demanda de los productos entregados a los clientes, como la entregada por ellos a los vehículos y que debe ser transportada de regreso al depósito.

El objetivo es minimizar el número de vehículos, el tiempo de viaje y mantener la restricción de capacidad explicada anteriormente.

VRP con ventana de tiempo (CRPTW)

Hay problemas en los que cada cliente tiene dos instantes de tiempo, uno inicial y otro final, dentro de los cuales debe ser atendido, si llega el vehículo con anterioridad al momento inicial, éste debe esperar a que se cumpla la hora para poder atenderlo, si llega dentro del intervalo lo atiende, pero si llega después del tiempo final, el cliente no puede ser atendido.

El objetivo es minimizar el tiempo total de viaje, la flota de vehículos y el tiempo de espera para satisfacer a los clientes.

4.5 Técnicas para solucionar VRP

4.5.1 Métodos exactos

Cuando el tamaño del problema es pequeño, se puede resolver mediante métodos exactos que se resuelven en un tiempo computacional aceptable. Esto se puede lograr mediante relajaciones del problema y la utilización de algoritmos de ramificación como el *Branch and Bound* o de programación dinámica.

4.5.2 Heurísticas

El origen de la palabra heurística es similar al de la palabra eureka, puede usarse como sustantivo haciendo referencia al arte de descubrir o se puede usar como adjetivo, refiriéndose a cosas más específicas, como reglas heurísticas, estrategias heurísticas o conclusiones heurísticas.

En algoritmos de programación las heurísticas son métodos que hacen una exploración no exhaustiva del espacio de soluciones, llegando a respuestas aceptables en tiempos de procesamiento moderados. Dadas estas características es probable que la respuesta haga parte de un mínimo local y se encuentre lejos del óptimo.

Dentro de las principales heurísticas se encuentran:

- Algoritmo de ahorros
- Inserción secuencial de Mole & Jamenson
- Inserción en paralelo de Christofides, Mingozzi y Toth
- Heurística de barrido o sweep
- Heurística de asignación generalizada de Fisher y Jaikumar
- Heurística de localización de Bramel y Simchi-Levi
- Método rutear primero - asignar después
- Algoritmos de pétalos
- Algoritmos de búsqueda local
 - Operador λ -intercambio
 - Algoritmo de Lin-Kernigham
 - Operador Or-opt
 - Operadores de Van Breedam

4.5.3 *Metaheurísticas*

Las metaheurísticas son procedimientos más especializados que las heurísticas clásicas, pues realizan una búsqueda más amplia dentro del espacio de soluciones y genera mejores respuestas que las obtenidas con las heurísticas. Para hacer una mejor exploración del universo de soluciones, es necesario sacrificar tiempo, pero aún así éste sigue siendo considerablemente menor que el empleado por las soluciones exactas.

Las metaheurísticas generalmente se usan para resolver problemas de optimización combinatoria, en donde lo que se busca es encontrar una configuración discreta que

minimice o maximice una función objetivo que es definida por el sujeto que implementa la metaheurística. La existencia de un determinado número de configuraciones construye el espacio de soluciones, sobre el cual se va a hacer la búsqueda especializada. Dentro de las metaheurísticas más utilizadas está la colonia de hormigas, algoritmos genéticos, *tabú search*, *simulated annealing*, entre otras.

Simulated annealing es un algoritmo que nace a mediados de los años ochenta, da una buena aproximación al óptimo global debido a su amplia búsqueda por el espacio de soluciones. Su nombre y metodología vienen de la analogía con el proceso de enfriamiento de metales y cristales, obteniendo buenas configuraciones internas a medida que la temperatura desciende.

Tabu search generaliza la búsqueda local para no explorar siempre en dirección de mejores soluciones, llevando el algoritmo a salir de mínimos locales. Al igual que otras metaheurísticas, permite aceptar movimientos a soluciones de menor calidad respecto a las ya encontradas, esto se logra utilizando una lista que almacena una cantidad determinada de movimientos, los cuales son excluidos por considerarse que conllevan a mínimos locales.

Los algoritmos genéticos se basan en el concepto de la evolución natural de los seres vivos para resolver problemas de optimización. El proceso de búsqueda de una buena solución arranca con una población inicial, la cual representa diferentes estados que pertenecen al espacio de soluciones, la siguiente generación de individuos se obtiene a partir de combinaciones, y alteraciones de los padres, emulando el proceso que se da entre los seres vivos en la realidad, la selección, el cruce y la mutación.

4.6 Análisis de alternativas

4.6.1 Clarke & Wright

Es uno de los algoritmos más utilizados para solucionar problemas de enrutamiento de vehículos. Se basa en el ahorro que puede tener unir dos rutas diferentes, una $(0, \dots, i, 0)$ con $(0, j, \dots, 0)$ como se muestra en la Figura 4-3, en donde se obtiene la ruta $(0, \dots, i, j, \dots, 0)$ con un ahorro (s):

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

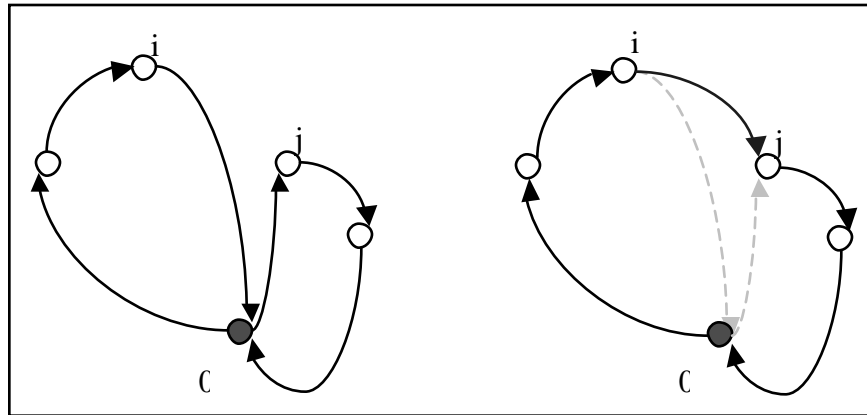


Figura 4-3 Esquema Clarke & Wright.

Se considera un ahorro el hecho de que en la solución de la derecha no se van a recorrer ni el arco $(i,0)$ ni el $(0, j)$, pero si se agrega el (i, j) .

A continuación se muestra el algoritmo implementado:

Paso 1 (inicialización). Para cada cliente i construir la ruta $(0, i, 0)$.

Paso 2 (cálculo de ahorros). Calcular s_{ij} para cada par de clientes i y j .

Paso 3 (mejor unión). Sea $s_{i^*j^*} = \max s_{ij}$, donde el máximo se toma entre los ahorros que no han sido considerados aún. Sean r_{i^*} y r_{j^*} las rutas que contienen a los clientes i^* y j^* respectivamente. Si i^* es el último cliente de r_{i^*} y j^* es el primer cliente de r_{j^*} y la combinación de r_{i^*} y r_{j^*} es factible, combinarlas. Eliminar $s_{i^*j^*}$ de futuras consideraciones. Si quedan ahorros por examinar ir al paso 3, de lo contrario terminar.

Este algoritmo se utiliza en la presente investigación para generar la ruta semilla con que van a trabajar las metaheurísticas.

4.6.2 (*)*Tabu search*

Heurística propuesta originalmente por Fred Glover en el año de 1986, en varios casos las soluciones muy cercanas a las óptimas, sino eran las mejores. Este éxito con la búsqueda tabú (TS de ahora en adelante) aumentó su popularidad vertiginosamente en los últimos años. Uno de los problemas ejemplo más utilizados para mostrar las bondades de este algoritmo es el ya mencionado CVRP.

Algo que caracteriza esta metaheurística de las demás y la no utilización de técnicas aleatorias, optando por el uso de “técnicas de memoria adaptativa y de estrategias especiales de resolución de problemas” [iv]

“La Filosofía de la búsqueda tabú es derivar y explotar una colección de estrategias inteligentes para la resolución de problemas, basadas en procedimientos implícitos y explícitos de aprendizaje” [v].

Grosso modo la búsqueda tabú consiste en que:

Se tiene un conjunto de elementos que al estar ordenado de distinta forma altera el resultado total, por lo que se quiere encontrar una ordenación de tal forma que el conjunto dé un resultado óptimo. Para ello, se halla una solución inicial. Acto seguido se hallan los distintos intercambios posibles, de las todas las posiciones de los elementos que constituyen el conjunto inicial, dichos intercambios pueden realizarse de distinta forma, la utilización de uno y otro alterará el resultado final. Cada uno de los posibles intercambios se organizan de mayor a menor según mejoren o empeoren el resultado final. Se aplica el mejor resultado, dicho intercambio se almacena en una lista conocida como tabú, para que dentro de los próximos n (donde n es el tamaño de la lista tabú, que en realidad actúa más o menos como una cola) iteraciones no se vuelva aplicar dicho intercambio, así sea el mejor, en cuyo caso se tomará el segundo de la lista, que de también estar en la lista tabú, se seguirá con el siguiente, y así sucesivamente. Existe un caso en el que se viola la no aplicación de un movimiento que está tabú, y es cuando ese movimiento genera un resultado total, mejor que todos los encontrados hasta ese momento, dicha violación se conoce como Criterio de Aspiración. Con la lista tabú, se busca no caer en óptimos locales y explorar el más el mundo de la solución, lo que se conoce como diversificación.

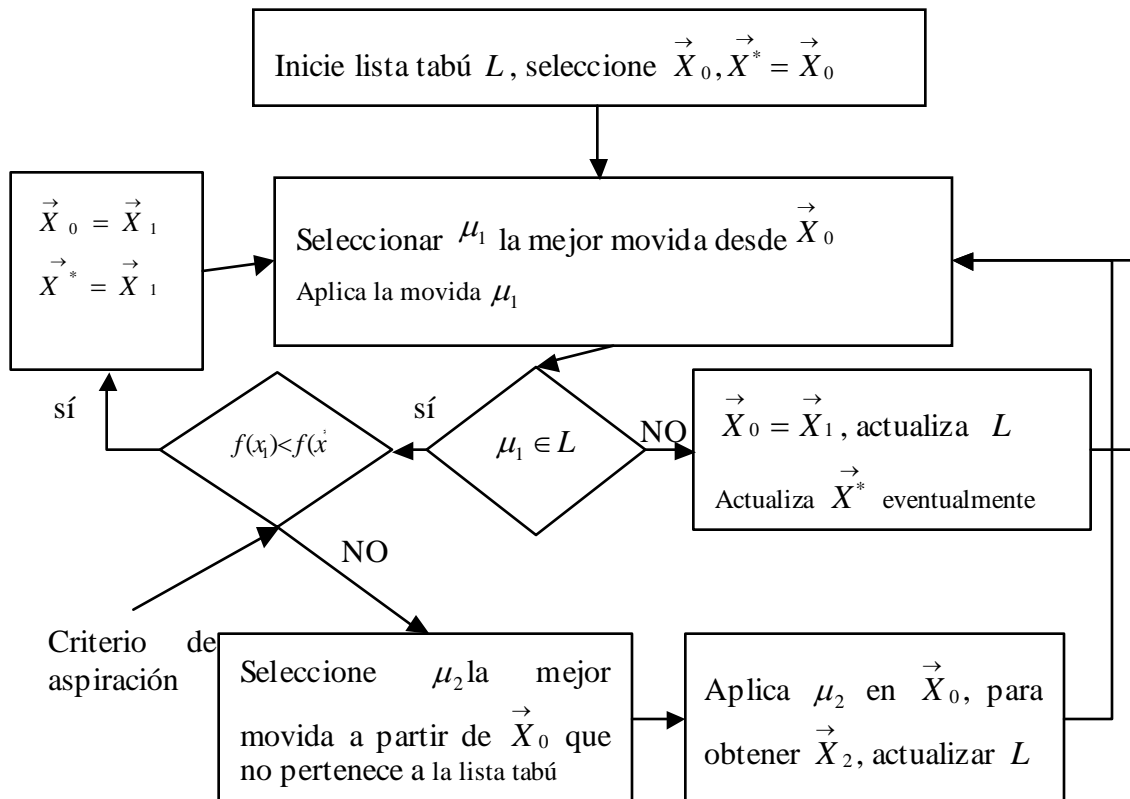


Figura 4-4 Algoritmo Búsqueda Tabú.

4.6.2.1 Simulated annealing

Simulated annealing (recocido simulado) nació como metaheurística a comienzos de los años ochenta Kirkpatrick *et al.* [vi], y desde entonces se ha convertido en un algoritmo relevante dentro del conjunto de los que se ocupan de solucionar problemas de optimización combinatoria. Su principal acierto es especializar la búsqueda local para que por medio de exploraciones hacia respuestas peores que las ya encontradas, no se restrinja permanentemente a sectores que conducen a mínimos locales.

En la naturaleza, cuando un metal o cristal se calienta a altas temperaturas, sus átomos se encuentran en estado de gran excitación, moviéndose de un lado para otro con total libertad. Esto permite que se pueda caer en un estado en el que la configuración de los enlaces entre los átomos no es la mejor, pero por sus constantes movimientos, en un siguiente estado puede llegar a colocarse los átomos en lugares en que se obtiene una mejor configuración. En esta etapa de temperaturas muy altas el material se encuentra en fase líquida.

A medida que la temperatura disminuye los átomos van perdiendo energía y por consiguiente, la posibilidad de moverse con facilidad a cualquier lado también va

disminuyendo. Cuando finalmente se enfría por completo, ningún átomo se puede mover dejando el material en fase sólida.

Cuando el enfriamiento se hace de manera paulatina, los átomos se van acomodando de manera que la configuración final es más resistente y de mejor calidad. Por el contrario, cuando el enfriamiento se hace precipitadamente, entonces la configuración puede dejar zonas con enlaces más débiles que otras.

En 1953 Metropolis et al., lograron en sus trabajos de termodinámica estadística modelar el comportamiento que tienen las partículas que conforman un metal en su proceso de enfriamiento, convergiendo desde estados de alta energía hasta el total enfriamiento.

Según la termodinámica se dice que a una temperatura (t) la probabilidad de un incremento energético (δ) se puede obtener por:

$$P_t(\delta) = e^{-\frac{\delta}{k_B t}}$$
$$\delta = f(s) - f(S_0)$$

En donde (k_b) es la constante física de Boltzman, (s) es el último estado obtenido en el enfriamiento, (S_0) es el penúltimo estado al que llegó el metal, (t) es la temperatura.

Con base en la descripción del fenómeno físico, se hace un símil con los problemas de optimización combinatoria. En donde lo que se busca es encontrar una combinación del problema que minimice una función objetivo. En el caso de enrutamiento de vehículos se busca generar dentro del algoritmo un buen número de combinaciones de paradas (s) para encontrar una ruta de bajo costo.

La generación de las diferentes combinaciones está regida por la temperatura (t) del modelo físico, la cual es un mecanismo de control para aceptar o rechazar buenas y malas soluciones (vecinos) a medida que el algoritmo avanza y la temperatura desciende. Si por ejemplo en un momento dado se tiene que la mejor solución encontrada hasta el momento (S_0) es 500, con una nueva ruta cuyo costo (s) es 300 (obtenida por el intercambio de dos paradas de la ruta original) y temperatura de 500, el algoritmo aprueba el movimiento a la nueva solución que tiene menor costo.

Pero si por el contrario la nueva solución tiene un costo de 600, el delta de energía (δ) es de 100, la probabilidad de que se haga el movimiento es de 0.81.

$$0.81 = e^{-\frac{(500-600)}{50}}$$

A medida que el algoritmo avanza la temperatura descende, si para el mismo ejemplo se tiene una temperatura de 50, la mejor solución en (S_0) es 500 y la nueva solución (s) tiene un costo de 600. La probabilidad de que se haga este movimiento disminuye a 0.13, reduciendo la posibilidad de aceptar el movimiento a esa mala solución.

$$0.13 = e^{-\frac{(500-600)}{50}}$$

En los ejemplos de aplicación de *simulated annealing* la constante de Boltzman k_b , carece de significado, por lo que no es tomada en cuenta.

4.6.2.2 Criterios de modelación

Es indispensable definir ciertos criterios que gobiernan la implementación del algoritmo, a continuación se definen y explican los que tienen mayor incidencia sobre los problemas de optimización combinatoria.

Definición de la temperatura inicial

Es importante saber la temperatura a la cual va a empezar a trabajar el algoritmo. Se recomienda tener valores altos para permitir una amplia exploración de soluciones y no sesgar la respuesta. La determinación de dicho valor puede ser estimado con base en el análisis de la diferencia en costos entre una solución y otra [vii].

Determinación de la solución inicial

El algoritmo debe partir de una solución inicial, la cual puede ser encontrada de manera aleatoria o para refinar el proceso, se puede implementar una heurística para obtener una buena solución.

Determinación del esquema de enfriamiento

A medida que avanza el algoritmo *simulated annealing*, se debe ir disminuyendo la temperatura, ese decremento puede tener una configuración particular que se acomoda

dependiendo la naturaleza del problema. Las formas más comunes de dicho esquema se muestran en la Figura 4-5.

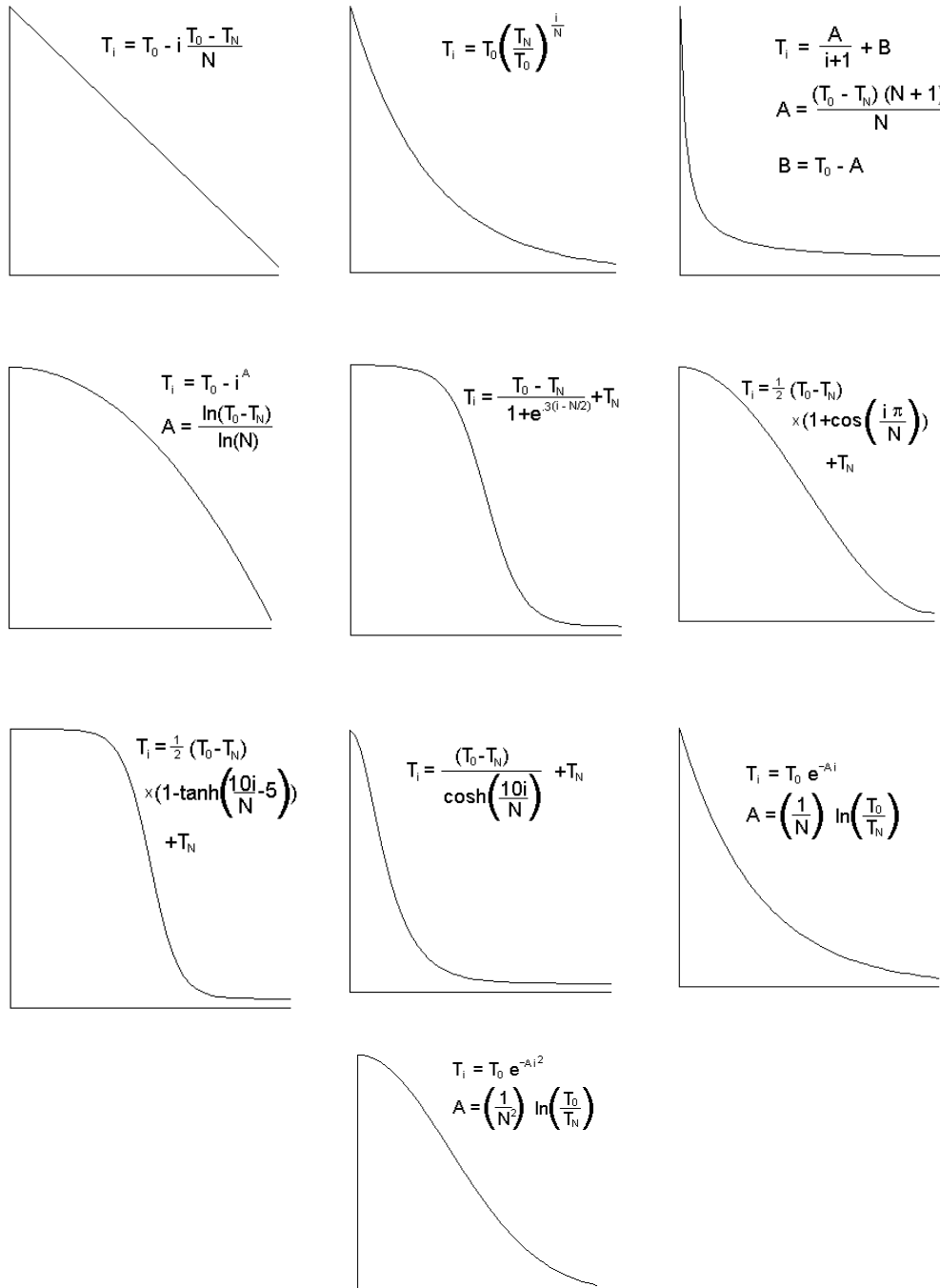


Figura 4-5 Esquemas de enfriamiento.

En donde:

- i indica la iteración en la que se encuentra.
- T_i indica la temperatura en la iteración i .
- T_o indica la temperatura al comienzo del proceso.
- N indica el número total de iteraciones.
- T_N indica la temperatura esperada al finalizar el proceso.

Velocidad de enfriamiento

Este valor está determinado por el número de simulaciones que realiza el algoritmo para conseguir que el modelo llegue a una temperatura muy baja. Como ya se ha mencionado anteriormente, entre más rápido se enfríe el sistema, la respuesta es de menor calidad que la encontrada en periodos mas largos de simulación.

Número de búsquedas de vecinos

El número de iteraciones que se realiza cada vez que se produce un cambio de temperatura, es el número de búsquedas de vecinos que hace para someter a la función de probabilidad que aprueba o rechaza el cambio el nuevo estado.

Forma de determinar los vecinos

A medida que las iteraciones van pasando, se deben encontrar vecinos de la última solución. Para esto, se han determinado diferentes métodos de búsqueda que en su gran mayoría son aleatorios. El objetivo es mezclar y alterar la solución actual para encontrar una nueva opción dentro del espacio de soluciones. Algunas de las técnicas más usadas son: *SWAP*, *RE-INSERT* y *RE-LINK*.

Valor final de la temperatura

Este criterio determina la condición de parada del algoritmo, se busca que la temperatura llegue a cero. Pero dependiendo de la naturaleza del problema, se define si es obligatorio

que se detenga la búsqueda cuando la temperatura llega a cero o simplemente que se consiga un valor muy pequeño.

4.6.2.3 Algoritmo

Antes de ejecutar el algoritmo de recocido simulado, se debe contar con la solución inicial (S_0), la temperatura a la cual parte el proceso (t_0), el costo de la solución inicial $f(S_0)$, el número de iteraciones de Montecarlo ($numIteracionesMT$) y haber definido el esquema de enfriamiento $\alpha(t)$.

```

INICIO ciclo externo
  INICIO ciclo interno
    Seleccionar vecino  $s$ 
    Calcular el costo de  $s$   $f(s)$ 
    Calcular  $\delta = f(s) - f(S_0)$ 
    SI  $\delta < 0$  ENTONCES  $S_0 = s$ 
    SINO
       $u = aleatoric(0,1)$ 
      SI  $u < e^{-\frac{\delta}{t}}$  ENTONCES  $S_0 = s$ 
    FIN
  SINO
    MIENTRAS  $numIteracionesMT > 0$ 
       $t = \alpha(t)$ 
    HASTA QUE  $t = 0$ 
  
```

Figura 4-6 Algoritmo Simulated Annealing

En la literatura se encuentra que esta metaheurística al igual que la de *tabu search* son las que mejores resultados han demostrado dar para el problema de enrutamiento de vehículos. Esto se debe a su amplia búsqueda por el espacio de soluciones y a la manera como avanza el algoritmo tratando de no caer en mínimos locales.

La Figura 4-6 muestra el algoritmo de la metaheurística *simulated annealing*. Su implementación es relativamente sencilla, pero lo que en realidad hace que sea exitoso es la calidad de los criterios de modelación, los cuales se obtienen luego de probar varios de ellos con el problema de optimización que se desea resolver.

5 Descripción del problema

En el presente capítulo se describe el problema analizado en la presente investigación, su estado del arte, la definición de los términos utilizados en el mismo, también se contextualiza dentro del VRP y su complejidad NP-Hard, y por último la definición matemática del problema.

5.1 Problema de enrutamiento de vehículos de transporte escolar

Los planteles educativos prestan un servicio de vital importancia para la sociedad, formando y educando los niños y jóvenes que serán el porvenir de la comunidad. A los colegios asisten alumnos que se encuentran en muy tierna edad, entre los 5 y 17 años aproximadamente, lo que les hace vulnerables a los riesgos que presenta la calle.

En ciudades como Bogotá D.C. los colegios están en la mayoría de los casos a varias cuadras de distancia del lugar de residencia de los estudiantes. Si las condiciones económicas de la familia del alumno lo permiten o si se cuenta con algún apoyo de las autoridades, el estudiante por comodidad y seguridad puede hacer uso de transporte especializado en el desplazamiento de alumnos.

El diseño de rutas se hace con base en el conocimiento general de la malla vial de la ciudad, sin entrar en ningún detalle técnico ni de optimización para obtener mejores diseños. En algunos planteles por ejemplo, se define a priori un conjunto de rutas de tal manera que abarquen distintas zonas alrededor del colegio, una vez definidas las rutas, los estudiantes se suscriben a la que pasa más cerca de su lugar de residencia, generando en la mayoría de los casos rutas con recorridos innecesarios, costosos y que en conjunto reducen el desempeño del servicio.

Otra manera de generar rutas es a partir de la ubicación de los estudiantes, se determina el orden en que se van a recoger o dejar y el vehículo circula por las vías principales más cercanas. Es otra manera “artesanal” de diseñar rutas, tampoco tiene en cuenta distancias ni la distribución espacial de los estudiantes.

Adicionalmente al problema planteado, se ha encontrado otra variación que se puede ingresar al sistema en el momento de su modelación. Hay estudiantes que pueden desplazarse desde su lugar de vivienda a algunas cuadras a la redonda para esperar el vehículo transportador, facilitando el recorrido de la ruta. Pero hay casos en los que por

limitaciones físicas, seguridad o edad, el estudiante no puede desplazarse, entonces la ruta debe pasar en frente de su casa. Este servicio se conoce como servicio “puerta a puerta”.

Para el caso en que el estudiante puede caminar algunas cuadras, el problema debe contemplar no solo el punto de residencia sino que se deben determinar algunos puntos a la redonda para ser tenidos en cuenta en el algoritmo de solución. Más adelante se explica con detenimiento este planteamiento y la forma de solucionarlo.

Con respecto a la información con que se trabaja en la presente investigación se tomó para el diseño de las rutas una malla vial real, la de la ciudad de Bogotá D.C., la cual cuenta con 72.135 cruces y 120.630 arcos, con las longitudes de cada uno de sus tramos y la dirección de los mismos. La información de los estudiantes consta de su dirección catastral, la cual fue suministrada por un colegio que por razones de confidencialidad no se coloca en el presente documento.

Para colocar la anterior información (malla vial – dirección estudiantes) en un mismo contexto, es necesario realizar una geocodificación, para poder ubicar geográficamente cada uno de los estudiantes y poder implementar los algoritmos de optimización que se exponen en capítulos posteriores.

El nombre del aplicativo realizado en la presente investigación es Hariti, inspirado en la mitología hindú:

[viii]“... Hariti es una diosa budista protectora de los niños. De acuerdo con la mitología hindú, Hariti (en sánscrito, "la arrebatadora") era originalmente una diablesa con forma de mujer y tenía 500 hijos. Disfrutaba comiéndose a los hijos de otras mujeres, así que Buda escondió al más pequeño de sus hijos bajo un plato de limosna vacío. Hariti sufrió mucho buscándole y entonces Buda se lo devolvió y le dijo: "Si tú has sufrido tanto por perder a 1 de tus 500 hijos, ¿que no sufrirá una madre a la que le has quitado su único hijo?" Desde entonces, Hariti pasó a ser la diosa de los niños, extendiéndose su culto hasta China y Japón donde tuvo gran popularidad”.

5.2 Soluciones conocidas al problema de transporte escolar

A continuación se presenta un trabajo realizado en España sobre enrutamiento de vehículos de transporte escolar.

5.2.1 Proyecto SIGTEBAL

El proyecto SIGTEBAL [1] “Sistema de información geográfica para el diseño, gestión, análisis y planificación de rutas de transporte escolar en la Baleares” fue concebido para administrar las rutas de diferentes centros educativos del gobierno Balear.

Para el diseño de las rutas utilizan la malla vial de la ciudad, la cual cuenta con 11.076 cruces y 15.170 arcos. La malla vial tiene datos como distancia, dirección y velocidad media. De igual manera, se cuenta con la ubicación de los planteles educativos y los estudiantes.

El objetivo del aplicativo es unir el sistema de información geográfica sobre el cual está montada toda la información con procesos de optimización para encontrar rutas de menor costo y duración. Para diseñar las rutas primero implementan 4 fases de búsqueda local a partir de vecinos que van mejorando la solución encontrada y luego aplican la metaheurística de búsqueda tabú para mejorar la solución. Terminando el proceso con una fase de intensificación en donde se hace una búsqueda más rigurosa en las vecindades de las soluciones ya encontradas.

El aplicativo fue elaborado en Visual Basic y permite ver la distribución final de las rutas, las paradas y la localización de los planteles educativos.

Los resultados que obtienen en la mayoría de los casos son satisfactorios, reducen la distancia total recorrida y por ende el tiempo de viaje, pero hay casos en los que los resultados obtenidos son más costosos que las rutas que en la realidad se utilizan, debido a deficiencias en la información suministrada.

5.3 Definición de términos

Paradero

Ubicación en el que es recogido y dejado el estudiante por el vehículo transportador.

Recorrido

Trayecto que realiza el vehículo empezando en el garaje, pasando por los paraderos asignados para recoger estudiantes, hasta la institución educativa, o viceversa.

Recorrido libre

Comprende el trayecto en el cual el bus se encuentra desocupado dentro del recorrido. Este puede ir desde el garaje hasta el ingreso en el vehículo del primer estudiante dentro del trayecto de recogida, o desde el último estudiante que abandona el vehículo hasta el ingreso en el garaje.

Ruta o Recorrido ocupado

Trayecto en el cual el vehículo se encuentra con estudiantes a bordo.

Trayecto recogida

Trayecto que el vehículo realiza con la finalidad de llevar a los estudiantes al centro educativo.

Trayecto distributivo

Trayecto que el vehículo realiza con la finalidad de distribuir a los estudiantes en sus viviendas después de la jornada escolar.

5.4 Rutas de colegio como VRP y NP-Hard

Con base en lo presentado en numerales anteriores, se puede concluir que el problema de enrutamiento de vehículos de transporte escolar se puede enmarcar dentro de un problema típico de VRP. Para ser más precisos, es un DCVRP (Dinstance Constrained Capacitated VRP por sus siglas en inglés) ya que se busca minimizar la distancia total recorrida por cada vehículo con la limitante de capacidad que cada uno de ellos tiene.

Determinar la mejor ruta en un problema de enrutamiento en donde se minimice el recorrido, involucra mirar todas las posibles soluciones para encontrar cual es la menos costosa, si esta labor se realiza con un número pequeño de nodos, tres o cuatro, el proceso de búsqueda de la mejor solución es factible y se puede realizar en una prueba de escritorio. Pero a medida que el número de nodos se incrementa, el número de combinaciones aumenta y se hace necesario recurrir a la ayuda de un computador. Se puede llegar a situaciones en las que ni siquiera la capacidad de cómputo es capaz de encontrar la mejor combinación evitando llegar a la solución correcta en un buen tiempo.

El problema entra en la categoría de los problemas de decisión con complejidad NP-*Hard*. Para este tipo de problemas se hace necesario el uso de heurísticas y metaheurísticas que faciliten la búsqueda de la solución, las cuales se explican en capítulos posteriores.

5.5 (**)*Definición matemática del problema*

Determinar el conjunto de recorridos óptimos que minimicen el número de elementos y la distancia total del viaje de una flota de vehículos de transporte escolar, los cuales sirven a un centro educativo en la movilización de sus estudiantes desde y hasta sus residencias. Se parte de un número mínimo de buses escolares que suplen a la totalidad de estudiantes.

Se trabaja con un grafo dirigido que se obtiene del análisis de la malla vial y la ubicación geográfica de cada uno de los estudiantes. La malla vial tiene como atributos el sentido de las vías y su longitud, permitiendo generar un grafo dirigido entre todos los posibles puntos de parada de la ruta. La información de los costos se almacena en una matriz origen destino, en donde la fila en donde está el dato indica el origen y la columna el destino, es una matriz asimétrica. La diagonal de la matriz está llena de valores positivos muy grandes para evitar que se arme la ruta de un punto consigo mismo.

El problema se resume en:

1. Se tiene un grafo dirigido $G = \{V, A\}$ en donde:
 - $V = \{0, 1, \dots, n\}$ es el conjunto de vértices.
 - $\{1, 2, \dots, n-1\}$ conjunto de vértices que representa a los estudiantes.
 - 0 representa el vértice de partida.
 - n representa el punto de llegada.
 - $A = \{(vi, vj) \mid vi \in V, vj \in V; i \neq j\}$ es el conjunto de arcos.
2. c es una matriz con los costos (distancias) c_{ij} entre los vértices v_i y v_j .
 - Hay vértices en donde el costo que tiene ir en un sentido, no es el mismo si el recorrido es en sentido contrario.

- $c_{ij} > 0$ cuando existe un arco entre i y j , de lo contrario $c_{ij} = \max$. Cuando $i = j$ entonces $c_{ij} = \max$.

Matriz c

	0	1	2	3	4	5	6	7	8
0	max	10	11	18	16	20	25	50	10
1	20	max	15	30	16	16	11	12	18
2	20	15	max	9	18	16	15	16	30
3	16	16	15	max	3	18	16	20	9
4	20	18	16	10	max	3	18	15	15
5	16	30	18	18	10	max	3	18	18
6	30	15	18	15	18	10	max	16	20
7	9	16	30	16	30	18	30	max	50
8	3	18	9	18	9	30	9	3	max

3. $F = \{1, \dots, k\}$ es la flota de vehículos, en donde k es el número total de vehículos.
4. D es un vector de demanda en donde D_i indica la demanda para el nodo i . Es decir cuantos estudiantes se recogerán en un punto determinado.

Vector D

i	D
0	2
1	3
2	1
3	1
4	2
5	1
6	1
7	3
8	2

5. C es un vector de capacidad en donde C_i indica la capacidad del vehículo i .

Vector C

i	C
0	4
1	4

6. x es una matriz de flujo que indica si un vehículo atraviesa un arco en la solución encontrada. Toma los valores de 1 si el arco $(i, j) \in A$ y 0 de lo contrario.

Matriz x

	0	1	2	3	4	5	6	7	8
0		1		1					
1					1				
2						1			
3			1						
4								1	
5							1		
6									1
7									1
8									

7. Los vehículos se desplazan a una velocidad promedio por la malla vial v_{ruta} .
8. El tiempo que el vehículo permanece libre se denota t_{libre} .
9. El tiempo que el vehículo permanece ocupado se denota t_{ruta} .
10. La duración total de la ruta es $t_{recorrido} = t_{ruta} + t_{libre}$.
11. o es una matriz que contiene los identificadores de las rutas. Esta matriz es similar a la matriz x pero en vez de contener 1 cuando un arco está en la solución óptima, o_{ij} contiene el identificador del vehículo que sirve al nodo j proveniente del nodo i .

Matriz o

	0	1	2	3	4	5	6	7	8
0		1		2					
1					1				
2						2			
3			2						
4								1	
5							2		
6									2
7									1
8									

FUNCIÓN OBJETIVO

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

s.a.

$$(1) \quad \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0, n\}$$

$$(2) \quad \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0, n\}$$

En (1) se estipula que la sumatoria de todos los elementos de una fila es igual a uno sin tener en cuenta la primera y la última fila. Lo mismo sucede con las columnas (2). Esta restricción indica que a un vértice solo le llega un vehículo y de él solo puede salir un vehículo, lo cual permite definir que todas las rutas son disyuntas.

$$(3) \quad \sum_{i \in V} x_{i0} = 0$$

$$(4) \quad \sum_{j \in V} x_{nj} = 0$$

La sumatoria de la primera columna tiene que ser igual a cero (3), esto indica que ningún vehículo regresa al punto de origen.

La sumatoria de la última fila debe ser cero (4), lo que muestra que ningún vehículo va a ir desde el punto de destino a cualquier otro.

$$(5) \quad \sum_{j \in V \setminus \{n\}} x_{0j} = K$$

$$(6) \quad \sum_{i \in V \setminus \{0\}} x_{in} = K$$

La sumatoria de todos los campos de la primera fila sin el último, indica el número total de rutas que salen hacia algún paradero (5). De igual manera, la sumatoria de la última columna sin el primer elemento, indica el número de rutas que llegan al destino final de la ruta (6).

$$(7) \quad x_{0n} = 0$$

La restricción (7) evita que una ruta pueda ir directamente del lugar de origen al destino sin pasar por algún paradero.

$$(8) \quad \sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(s) \quad \forall S \subseteq V \setminus \{0, n\}, S \neq \emptyset$$

	0	1	2	3	4	5	6	7	8
0		1		1					
1					1				
2						1			
3			1						
4								1	
5							1		
6									1
7									1
8									

La restricción (8) es aplicable al caso en el que todos los vehículos tienen la misma capacidad. S es un conjunto de clientes cualquiera el cual es atravesado por un número de arcos no menor que $r(S)$ (indica el número mínimo de vehículos que son necesarios para servir S),

Por último se tiene una restricción para cuando las capacidades de los vehículos son diferentes, para tal caso, lo que se busca es que en el momento en que en la matriz o aparezca el vehículo k , se determina cual es la demanda para ese vértice acumulándose. La sumatoria debe ser menor o igual a la capacidad del vehículo k .

6 Solución propuesta

Con base en el planteamiento que se ha hecho hasta ahora del problema y sus diferentes fundamentos teóricos y prácticos, a continuación se presenta la solución planteada para este problema, la cual consta de una serie de módulos que van transformando y analizando la información de entrada para poder generar la respuesta.

Inicialmente se hace una geocodificación para ubicar a los estudiantes en la malla vial. Luego se segmentan los estudiantes, agrupándolos en clusters que van a ser atendidos por las rutas. Dentro de cada cluster se obtiene la una matriz origen destino para su posterior uso en la generación de la ruta. Una vez obtenida la ruta, se hace un proceso de post-optimización y finalmente se despliega la ruta final (Figura 6-1).

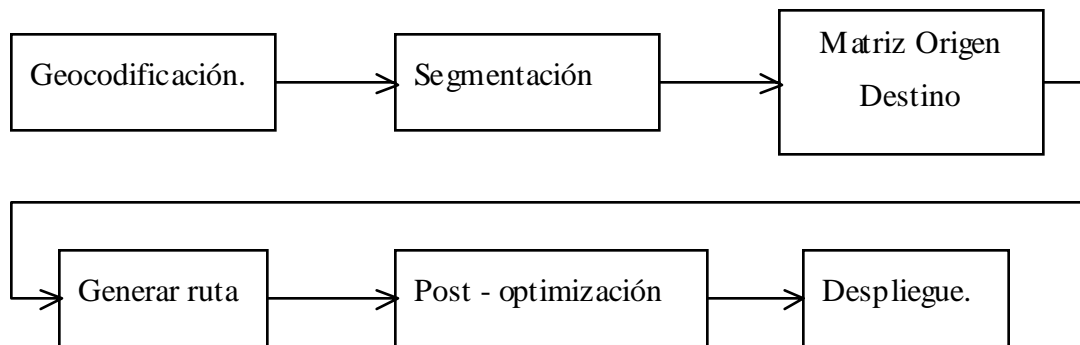


Figura 6-1 Esquema solución propuesta

6.1 Módulo de geocodificación

Por lo general los datos de ubicación de los estudiantes, del colegio y del garaje están en el formato que tiene una dirección catastral de una ciudad común y corriente. Para poder trabajar con ellos sobre una malla vial, es necesario ubicarlos en las mismas coordenadas que tiene la malla.

Para esto se utiliza una herramienta que tiene ArcGIS 9.1, la cual recibe la dirección catastral de los diferentes puntos que se desean ubicar con un formato determinado y genera las coordenadas X y Y sobre la malla vial. Esta herramienta se utiliza para la investigación, dado que no es objetivo de la misma implementar un geocodificador.

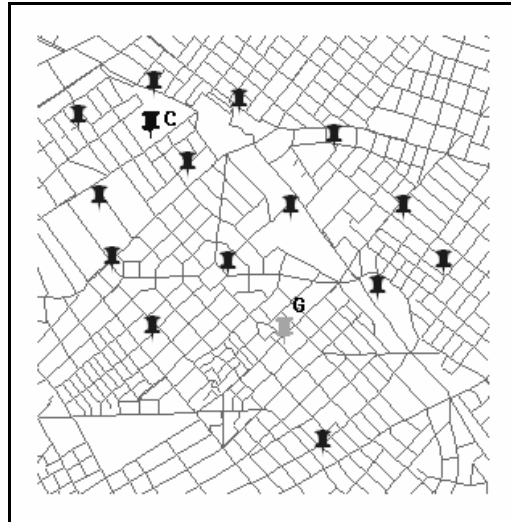


Figura 6-2 Esquema geocodificación.

6.2 Módulo de segmentación

Dada la naturaleza del problema, en donde a medida que aumenta el número de estudiantes aumenta la complejidad, se plantea una solución en donde primero se realiza una segmentación del conjunto total de estudiantes, para así hacer el enrutamiento en conjuntos más pequeños.

La segmentación se hace con base en la restricción de capacidad que tiene cada uno de los vehículos, así mismo se garantiza que la suma de las capacidades de los vehículos, es mayor o igual al número total de alumnos que van a ser recogidos. Para este módulo se utiliza el algoritmo de K-means con algunas variaciones para hacer cumplir las restricciones del problema y adaptar la metodología al caso específico de rutas de transporte escolar.

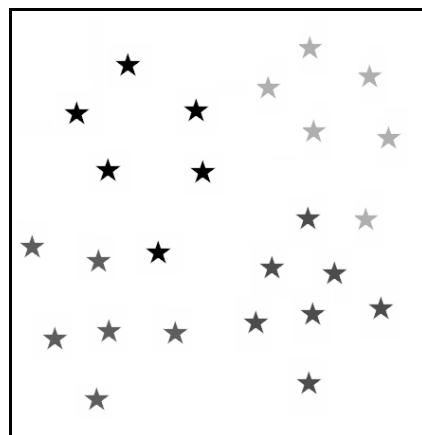


Figura 6-3 Esquema segmentación.

6.3 Módulo matriz origen-destino

Una vez se tienen varios grupos de estudiantes, en donde cada uno contiene los que van a ser atendidos por una ruta, se construye una matriz de costos, cómo la se muestra en la Figura 6-4. Esta tabla contiene el costo de ir de un punto a todos los demás.

Destino Origen	1	2	3	4	5
1	9999	12.3	18.6	152.3	65.22
2	45.4	9999	65.22	697.34	46.25
3	75.12	41.24	9999	30.25	39.45
4	91.43	36.44	100.58	9999	94.2
5	81.37	62.54	94.31	45.36	9999

Figura 6-4 Matriz de costos.

Se obtiene entonces una matriz asimétrica, en la que el costo de la diagonal es infinito o muy grande, con el fin de que en el análisis que genera las rutas, no se tenga en cuenta esta posibilidad y asegurar de esta manera una de las restricciones del problema, en donde no se pueden generar rutas que inician en un punto para llegar a ese mismo.

Para la generación de la matriz origen destino se utiliza la herramienta ArcGIS 9.1, la cual presta la funcionalidad de que a partir de una malla vial y un conjunto de puntos, se pueda obtener el costo que hay entre cada uno de los puntos con los demás.

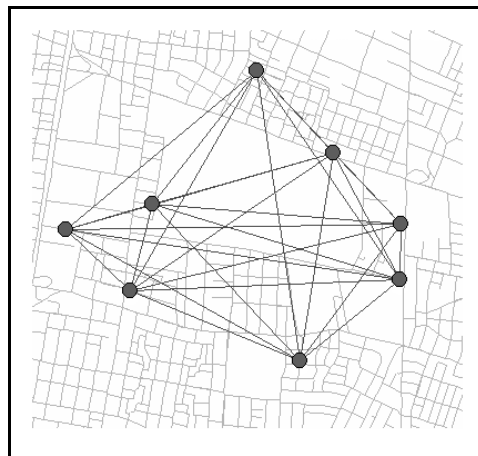


Figura 6-5 Esquema matriz origen-destino.

6.4 Módulo generador de ruta

Con base en la matriz de costos generada en el módulo anterior, se obtiene una primera ruta utilizando la heurística de *Clarke & Wright*. Esta ruta es la que va a ser utilizada por la metaheurística *simulated annealing* para generar la ruta definitiva.

La metaheurística recibe la semilla o solución inicial a partir de la cual va a generar nuevas opciones para evaluarlas y así aceptarlas o rechazarlas. Es importante resaltar que cada ruta en este punto del algoritmo, tiene como inicio y fin los puntos que representan el garaje y el colegio. Puntos que durante todo el proceso no van a ser modificados para garantizar que el recorrido que hace el vehículo transportador es lógico. Por ejemplo, cuando la ruta se encarga de repartir los estudiantes, el punto que representa al colegio (C) debe ser el primero en la secuencia, mientras que el garaje debe ser el último (G).

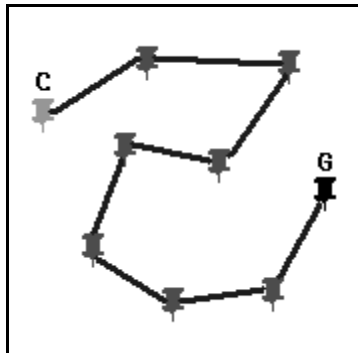


Figura 6-6 Esquema generador ruta.

6.5 Módulo de post-optimización mediante implosión-explisión

Como se ha explicado en capítulos anteriores, la investigación comprende la posibilidad de que un estudiante pueda desplazarse una determinada distancia desde su lugar de residencia hasta un punto en donde pueda pasar el vehículo transportador. Tanto en el recorrido en que recogen como en el que deja a los estudiantes.

Es por eso que se decide agregar a la solución del problema una post-optimización. Cada estudiante tiene un atributo que indica si él puede o no desplazarse a las vecindades de su casa, entonces el algoritmo genera un número determinado de puntos alrededor de los estudiantes que cumplen con la condición (puntos azules en la Figura 6-7), para que sin variar la secuencia de la ruta obtenida en el módulo anterior, se generen nuevas sub-rutas en busca de una mejora o disminución en el costo total.

El proceso en el que un estudiante puede estar representado por un determinado número de puntos alrededor de él se considera como implosión, todos representan el mismo estudiante.

Una vez se ha determinado cual de esos puntos hace que minimice el costo total de la ruta, se asigna cómo punto definitivo dentro de la ruta. Este último proceso se denomina explosión.

La siguiente figura es el esquema del grafo con el que se trabaja, los arcos son las líneas rojas que van desde cada posibilidad a todas las del siguiente estudiante (si las tiene), su costo es la distancia real del recorrido que une cada par de puntos. Los nodos son las estrellas y los puntos azules, los cuales representan los estudiantes y los posibles movimientos, respectivamente.

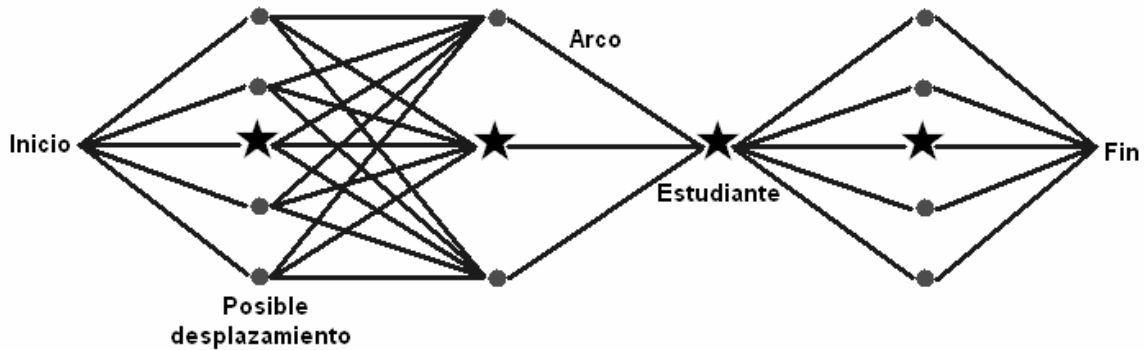


Figura 6-7 Esquema implosión - explosión.

6.6 Módulo despliegue

El módulo final consiste en la presentación en pantalla de las diferentes rutas obtenidas, cada ruta está identificada con un color y el orden en que se recorre está indicado por un número en cada parada.

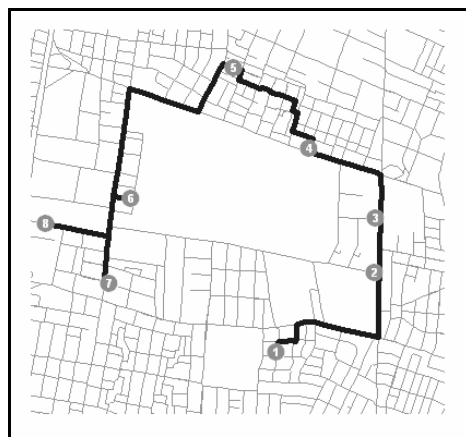
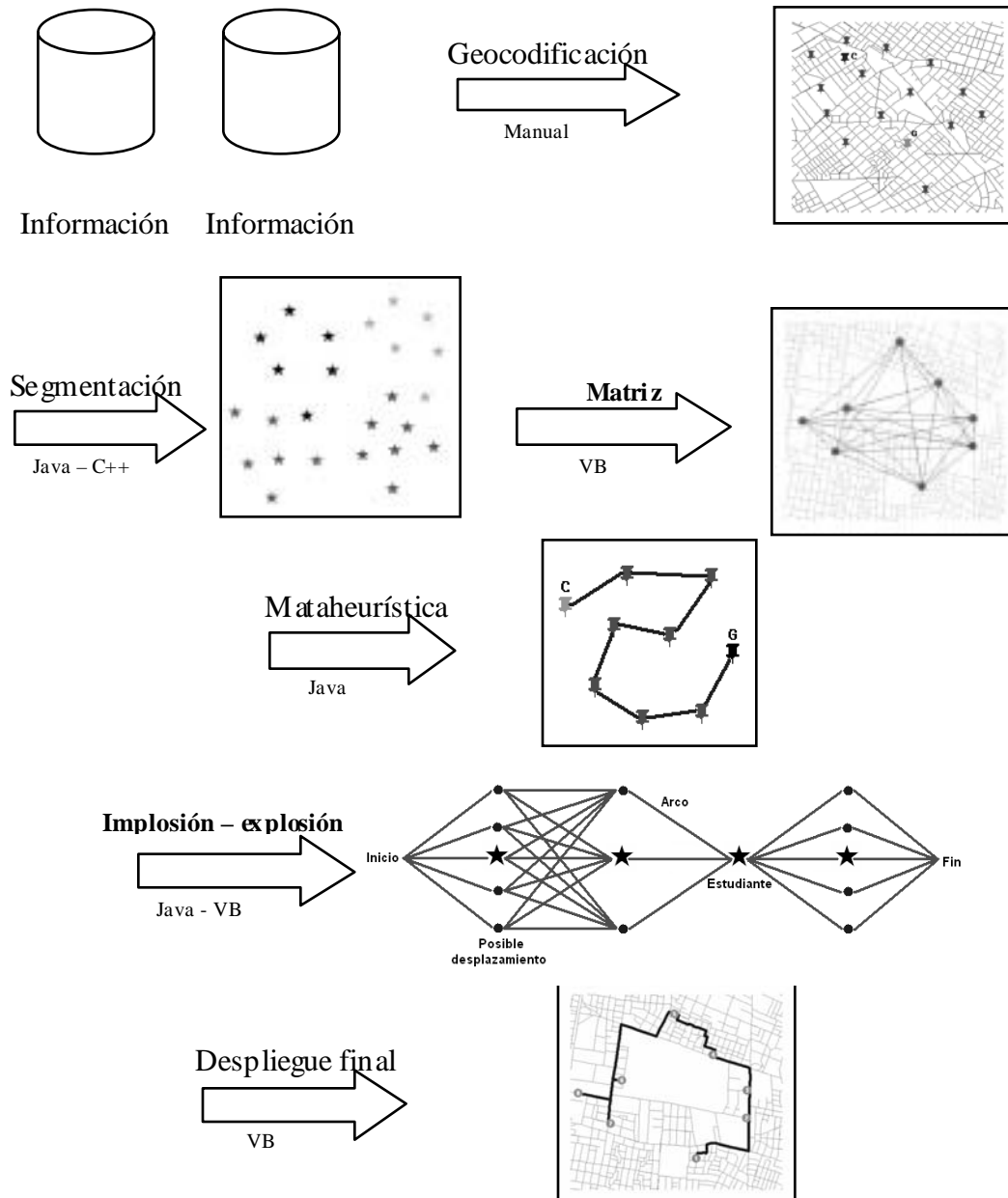


Figura 6-8 Esquema despliegue.

7 Implementación

En este capítulo se expone en detalle la implementación de los diferentes módulos que conforman la herramienta generadora de rutas de transporte escolar, a continuación se muestra un esquema general de todo el proceso y las herramientas utilizadas.



El programa utiliza la siguiente información para generar las rutas:

- **Ubicación del colegio y del garaje**

En un archivo shapeFile (*.shp) se almacena la ubicación geográfica del colegio y del garaje, cabe anotar que el tipo de objetos en los que se guarda dicha información es de tipo "Point". Esto facilita el acople con el resto de la herramienta, ya que los estudiantes, luego del proceso de geocodificación, también estarán almacenados en el mismo tipo de objeto.

- **Estudiantes**

Se debe contar con las direcciones catastrales de cada uno de los estudiantes, además de conocer si pueden o no desplazarse desde y hacia su lugar de residencia para esperar el transporte escolar. Cómo el proceso de geocodificación es manual, no importa el formato original en que se obtiene la información.

- **Malla vial**

Se usa una malla vial en formato (*.shp) la cual se adapta para que el programa ArcGis la pueda interpretar y se generen los datos que sus herramientas pueden brindar, entre ellos, la generación de la matriz origen destino, desplegar en ella las rutas finales, etc.

- **Vehículos**

En un archivo plano que tiene como separador de datos el símbolo (|), se almacena en la primera columna el identificador del vehículo (cualquier cadena), y en la segunda columna la capacidad con que dispone (entero).

7.1 Implementación módulo de geocodificación

En este módulo se trabaja con la malla vial y la base de datos que tiene la información de los estudiantes. ArcGIS 9.1 cuenta con un geocodificador que permite obtener las coordenadas geográficas de una dirección sobre una malla vial.

Debido a que ArcGIS está diseñado para trabajar con direcciones de ciudades norteamericanas, es necesario adaptar nuestra nomenclatura a dicho formato. Las ciudades norteamericanas utilizan un nombre para identificar cada calle acompañado del número de

la casa, el tipo de vía y en algunas ocasiones el código postal y el nombre de la ciudad. Cómo por ejemplo:

Ave	Aurora	572	02457812	Salt Lake City
Tipo de vía	Nombre calle	Número casa	Cod. Postal	Ciudad

De esta manera, se puede hacer una geocodificación a partir del nombre de la calle con el tipo de vía, estos dos datos reducen el campo de búsqueda a una sola calle que tiene como longitud una cuadra, el dato adicional es el número de la casa, el cual si es par está a un costado y si es impar está al otro de la vía.

Tabla 7-1 Formato inicial dirección.

1	0	99	100	53A-119A	S	Kr 53A	
From left	From right	To left	To right	Street	Predir	Street name	Sufdir

En donde:

From left: indica el número de la primera casa que se encuentra a la izquierda de la calle.

From right: indica el número de la primera casa que se encuentra a la derecha de la calle.

To left: indica el número de la última casa que se encuentra a la izquierda de la calle.

To right: indica el número de la última casa que se encuentra a la derecha de la calle.

Street: indica la calle.

Predir: indica si la primera componente de la dirección se encuentra al norte, sur, este u oeste.

Street name: indica el nombre de la vía.

Sufdir: indica si la segunda componente de la dirección se encuentra al norte, sur, este u oeste de la ciudad.

En el caso colombiano es necesario adaptar la nomenclatura para poder hacer un buen uso del geocodificador. El formato inicial en el que están las direcciones de la malla vial de Bogotá D.C. es como el que se muestra en la Tabla 7-1.

Como en Colombia se cuenta con una dirección compuesta, en donde el número de la casa no puede estar relacionado solo con un número, la tabla de datos de la malla vial tiene que ser modificada para quedar de la siguiente manera (Tabla 7-2).

Tabla 7-2 Formato final dirección.

119A-1	119A-0	119A-99	119A-100	53A-119A	S	Kr 53A	
From left	From right	To left	To right	Street	Pedir	Street name	Sufdir

Para lograr transformar los datos fue necesario implementar en C++ una rutina que segmenta la dirección original y la une con forma se necesita en el geocodificador.

De igual forma, las direcciones de los estudiantes se deben colocar en un formato similar para lograr su ubicación geográfica. Como se muestra a continuación, se debe intercambiar en nombre de la vía con el número sobre la calle.

Antes	Después
Cl 73 12-22	12-22 Cl 73

Una vez los datos están en el formato adecuado, se genera un *Address Locator* en *ArcCatalog* que utilice como datos de referencia la malla vial acondicionada. El siguiente paso en la geocodificación es utilizar la herramienta *Geocoding* de ArcMap para obtener las coordenadas X y Y de los estudiantes y generar el *shape* de puntos para su posterior análisis.

La ejecución de este módulo es por medio de procesos manuales soportados por herramientas como ArcGIS, Access, Excel, entre otros.

7.2 (*) *Implementación del módulo de segmentación*

Los datos con que trabaja este módulo son un archivo plano con la información del estudiante: id_estudiante, direccion, coordenada X, coordenada Y, y el atributo de si puede o no caminar desde su lugar de residencia.

El proceso de segmentación consiste en dividir a los diferentes estudiantes de un plantel educativo en grupos según sus direcciones de vivienda, de tal manera que cada uno pueda

ser atendido por un bus. Debido a lo anterior el grupo n , atendido por un bus de capacidad m , no puede ser de más de m estudiantes.

El algoritmo de segmentación combina la heurística *KMeans* con la de Metropoli Montecarlo. La primera es utilizada para determinar los diferentes segmentos en que se divide la población estudiantil según su dirección de vivienda y la segunda para determinar una buena asignación de capacidades a cada uno de los segmentos hallados.

Los pasos de ejecución del algoritmo son los siguientes.

1. Se carga la lista de estudiantes.
2. Se aplica la heurística K-means para segmentar el grupo de estudiantes. El número de segmentos que se obtiene al final del proceso viene dado por la cantidad de buses (n) que suplen a todos los alumnos, los cuales a su vez son incluidos en cada uno de los segmentos generados (Figura 7-1).
3. A cada uno de los segmentos generados se le asigna de manera aleatoria un vehículo, junto con la capacidad del mismo (Figura 7-2).
4. Se eliminan todos los estudiantes de todos los segmentos, quedando únicamente los diferentes centroides de cada uno (Figura 7-2).
5. Se reasignan los estudiantes a los diferentes segmentos, teniendo en cuenta el que tenga su centroide más cerca y respetando la capacidad del segmento específico. Si un estudiante tiene el segmento con centroide c_{ij} más cerca de él que cualquier otro; y siempre y cuando dicho segmento tenga capacidad; el estudiante se le es asignado. En caso contrario, se le asigna al segundo centroide más cercano, y así sucesivamente. El orden en que son asignados los estudiantes también es aleatorio, para garantizar igualdad de condiciones para todos los estudiantes.

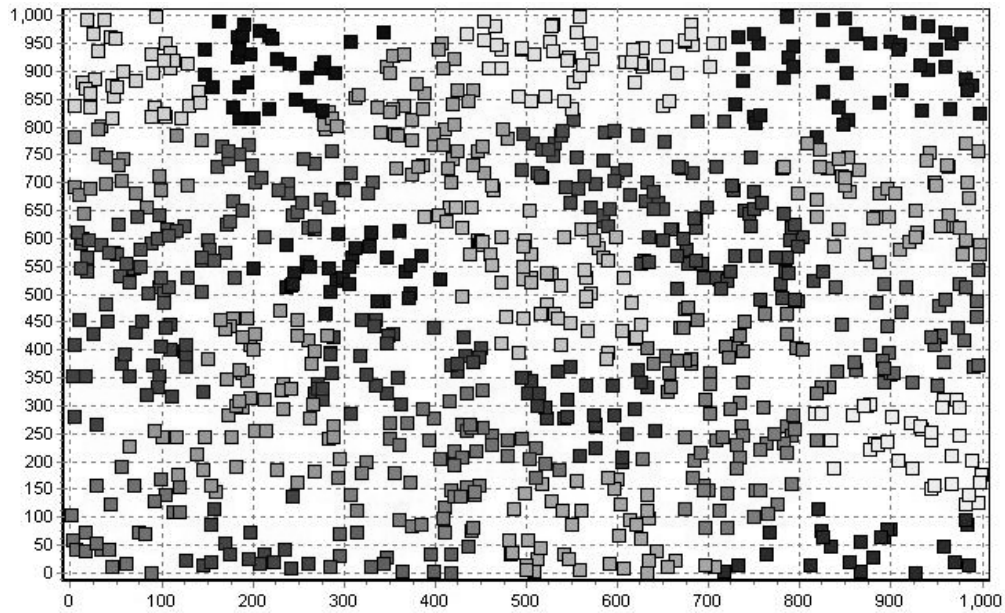


Figura 7-1 Segmentación con KMeans (1000 estudiantes – 30 segmentos).

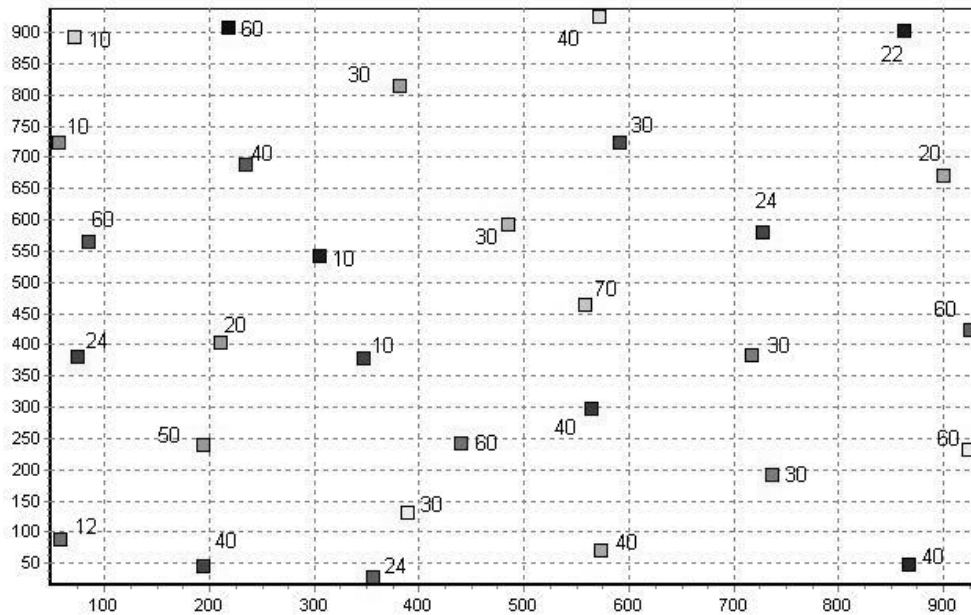


Figura 7-2 Centroides después del proceso de KMeans.

6. Dentro de cada segmento se realiza la sumatoria de las distancias euclidianas desde cada punto a su centroide. Luego se suman todos los resultados de todos los segmentos.
7. Se repite por un número determinado de veces los pasos desde el 3°. Si sumaDistancias es mayor que la nueva suma de distancias, entonces se actualiza sumaDistancias con la nueva sumatoria de distancias y se almacena la nueva distribución de estudiantes y capacidades.

Al finalizar el algoritmo se obtienen segmentos con capacidades correspondientes a la de los buses, y con los estudiantes asignados a los mismos (Figura 7-3).

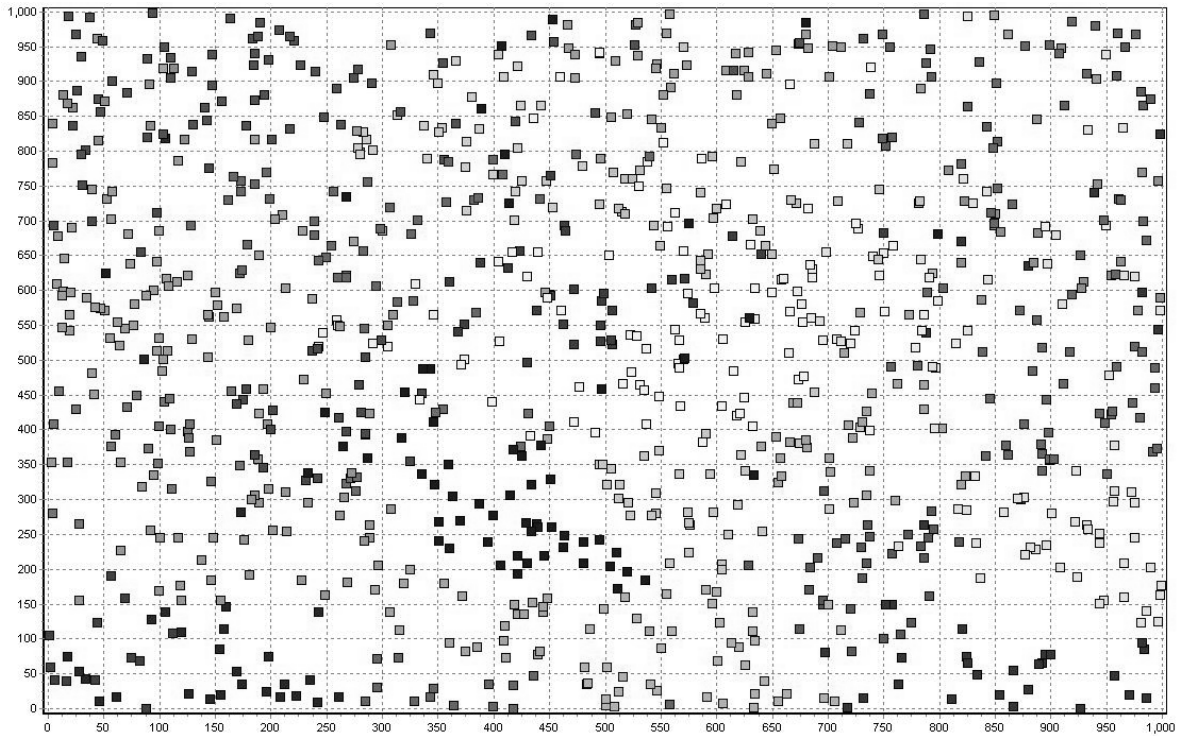


Figura 7-3 Segmentación con KMeans con capacidad.

El número de iteraciones del algoritmo determina la precisión del mismo. En todas las pruebas realizadas, después de 750 iteraciones no se obtenían mejoras significativas, por lo que 1000 iteraciones es un buen número de repeticiones.

La implementación de este módulo se realiza en java, la herramienta que se utiliza para desplegar los resultados y poderlos visualizar se desarrolla en C++.

7.3 Implementación módulo matriz origen-destino

Este módulo trabaja a partir de un archivo plano que es generado por el módulo de segmentación, en él se encuentra el identificador de cada segmento generado, el identificador del estudiante, y las coordenadas X y Y (Tabla 7-3).

Tabla 7-3 Ejemplo archivo entrada matriz origen destino.

Id_segmento	Id_estudiante	X	Y
YUV-356	60	-74.0701504996	4.75559142896
YUV-356	47	-74.0507137281	4.73340310668

YUV-356	31	-74.0401612632	4.71715416135
BPC-234	16	-74.0930440152	4.60740870226
BPC-234	56	-74.1071185856	4.62769762416
BPC-234	1	-74.0986965289	4.57098717879

También se trabaja con la malla vial de la ciudad, la cual tiene el formato adecuado para realizar análisis de costo.

El programa ArGIS cuenta con una extensión que se llama *Network Analyst*, la cual permite realizar estudios de enrutamiento sobre una malla vial, en este caso, tiene la funcionalidad de generar una matriz origen destino, con la cual se obtiene el costo de ir de un punto a otro teniendo en cuenta restricciones de dirección sobre la vía.

El resultado de este proceso es un archivo plano que contiene (Tabla 7-4) en la primera columna el identificador del nodo de origen, en la segunda, el identificador del nodo de destino y en la última el costo en metros que tiene ir de un lugar a otro. Como las vías tienen sentido, en la mayoría de los casos no es lo mismo ir del nodo A al B, que del B al A, generando una matriz asimétrica que eleva el nivel de complejidad del problema a resolver.

Tabla 7-4 Ejemplo resultado matriz origen destino

Id_nodo_origen	Id_nodo_destino	Costo (m)
16	44	1962.38814572923
44	16	2126.45837999768
34	56	1538.90330828341
56	34	1369.66949919188

La implementación de este módulo se realiza en Visual Basic 6, utilizando librerías de ArcGIS.

7.4 Implementación módulo generador de ruta

7.4.1 Clarke & Wright

En la presente investigación se decidió implementar la heurística de Clarke & Wright para encontrar la solución inicial que es usada por la metaheurística que se presenta en este trabajo y en el del ingeniero Robert Antonio Barraza Gamero.

La heurística de Clarke & Wright recibe el archivo plano generado en el módulo anterior con los costos y los identificadores de los nodos inicial y final, lo cual hace que varíe un poco con el planteamiento inicial del algoritmo. Es necesario definir en este punto de la solución en donde termina y donde comienza la ruta para no mover dichos puntos, en donde F representa el punto en el que termina la ruta e I el punto en donde inicia. Para satisfacer esto, el cálculo del ahorro presentado en el Marco Teórico del presente documento se adapta de la siguiente manera (Figura 7-4):

$$s_{ij} = c_{iF} + c_{Ij} - c_{ij}$$

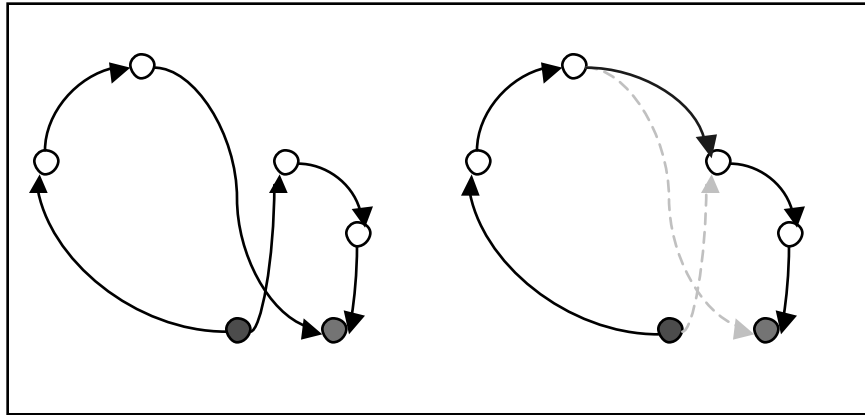


Figura 7-4 Esquema Clarke & Wright.

El algoritmo trabaja de la siguiente manera:

1. Cargar los costos que hay entre cada par de nodos.
2. Calcular el ahorro entre cada par de nodos, conforme en lo explicado con anterioridad.
3. Ordenar de mayor a menor los ahorros.
4. En la primera iteración se agrega a la ruta definitiva tanto el comienzo y el final de la ruta, como la pareja con el mayor ahorro encontrado.
5. Mientras no se termina de evaluar la estructura que contiene los ahorros, se recorre de mayor a menor buscando que alguno de los nodos i o j sean el primero o el último elemento de la ruta creada, al encontrarlo se agrega a la ruta definitiva el nodo que lo acompaña.
6. Al final se obtiene la ruta que va a ser la semilla para la metaheurística.

Nunca se puede contemplar el ahorro entre un nodo consigo mismo.

Esta parte del módulo se implementa en java.

7.4.2 *Simulated annealing*

Para la implementación de la metaheurística se tomaron en cuenta los criterios de modelación que se expusieron en el marco teórico, a continuación se muestra la forma cómo se implementó cada uno de ellos.

Definición de la temperatura inicial

A pesar de que la literatura recomienda comenzar a una temperatura alta, la cual se puede determinar por la mayor diferencia encontrada entre un par de nodos (se puede relacionar con el mayor ahorro que se puede obtener), se realizó un gran número de pruebas a temperaturas mayores, iguales y menores al máximo ahorro encontrado.

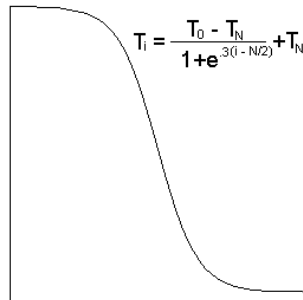
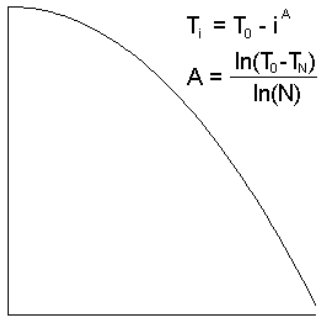
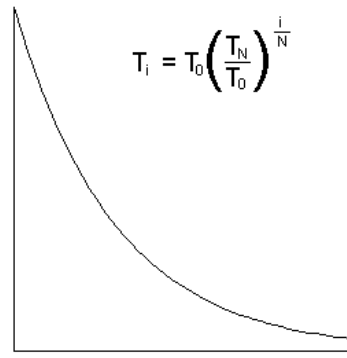
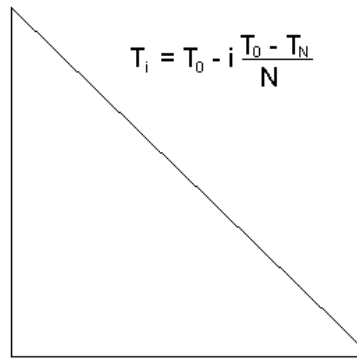
De lo anterior se llegó a la conclusión de que las mejores respuestas se consiguen inicializando la temperatura inicial a un cuarto (1/4) del máximo ahorro conseguido con la heurística de Clarke & Wright. Un análisis más detallado de los resultados obtenidos se presenta en un capítulo posterior.

Determinación de la solución inicial

Cómo se mencionó anteriormente, la solución inicial se obtiene por medio de la heurística de los ahorros (Clarke & Wright).

Determinación del esquema de enfriamiento

Se implementaron y probaron los siguientes esquemas de enfriamiento:



En donde:

- i indica la iteración en la que se encuentra.
- T_i indica la temperatura en la iteración i .
- T_0 indica la temperatura al comienzo del proceso.
- N indica el número total de iteraciones.
- T_N indica la temperatura esperada al finalizar el proceso.

Velocidad de enfriamiento

El número de iteraciones externas está relacionado con la calidad esperada en la respuesta, a medida que se incremente el número de iteraciones, la manera en que decrece la

temperatura es más lenta, permitiendo que se consigan mejores respuestas sacrificando tiempo de cómputo.

Número de búsquedas de vecinos

Este valor es el número de búsquedas que va a hacer la heurística interna de Metropoli Montecarlo para encontrar vecinos. Se probaron diferentes combinaciones en donde se relacionaba la velocidad de enfriamiento con el número de búsquedas de vecinos. Las mejores respuestas se encontraban cuando el número de búsquedas de vecinos era el 30% del número total de iteraciones que tendría al algoritmo total.

Forma de determinar los vecinos

En la presente investigación se implementaron dos tipos de operadores, los cuales intercambian de forma aleatoria nodos de la ruta para buscar disminuciones en el costo de la misma.

El primer operador de intercambio que se implementó fue el RE-INSET (Figura 7-5), el cual escoge de manera aleatoria dos nodos conectados por un arco y un nodo adicional. La pareja de nodos contiguos puede tener el nodo inicial o el nodo final, pero el otro nodo, que debe ser diferente a los ya seleccionados, no puede contener ni el nodo inicial ni el final. Esto con el fin de garantizar que la ruta inicia y termina en los puntos ya definidos y que son inamovibles.

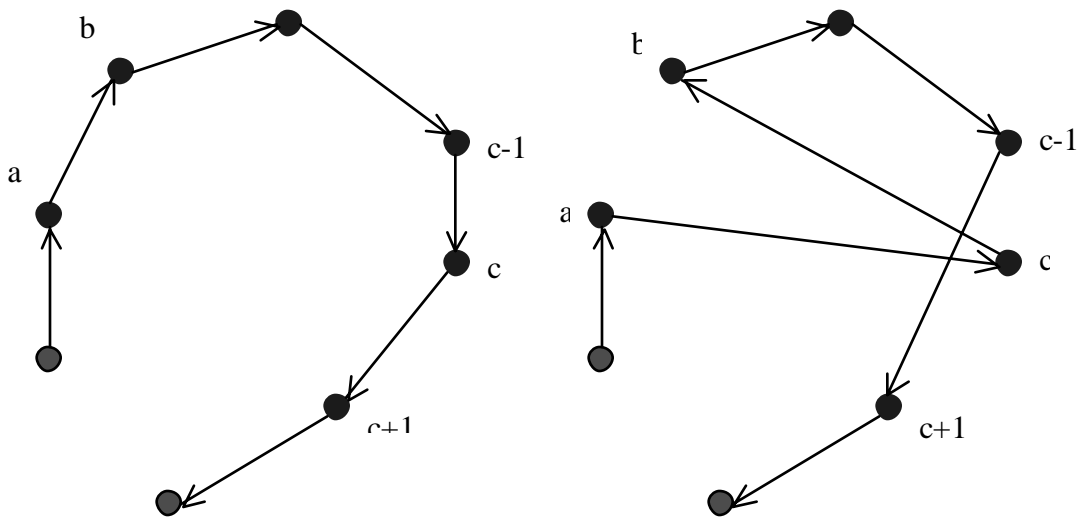


Figura 7-5 Esquema RE-INSET.

Para optimizar el cálculo del costo que tiene la nueva ruta sin tener que recalcularse la suma de todos los arcos, se realiza la siguiente operación teniendo en cuenta el costo ($C(n1, n2)$) de los arcos que ya no se usan y los que si es necesario tener en cuenta:

$$\begin{aligned} \text{NuevoCosto} = & \text{CostoAnterior} - C(a, b) - C(c-1, c) \\ & - C(c, c+1) + C(a, c) + C(c, b) + C(c-1, c+1) \end{aligned}$$

El segundo operador implementado es SWAP (Figura 7-6), el cual selecciona de manera aleatoria dos puntos y los intercambia, ninguno de los dos puntos puede ser el inicio y fin de la ruta.

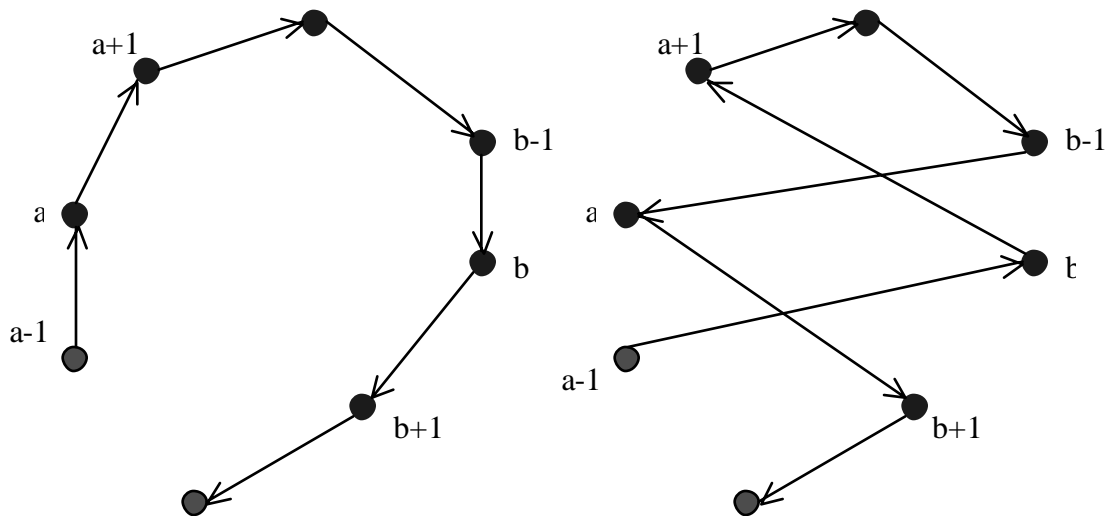


Figura 7-6 Esquema SWAP .

Para optimizar la determinación del nuevo costo se hace el siguiente cálculo:

$$\begin{aligned} \text{NuevoCosto} = & \text{CostoAnterior} - C(a-1, a) - C(a, a+1) - C(b-1, b) \\ & - C(b, b+1) + C(a-1, b) + C(b, a+1) + C(b-1, a) + C(a, b+1) \end{aligned}$$

Luego de varios análisis se encontró que el mejor operador de intercambio entre estos dos para este tipo de problemas es el RE-INSERT, los resultados se muestran en capítulos posteriores.

Valor final de la temperatura

Desde el comienzo de la implementación se determinó que el valor final de la temperatura sería de cero, esto facilita el análisis de los distintos escenarios de prueba que se generaron.

La implementación de la metaheurística Simulated Annealing se realiza en java, su codificación es bastante sencilla y se hace acorde con el algoritmo planteado en el Marco teórico. Al final de su ejecución se obtiene el orden de paradas de la ruta, información que es utilizada por el siguiente módulo para realizar el proceso de post-optimización.

7.5 (*) Implementación módulo de post-optimización mediante implosión-explosión

El módulo de post-optimización tiene como función la mejora de la ruta arrojada por la metaheurística implementada.

Para esto, se vale del hecho que los estudiantes se pueden desplazar un número de metros dados alrededor de su casa, acercándose al sitio por donde transita el bus. Existen otros casos en que los estudiantes cuentan con el servicio puerta a puerta, lo que significa que el bus debe recoger y dejar obligatoriamente al estudiante en su casa, esto se presenta para los estudiantes de más baja edad.

Para la determinación de la nueva ruta teniendo en cuenta esta nueva características se realiza el siguiente proceso.

- Se calculan n puntos alrededor de los estudiantes que se puedan mover de su casa (no cuentan con el servicio puerta a puerta), dichos puntos no deben significar un desplazamiento mayor a una cantidad m de metros para el estudiante. Estos puntos se obtienen de la división en n avos la circunferencia, en el caso de ser 4, se ubican los puntos en los grados 0° , 90° , 180° y 270° , a parte de de esto, se ubican puntos, siempre en estos grados, cada p metros. Por ejemplo si la distancia máxima a desplazarse por cada estudiante es de 200 metros se calculan puntos, según la distancia euclidiana, cada 50 metros hasta un número menor a 200, o sea, 150, y esto se repite para cada uno de los grados obtenidos de dividir la circunferencia en n grados. Con la herramienta ArcGIS se calculan los puntos alrededor del estudiante que más cerca estén de significar para el estudiante un desplazamiento de m metros (Figura 7-7), esta es la razón por la cual no se establece el punto a los 200 metros, ya que siempre la distancia real es un poco mayor que la distancia euclidiana, por lo que nunca la distancia del estudiante a un punto a 200 metros euclidianos, significa 200 metros reales de desplazamiento. Como se ve en la figura, puede que no exista un punto valido para uno

de los grados, en este caso para 270° . A este conjunto de puntos obtenidos se le suma el estudiante, repitiendo este proceso para cada uno de los estudiantes que se puedan desplazar.

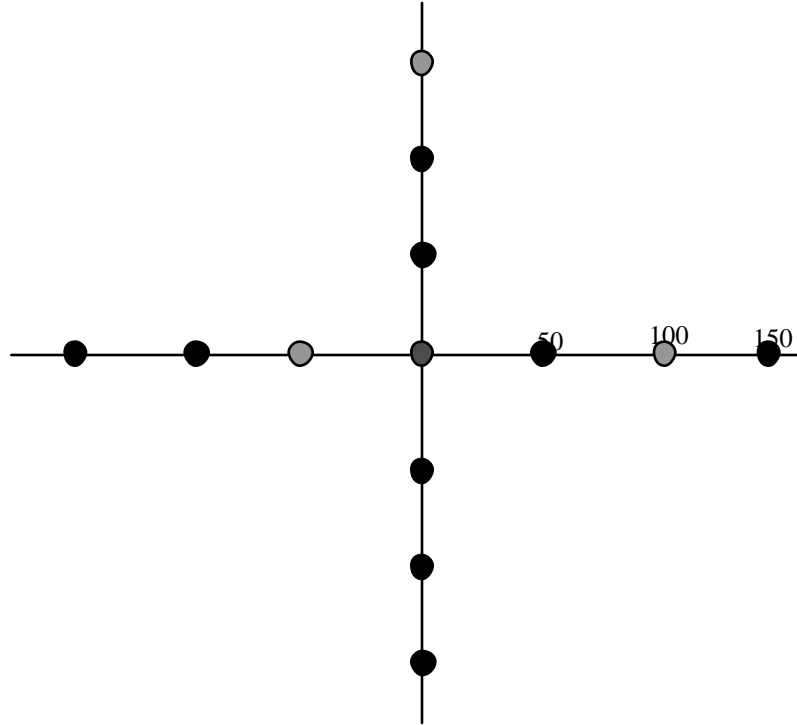


Figura 7-7 Puntos alrededor de un estudiante, 4 segmentos, 200 metros

- El módulo anterior arroja una ruta que se obtiene de aplicar la metaheurística. Basados en este orden, en la ruta recogida de los estudiantes, se calculan las distancias de todos los puntos de dos estudiantes adyacentes. (Figura 7-8).

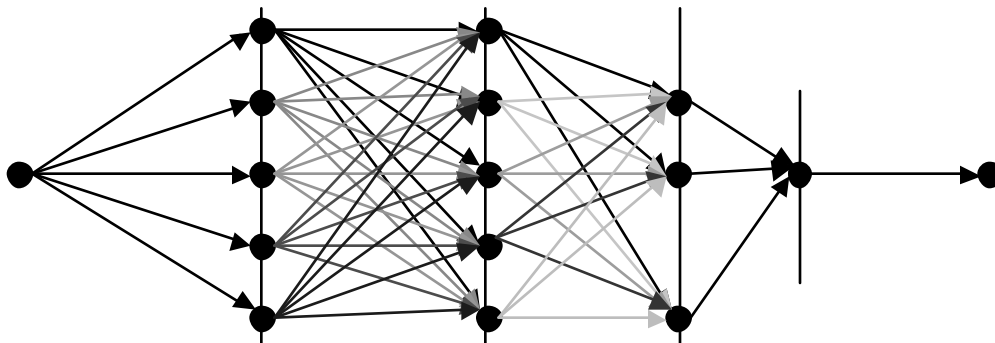


Figura 7-8 Obtención puntos alrededor de un estudiante.

- Una vez se tiene las matrices origen y destino de cada uno de estos segmentos (estudiante1 con estudiante2, estudiante 2 con estudiante 3, etc...) se procede a calcular con la heurística Metrópoli Montecarlo el camino que minimice el recorrido total.

Al terminar el algoritmo, arroja una ruta en el mismo orden arrojado por la metaheurística, pero con los caminos acortados por los desplazamientos de los estudiantes.

7.6 Implementación módulo despliegue

Una vez se ha post-optimizado la ruta, lo único que falta es mostrarle al usuario final la ruta generada, para esto se programa en Visual Basic una interfaz gráfica que le permite al usuario ver sobre la malla vial el resultado de todo el proceso de generación de rutas de transporte escolar.

8 Resultados

En el presente capítulo se muestran los resultados obtenidos de la implementación de la solución propuesta, tanto de la implementación de la metaheurística de Simulated Annealing como la comparación con Tabu search.

8.1 Resultados segmentación

Este módulo busca encontrar segmentos en los que cada uno mantenga la capacidad del vehículo que se asigna para servir sus estudiantes. Además se agrega a cada segmento la ubicación del colegio y del garaje para que en su generación se tenga en cuenta esos dos puntos comunes para todos los grupos.

La función objetivo es minimizar la sumatoria total de las distancias euclidianas (d) de cada uno de los estudiantes (E) al centroide (C) de cada uno de los segmentos, sin exceder la capacidad del segmento (S). En la Figura 8-1 se muestra un esquema del problema seguido del planteamiento matemático del modelo.

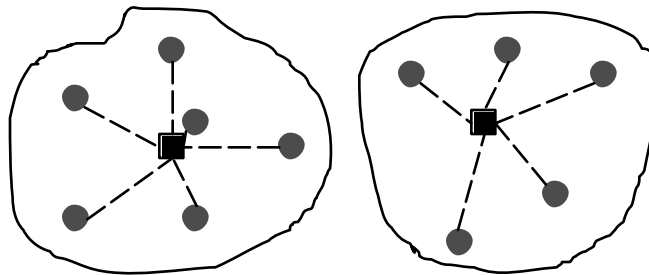


Figura 8-1 Modelo segmentación

$$\min \sum_{i \in C} \sum_{j \in E} d(i, j) X_{ij}$$

sa

$$\sum_{j \in E} X_{ij} \leq S_i, i \in C$$

$$\sum_{i \in C} X_{ij} = 1, j \in E$$

$$X_{ij} \in \{0,1\}$$

Para validar la solución propuesta se implementa en Xpress-IVE un algoritmo para conocer la solución exacta al problema. Esta herramienta se utiliza para resolver problemas de optimización en donde se tiene una función objetivo y un número de restricciones que aplican sobre el modelo. A continuación se muestran los resultados (Tabla 8-1), se puede apreciar que a medida que el tamaño crece la calidad de la respuesta va disminuyendo, pero es preciso anotar que si el algoritmo implementado para la investigación se alarga, los resultados mejoran.

Tabla 8-1 Resultados validación segmentación

Num. Estudiantes	Sumatoria distancia centroides		% del óptimo
	Implementación	Exacta	
20	5309.65	5244.3	1.25%
100	21741.43	21403.8	1.58%
150	28658.37	25241.8	13.54%
250	42361.09	36558.5	15.87%
300	42032.29	36981.7	13.66%
400	52960.81	44891.1	17.98%
600	77148.49	63793.7	20.93%
800	81144.12	64228.4	26.34%
1000	104624.89	88806.6	17.81%

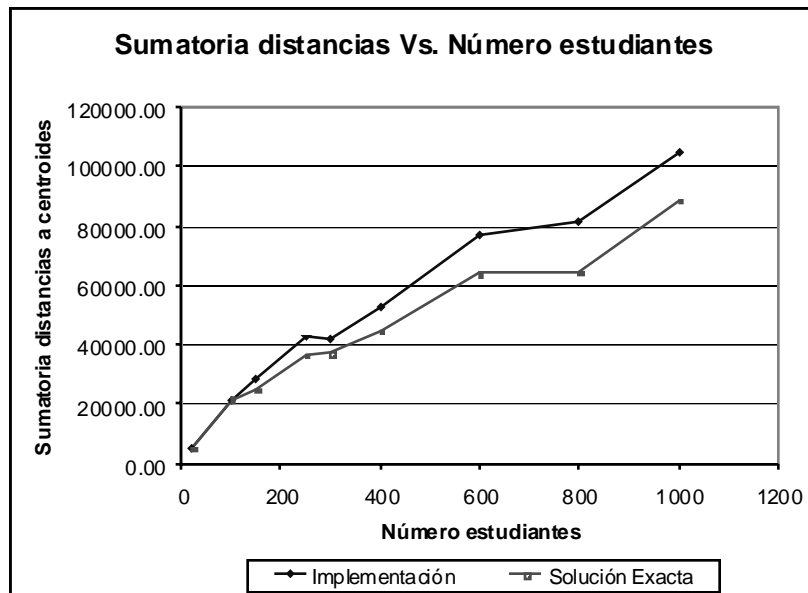


Figura 8-2 Comportamiento del modelo.

8.2 Resultados Simulated Annealing



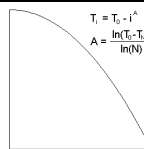
Para poder validar la implementación y ver que se están obteniendo datos lógicos y cercanos a una buena respuesta, se consultaron en Internet instancias de ATSP (*Asymmetric Travelling salesman problem* - por sus siglas en inglés) las cuales proporcionan una matriz de costos asimétrica y una solución conocida con su costo total.

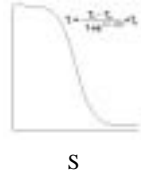
Se utilizan dos instancias, una de 36 nodos (*ftv36.atsp*) y otra de 64 (*ftv64.atsp*). Como nuestro problema particular define que el inicio y el fin de la ruta no pueden cambiar de posición dentro del trayecto de la ruta, se dejan fijos el punto inicial y final de cada respuesta encontrada.

De acuerdo a lo que se expuso en capítulos anteriores, para la implementación de simulated annealing es necesario tener en cuenta varios criterios, entre ellos están: temperatura inicial, esquema de enfriamiento, velocidad de enfriamiento (*numIteracionesSA*), número de búsquedas de vecinos (*numIteracionesMC*), determinación de vecinos (operador intercambio).

Como cada tipo de problema es específico y no existe una configuración de dichos criterios que se acomode de manera genérica a todos los problemas, se debe probar un gran número de posibilidades para determinar de qué manera se llega a buenos resultados sin que el tiempo sea excesivo. En la Tabla 8-2 se resumen las posibilidades que hay para cada uno de los criterios.

Tabla 8-2 Parámetros de Simulated Annealing

CRITERIOS	POSIBILIDADES
Esquema de enfriamiento	 <p>Lineal</p>
	 <p>Convexa</p>
	 <p>Cóncava</p>

	
Operador	RE-INSERT
	SWAP
Temperatura inicial	>max. Ahorro solución inicial
	=max. Ahorro solución inicial
	<max. Ahorro solución inicial
numIteracionesSA	Varias posibilidades
numIteracionesMC	Varias posibilidades

8.2.1 Resultados preliminares esquema de enfriamiento

En el primer análisis se varían los esquemas de enfriamiento, el operador de intercambio y la temperatura inicial. El número de iteraciones de Simulated Annealing se deja constante en 300 y el de Monte Carlo en 800. Generando 24 escenarios ejecutando cada uno de ellos 5 veces para la instancia de 35 puntos.

Cada una de las pruebas tiene un nombre para su posterior identificación y análisis, por ejemplo: 14Concav_SW_412_300_800. En la Tabla 8-3 se explican las convenciones del nombre de la prueba.

Tabla 8-3 Esquema nomenclatura pruebas.

14	Concav	SW	412	300	800
Número prueba	Esquema enfriamiento	Operador	Temperatura inicial	numIteracionesSA	numIteracionesMC

En la Tabla 8-4 se muestran los resultados obtenidos, con el tiempo que gastó en ejecución, el costo de la solución y el porcentaje al cual se encuentra del óptimo.

Tabla 8-4 Resultados preliminares.

OPCIÓN	TIEMPO(m)	COSTO	% del óptimo
14Concav_SW_412_300_800	5.920	1476.333	0.23%
7Convex_SW_618_300_800	6.048	1481.000	0.54%
6Lineal_RI_206_300_800	4.024	1534.667	4.19%
17Concav_RI_412_300_800	4.073	1582.000	7.40%
2Lineal_SW_412_300_800	6.585	1640.667	11.38%
18Concav_RI_206_300_800	3.861	1649.333	11.97%

18Concav_RI_206_300_800	3.861	1649.333	11.97%
10Convex_RI_618_300_800	3.716	1669.333	13.33%
22S_RI_618_300_800	3.883	1689.333	14.69%
15Concav_SW_206_300_800	6.102	1714.333	16.38%
12Convex_RI_206_300_800	3.744	1742.000	18.26%
23S_RI_412_300_800	3.892	1742.667	18.31%
5Lineal_RI_412_300_800	4.302	1746.000	18.53%
20SUno_SW_412_300_800	5.847	1780.333	20.86%
9Convex_SW_206_300_800	6.022	1787.333	21.34%
8Convex_SW_412_300_800	6.152	1804.667	22.52%
21S_SW_206_300_800	6.035	1829.333	24.19%
4Lineal_RI_618_300_800	4.571	1907.000	29.46%
3Lineal_SW_206_300_800	6.239	1909.333	29.62%
24SUno_RI_206_300_800	3.874	1918.000	30.21%
11Convex_RI_412_300_800	3.709	1936.000	31.43%
1Lineal_SW_618_300_800	8.227	1990.000	35.10%
19SUno_SW_618_300_800	5.847	2052.667	39.35%
16Concav_RI_618_300_800	4.226	2130.333	44.63%
13Concav_SW_618_300_800	6.127	2404.000	63.20%

En la Figura 4-1 se muestra como cada uno de los escenarios va convergiendo a una solución. Se aprecia que las series que pertenecen a un esquema de enfriamiento de función convexa llegan a una buena solución en menor tiempo, sin embargo se seleccionan los mejores resultados para profundizar más en su comportamiento.

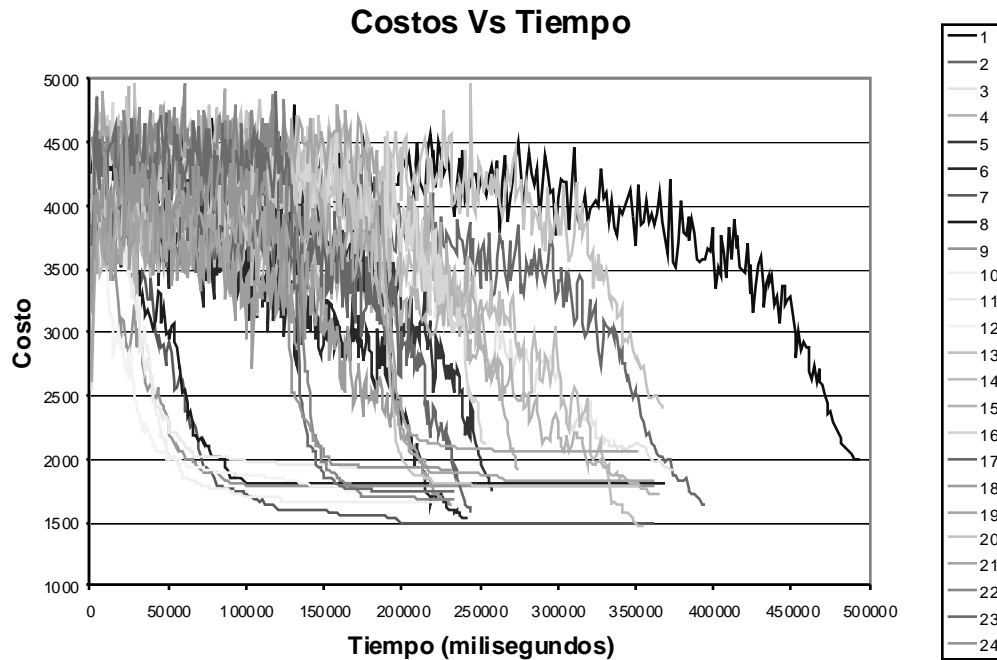


Figura 8-3 Escenarios de prueba preliminar.

Analizando la gráfica y los datos obtenidos se concluyó que las mejores opciones son las que se muestran en la Tabla 8-5, para esta selección se tuvo en cuenta el tiempo de respuesta y la calidad encontrada.

Tabla 8-5 Mejores resultados.

14Concav_SW_412_300_800
7Convex_SW_618_300_800
10Convex_RI_618_300_800
6Lineal_RI_206_300_800
22S_RI_618_300_800

8.2.2 Resultados esquema de enfriamiento

A partir de lo anterior se generaron de nuevo pruebas para ver el comportamiento frente a estos escenarios, pero ahora se incrementa el número de iteraciones tanto del Simulated annealing como de Metropoli Montecarlo, a 500 y 1000 respectivamente. A partir de 10 ejecuciones por escenario se encontraron los resultados (promedio) que se muestran en la Tabla 8-6 y en la Figura 8-4.

Tabla 8-6 Escenarios

OPCIÓN	TIEMPO(m)	COSTO	% del óptimo
10Convex_RI_618_500_1000	8.06388333	1597.67	8.24%
6Lineal_RI_206_500_1000	9.31962222	1612	9.21%
22S_RI_618_500_1000	8.55451667	1620.67	9.80%
14Concav_SW_412_500_1000	13.0355056	1806	22.36%
7Convex_SW_618_500_1000	12.5612056	1852.67	25.52%

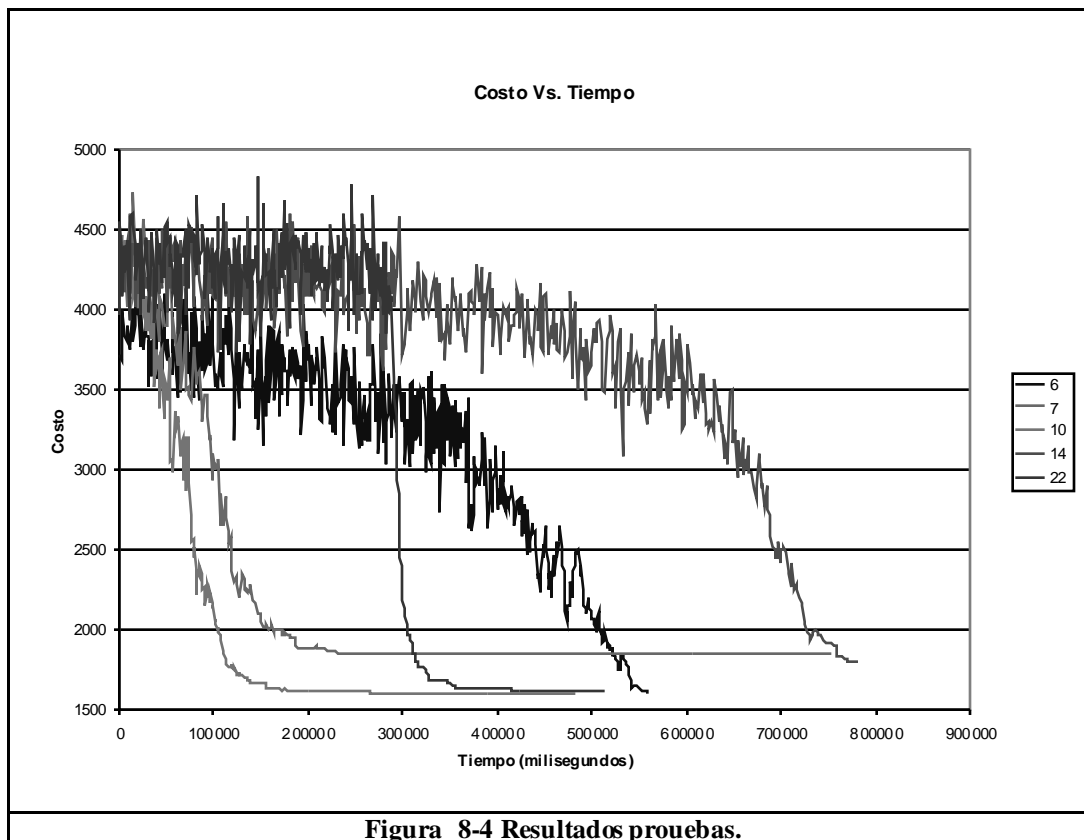


Figura 8-4 Resultados pruebas.

Con base en los análisis anteriores, se concluye que el mejor esquema de enfriamiento que se puede usar es el de la función convexa, y el mejor operador de intercambio es el de RE-INSERT.

En los análisis posteriores no se modifica el esquema de enfriamiento ni el operador de intercambio, ahora se corren varios escenarios en los que se mira la sensibilidad y comportamiento del modelo implementado frente a la temperatura inicial.

8.2.3 Resultados temperatura inicial

El siguiente parámetro que se analiza es la temperatura inicial, para esto se generaron tres escenarios que se ejecutan 10 veces cada uno. Uno con una temperatura mayor al mayor costo encontrado por la heurística de Clarke & Wright, otro con una temperatura igual a dicho costo, y un tercero con una temperatura por debajo de este costo (Tabla 8-7).

Tabla 8-7 Resultados

OPCIÓN	TIEMPO(s)	COSTO	% del óptimo
50Convex_RI_650_300_300	115.984	1510	2.51%
51Convex_RI_425_300_300	93.937	1530	3.87%
52Convex_RI_200_300_300	88.516	1485	0.81%

De lo anterior se concluye que la mejor temperatura inicial es aquella que es inferior a la mitad del mayor costo encontrado por Clarke & Wright, se aconseja tener un valor de un cuarto de ese valor.

8.2.4 Resultados número de iteraciones

Por último, se explora el número de iteraciones tanto para el ciclo de Simulated Annealing como para la búsqueda de Metropoli Montecarlo. Para esto se generan varios escenarios en donde el número total de iteraciones del algoritmo se mantenga alrededor de las 32.000 (para garantizar la misma duración), pero que vaya variando numIteracionesMC con respecto a numIteracionesSA. Los resultados luego de ejecutar 10 veces cada escenario se muestran en la Tabla 8-8.

Tabla 8-8 Resultados iteraciones

OPCIÓN	TIEMPO(s)	COSTO	% del óptimo
67Convex_RI_100_170_189	30.47	1526.67	3.64%
63Convex_RI_100_283_113	31.72	1581.67	7.38%
65Convex_RI_100_231_139	32.52	1584.67	7.58%
70Convex_RI_100_113_283	33.72	1589.00	7.88%
72Convex_RI_100_80_400	31.79	1603.33	8.85%
68Convex_RI_100_139_231	31.15	1617.33	9.80%
60Convex_RI_100_566_57	43.90	1620.33	10.00%
71Convex_RI_100_98_327	30.51	1625.00	10.32%
62Convex_RI_100_327_98	31.30	1657.67	12.54%
66Convex_RI_100_189_170	30.54	1669.33	13.33%
69Convex_RI_100_126_253	31.46	1704.67	15.73%


61Convex_RI_100_400_80	33.36	1706.00	15.82%
73Convex_RI_100_57_566	30.73	1719.00	16.70%
64Convex_RI_100_253_126	31.07	1747.33	18.62%

Se puede inferir que los mejores resultados se obtienen cuando el número de simulaciones del ciclo externo (numIteracionesSA) es aproximadamente el 90% del número de simulaciones del ciclo interno (numIteracionesMC).

8.2.5 Resultados finales

Este análisis se realiza de manera similar para la instancia de 64 nodos. La mejor combinación de criterios de la metaheurística se resume en la Tabla 8-9.

Tabla 8-9 Criterios escogidos

CRITERIOS	OPCIÓN
Esquema de enfriamiento	 Convexa
Operador	RE-INSERT
Temperatura inicial	¼ max. Ahorro solución inicial
numIteraciones SA	Valor definido por el usuario con base en el tiempo que desea que se demore el algoritmo.
numIteracionesMC	110% el valor de numIteracionesSA

8.3 Comparación Tabu Search – Simulated Annealing

Uno de los objetivos de la presente investigación es comparar los resultados obtenidos entre la implementación de la metaheurística simulated annealing y la de tabu search.

Para que la comparación se ambos algoritmos se realice en condiciones iguales, se define el presente ambiente de pruebas:

- Datos de prueba: se utiliza la instancia ftv64.atsp.

- Características del computador utilizado: Intel Pentium 4 1.70GHz
- Lenguaje de implementación: Java 1.5

Se ejecutaron varias veces las dos metaheurísticas para obtener datos comparables, se toma el tiempo y el costo de la solución final, en la Tabla 8-10 y Tabla 8-11, se muestran los resultados obtenidos y en la Figura 8-5 se puede ver el comportamiento de las implementaciones en el tiempo.

Tabla 8-10 Resultados para Simulated annealing

Tiempo(seg)	Costo	% del óptimo
73.95	2154.2	17.46%
160.68	2077.2	13.26%
167.81	2043.8	11.44%
446.93	2042.2	11.35%
624.31	2033.8	10.89%
1073.20	1932.3	5.34%

Tabla 8-11 Resultados para tabu search

Tiempo(seg)	Costo	% del óptimo
17.28	2215	20.77%
78.91	2076	13.19%
336.38	1982	8.06%
556.67	1982	8.06%
753.28	1982	8.06%
1078.67	1982	8.06%

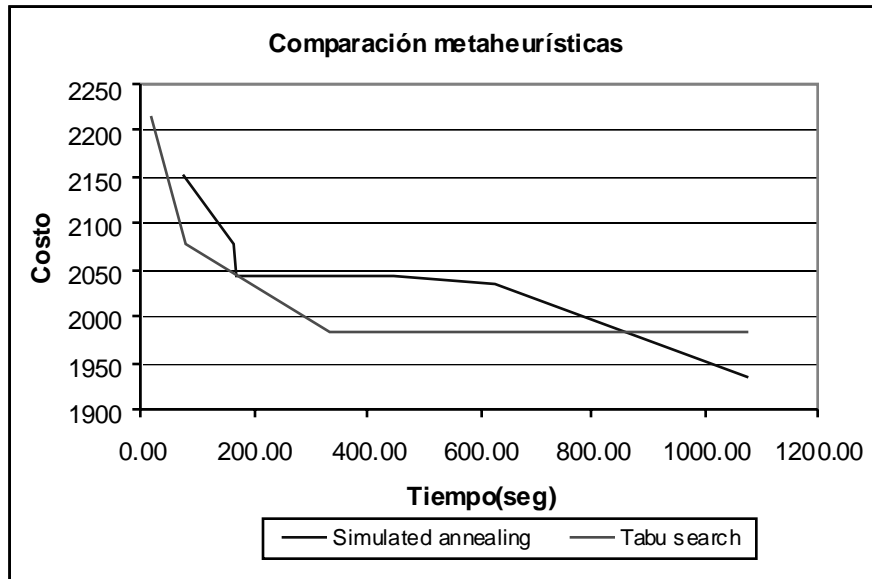


Figura 8-5 Resultados en costo y tiempo de las dos metaheurísticas

Mirando los resultados se puede ver cómo la metaheurística *tabu search* da buenos resultados cuando el número de iteraciones no es muy grande y en cuestión de minutos (para el caso de 64 nodos) se pueden obtener muy buenos resultados, pero a medida que la cantidad de iteraciones aumenta, no se ve una mejora sustancial en la calidad de los resultados.

Al contrario, en *simulated annealing* en pocas iteraciones los resultados no son tan buenos como los obtenidos en *tabu*, pero a medida que estas crecen, la calidad de los resultados mejora por que se explora mucho mejor el espacio de soluciones, en la literatura se encuentra que si este algoritmo se ejecuta infinitamente se puede llegar al óptimo global.

8.4 Comparación entre Clarke & Wright y las metaheurísticas

Es importante mirar en cuanto contribuye o mejora la respuesta entre la solución inicial y la implementación de la metaheurística, en la Tabla 8-12 se aprecia el costo que tiene la solución obtenida por Clarke & Wright y las mejores soluciones encontradas por Tabu Search y Simulated Annealing para la instancia de 64 nodos. En promedio la mejora es alrededor del 27%, dato que consideramos bueno si se mira el tamaño del problema y el contexto en el que se está trabajando, ya que se usa una malla vial real con restricciones reales.

Tabla 8-12 Mejoras en los resultados

Algoritmo	Costo	Mejora
Clarke&Wright	2669	
Simulated Annealing	1932	27.61%
Tabu Search	1982	25.73%

9 Conclusiones

Este documento describe el trabajo de investigación que se realizó como proyecto de grado para culminar los estudios de maestría. El objetivo principal era la generación de una herramienta que a partir de algoritmos de optimización y uso de herramientas SIG fuera capaz de generar rutas de transporte escolar.

Las conclusiones al trabajo son:

- Se encontró que el problema de generación de rutas de transporte escolar se puede fragmentar en varios módulos para atacar problemas diferentes y lograr optimizar varios aspectos por separado. Debido a que al tenerse un conjunto muy grande de estudiantes el análisis de información puede llegar a consumir demasiados recursos computacionales y de tiempo.
- Se encontró que para aplicar una metaheurística como *simulated annealing* a un problema en especial es necesario explorar un gran número de posibles escenarios en donde se varían parámetros como temperatura inicial, número de iteraciones, esquema de enfriamiento. Un análisis exhaustivo permitió concluir que la mejor configuración de estos parámetros para un problema como el expuesto en la presente investigación es:
 - La función de enfriamiento debe ser de tipo convexo.
 - El operador de intercambio debe ser RE-INSERT.
 - La temperatura inicial debe ser $\frac{1}{4}$ del máximo ahorro conseguido con la heurística de Clarke & Wright.
 - El número de iteraciones del ciclo interno (búsqueda por Metropoli Montecarlo) debe ser el 110% del número de iteraciones del ciclo externo.

Lo anterior se considera un aporte importante en la solución de problemas de enrutamiento de vehículos de transporte escolar, ya que en la literatura no se han definido con claridad que características son las que mejores resultados permiten obtener en este problema específico.

- Luego de obtener una ruta preliminar por medio de una metaheurística es importante realizar un proceso de post-optimización para acercarse más al modelo computacional al modelo del mundo, en el presente trabajo se modela la posibilidad que tiene el estudiante de moverse desde su lugar de residencia hasta un lugar en donde para el bus para así acortar el costo total de la ruta
- Un aporte significativo fue desarrollar una interfaz que acopla diferentes módulos independientes de optimización y manejo de la información, permitiendo el refinamiento del aplicativo cambiando o mejorando los módulos ya implementados, sin tener que depender uno del otro.
- Los resultados obtenidos por la metaheurística simulated annealing están muy cerca de las soluciones exactas encontradas en la literatura.
- Al comparar los resultados obtenidos por la metaheurística simulated annealing y tabu search, se concluye que ambas son buenas opciones para solucionar este tipo de problemas, para usar una o la otra, es necesario definir que limitaciones hay, por ejemplo en tiempo de espera de la solución, si se necesitan respuestas inmediatas se justifica el uso de *tabu search*. Pero si se cuenta con varios minutos, se puede optar por la utilización de simulated annealing.

10 Recomendaciones

- Los algoritmos desarrollados en la presente investigación demostraron efectividad para resolver problemas de enrutamiento de vehículos de transporte escolar, se recomienda profundizar y mejorar los módulos propuestos para llegar a una solución informática mejor y que no utiliza demasiados recursos computacionales.
- Como la solución propuesta se basa en módulos, se recomienda probar con otros módulos de optimización para mejorar los resultados y hacer que la eficiencia y rapidez del proceso global mejore.
- Algunos procesos se pueden paralelizar, lo cual puede servir para montar el aplicativo en un sistema distribuido, como en un GRID, y así lograr aprovechar la capacidad de cómputo de varias máquinas al tiempo mejorando los resultados. De igual manera que se puede intensificar la búsqueda y solucionar problemas de mayores proporciones.

11 Referencias

[i] Ruiz Perez,M: La planificación de rutas de transporte escolar a través de un SIG: El proyecto SIGTEBAL.

[ii] Barraza Gamero, R: Hariti, proyecto para el enrutamiento de transporte escolar

[iii] VRP Web <URL: neo.lcc.uma.es/radi-aeb/WebVRP >

[iv] Búsqueda Tabú. Fred Glover. Belén Melián. [En línea]. Disponible en Internet. <<http://tornado.dia.fi.upm.es/caepia/numeros/19/glover.pdf>>

[v] Aarts, E.H.L, Korst, J.H.M.: Simulated annealing and boltzman machines, Wiley, Chichester, 1989

[vi] Kathryn A.Dowsland: Heuristic design and fundamentals of de simulated annealing, 2003

[vii] Kathryn A.Dowsland: Heuristic design and fundamentals of de simulated annealing, 2003

[viii] Consulta sobre el origen del nombre Hariti [En línea] Disponible en Internet <<http://com3.runboard.com/bforosdetanoshii.fkochakocha20.t91> >