ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

INSTITUTO DE CIENCIAS MATEMÁTICAS
ESCUELA DE GRADUADOS

PROYECTO DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
"MAGÍSTER EN CONTROL DE OPERACIONES Y GESTIÓN LOGÍSTICA"

TEMA

DISEÑO DE UN SISTEMA DE SOPORTE DE DECISIONES PARA RESOLVER EL PROBLEMA DE RUTEO EN UN SERVICIO DE COURIER

AUTORES

JAMES TOMALÁ ROBLES
JOHNNY PINCAY VILLA

Guayaquil - Ecuador

AÑO
2010

Dedicatorias

A la evolución inmediata de mi algoritmo genético: Gisela, mi compañera elegida y complemento vital; Ester, fruto del amor y lo mejor de los dos, engendrada y nacida en esta etapa de estudios; y Kevin, la extensión ya creada y evolucionada en nuestro hogar.

James Tomalá Robles

A quienes mientras vivieron me dieron felicidad, lecciones y ahora cuidan de mí desde su morada: mi abuela Felícita, pilar y amor vivo de la familia; mi primo Hernán, habrías aprendido esto y mucho más; mi hermana María, tus ojos brillaron pocas horas y nuestro amor para siempre.

Johnny Pincay Villa

Agradecimientos

Gracias a mi esposa por el apoyo que siempre encuentro en ella, incondicional y amoroso, pero sobre todo sabio e inspirado en Dios, que hace superar cada obstáculo de nuestras vidas.

James Tomalá Robles

Necesitaría otra vida para demostrar lo agradecido que me siento con todos quienes entregaron su tiempo, paciencia, conocimientos y aliento... ¡gracias! Leofer, Lucy, Ámbar y Gloria, su confianza en mí marcó la diferencia.

Johnny Pincay Villa

Declaración Expresa

La responsabilidad por los hechos y doctrinas expuestas en este Proyecto de Graduación, así como el Patrimonio Intelectual del mismo, corresponde exclusivamente al **ICM** (**Instituto de Ciencias Matemáticas)** de la Escuela Superior Politécnica del Litoral.

Ing. James Tomalá Robles

Ing. Johnny Pincay Villa

Tribunal de Graduación

M. Sc. Fernando Sandoya Sánchez
PRESIDENTE

Ph. D. Walter Vaca Arellano
DIRECTOR DE TÉSIS

M. Sc. Pedro Echeverría Briones
VOCAL

Autores del proyecto de graduación

Ing. James Tomalá Robles

Ing. Johnny Pincay Villa

Tabla de contenido

Contenido de Figuras

Contenido de tablas

Objetivo General

Proponer un método evolutivo, basado en algoritmos genéticos, para la solución del problema de recolección de carga con ventanas de tiempo; y a partir de ello, plantear el desarrollo e implementación de un sistema de soporte a decisiones para la planificación de las rutas en la operación de un servicio Courier.

Objetivos Particulares

- Modelar el problema de ruteo de courier para la recolección de carga con restricciones de capacidad y tiempo.

- Diseñar e implementar una Metaheurística basada en los algoritmos genéticos que permita obtener una buena solución al problema de ruteo del courier.

- Proponer el desarrollo e implementación de un Sistema de Soporte a Decisiones para automatizar la programación de rutas en el servicio de un courier.

- Evidenciar posibles mejoras en el nivel de servicio con el uso de la Metaheurística propuesta.

Introducción

El Problema de Ruteo de Vehículos (VRP), en el cual se debe determinar un conjunto de rutas a un mínimo costo, tiene numerosas aplicaciones en las operaciones logísticas. El VRP es bien conocido como un problema de programación entera el cual cae dentro de la categoría de problemas NP-Hard, significando que los esfuerzos computacionales requeridos para resolver este problema se incrementan exponencialmente con su tamaño. Visto el VRP como un Problema de Optimización Combinatoria (OC), existen diferentes métodos de solución, los cuales son aplicados a VRP de diferentes maneras y con diferentes enfoques.

Durante la operación de recolección de carga en el servicio de Courier, se deben determinar las rutas que deben realizar cada uno de los camiones disponibles; actividad que se puede modelar como un VRP con ventanas de tiempo y restricciones de capacidad (CVRPTW). Actualmente, en Ecuador existen aproximadamente 800 agencias de couriers, y todas utilizan métodos empíricos para definir sus rutas lo cual no siempre garantiza obtener una buena planificación, encareciendo sus costos de operación y disminuyendo la calidad del servicio a los clientes. De aquí que es de suma importancia realizar investigaciones orientadas al desarrollo de sistemas de soporte de decisiones para este tipo de operaciones logísticas, donde se empleen procedimientos heurísticos de alta eficiencia.

El presente estudio aporta una alternativa a la adquisición de software de ruteo, que por lo general se consigue en el exterior a altos precios; parte de un enfoque conceptual al modelar el problema de entrega de carga como un CVRPTW, e investiga posibles alternativas de solución desde un enfoque evolutivo, es decir empleando uno de los métodos metaheurísticos más usados y desarrollados durante la última década, para luego proponer el desarrollo e implementación de un sistema de soporte de decisiones (DSS) que ayude a en la planificación de rutas. De esta manera, se aborda esta temática según su real naturaleza; es decir, se centra en resolver el problema de ruteo como uno de optimización combinatoria, buscando un método eficiente y moderno, y no en la implementación informática del DSS, pero sin dejar de darle importancia a ésta.

Por último, se utilizan los datos de una de las empresas courier líder en el mercado local, por su red de puntos de servicio y cobertura nacional, con el objetivo de evidenciar la aplicabilidad de la propuesta.

Capítulo 1

1. Introducción al problema de ruteo de vehículos

1.1 Introducción

La planificación de rutas es uno de los principales problemas en la optimización de las operaciones logísticas de transporte, cuyo objetivo principal es reducir los costos de esta actividad. Esos potenciales ahorros, entre el 5 % y el 20 % de los costos totales de transportación y entre el 10 % y el 20 % del costo final de los bienes, justifican el uso de técnicas de investigación de operaciones (IO) [1] [2]; concretamente, este tipo de problemas corresponden a la Optimización Combinatoria (OC), rama de la IO que estudia la optimización sobre conjuntos factibles discretos de gran cardinalidad, y que mediante las técnicas heurísticas permite el acercamiento a soluciones óptimas de problemas difíciles de resolver por métodos exactos, explorando un espacio de soluciones generalmente grande.

Los primeros trabajos que se realizaron al respecto fueron en 1959 por Dantzig y Ramser, quienes propusieron por primera vez un algoritmo aproximado y formulación de programación matemática aplicada al servicio de distribución de combustible en estaciones de servicio; luego en 1964, Clarke y Wright formularon un algoritmo de tipo glotón [2].

El estudio y aplicación del ruteo de vehículos se ha desarrollado gracias a la mejora paralela de la tecnología de información y comunicación (TIC), lo cual ha permitido reducir el tiempo de ejecución de los algoritmos e integrar información operacional a través de componentes tales como los sistemas de información geográfica (SIG).

El interés en el tema no ha sido solo práctico, la motivación académica se debe a la complejidad computacional – rama de la teoría de la computación que estudia los recursos necesarios de tiempo y espacio durante la ejecución de un algoritmo para resolver un problema –; los ordenadores resuelven algoritmos que tienen un coste computacional polinómico, es decir, el tiempo de ejecución depende del tamaño del problema, según una función polinómica; entendiéndose, por algoritmo a una lista de procedimientos ordenados y finitos que permitirán encontrar una solución.

Uno de los problemas más estudiados de OC es el problema del agente viajero (Traveling Salesman Problem - TSP): el problema consiste en visitar una y solo una vez un conjunto de clientes, partiendo de un punto inicial y finalmente retornando a éste, siguiendo una ruta de menor costo.

El problema de ruteo de vehículos (Vehicle Routing Problem – VRP) tiene planteamientos basados en TSP y consiste en determinar un conjunto de rutas de costo mínimo, para una flota de vehículos. Actualmente se resuelven estos problemas de manera aproximada y de forma eficiente con adaptaciones de algoritmos aplicados al TSP mediante la incorporación de metaheurísticas híbridas (unión y mezcla de algoritmos exactos y aproximados).

Debido a la complejidad de estos problemas, no siempre es posible usar algoritmos exactos, es decir que son tratados con algoritmos aproximados cuyas soluciones sin ser óptimas si se aproximan y se obtienen en tiempo aceptable. Los algoritmos metaheurísticos son una familia de algoritmos que tienen la ventaja de hacer una búsqueda más profunda, aunque a costo de mayor tiempo de procesamiento.

1.2 El problema de ruteo capacitado con ventanas de tiempo (CVRPTW).

1.2.1 Definición del CVRPTW.

El VRP está basado en el problema del agente viajero TSP a diferencia que la cantidad de rutas no está fijada de antemano como en el TSP. En el TSP se dispone de un vehículo que visita una y solo una vez un conjunto de clientes, en una sola ruta con el mínimo uso de recursos, construyendo un camino hamiltoneano (sucesión de aristas adyacentes que visitan todos los vértices de un grafo una sola vez). En el problema básico del TSP, no necesariamente existe un depósito y si hubiere uno se distingue de los clientes y el vehículo deberá visitar n-1 clientes, no hay demanda asociada ni restricciones de horario.

El VRP tiene por objetivo encontrar la ruta óptima para cada vehículo del depósito, cumpliendo con la demanda y restricciones de los clientes, capacidad de una flota de vehículos homogénea y minimizando el costo total.

Tratando de modelar la realidad de diferentes problemas de transporte, se han propuesto variantes y extensiones del VRP, como por ejemplo [29, 30, 31]:

1. MDVRP (multiple depots VRP), los vehículos parten desde diferentes depósitos.
2. VRPSD (VRP stochastic demands), se conocen las demandas de los clientes a través de una distribución de probabilidad.
3. VRPSC (VRP stochastic customers), la cantidad de clientes es una variable aleatoria, conocida con una determinada probabilidad.
4. VRPST (VRP stochastic time), los tiempos de viaje y de servicio son variables estocásticas.

5. SDVRP (Split delivery VRP), un mismo cliente puede ser atendido por varios vehículos, lo cual se aplica cuando la demanda del cliente sobrepasa la capacidad del vehículo.

6. VRPB (VRP bakchauls), el conjunto de clientes es divido en dos grupos, por ejemplo uno para realizar entregas y otro para recolección.

7. VRPPD (VRP picking & delivery), se realiza recolección y entrega en cada cliente visitado.

8. PVRP (period VRP), el horizonte de planeación es para varios días, el cliente es visitado una sola vez.

9. MFVRP (mix fleet VRP) o FSMVRP (fleet size & mix VRP), el depósito cuenta con una flota de vehículos heterogénea.

10. CVRPTW – VRPTW, el problema capacitado con ventanas de tiempo es una de las variantes más importantes del VRP. En este problema, una flota de vehículos con capacidad homogénea realiza el servicio dentro de una ventana de tiempo asociada a cada cliente y el vehículo debe permanecer con el cliente durante el servicio.

VRPSTW, en un problema con ventanas de tiempo suaves, las horas de visitas a los clientes pueden incumplirse a costo de una penalización, mientras que, las ventanas de tiempo duras, VRPHTW, impiden que el vehículo llegue después del cierre de la ventana horaria. Si el vehículo llega antes del inicio de la ventana de atención significa que el cliente no está listo para ser atendido y por lo tanto el vehículo debe esperar.

El problema CVRPTW a su vez puede ser estático o dinámico. Soumia Ichoa, Michael Gendreau y Jean Yves Potvin (2000) definen el problema VRP dinámico como aquel problema de ruteo donde todos o una parte de los requerimientos no es conocido de antemano sino que llegan a medida que se ejecuta el plan inicial de distribución; Zeimpekis, l. Minis, K. Mamassis y G.M. Giaglis proponen el problema dinámico como aquel que usa información en tiempo real para definir las rutas y programar el orden de las visitas a medida que se ejecuta la distribución.

El VRPSTW es un modelo más general e incluye al VRPHTW, el cual podría resolverse aumentando apropiadamente las penalizaciones por atrasos. Sin embargo, la generalización del problema aumenta considerablemente su complejidad. En efecto, se pierde la posibilidad de eliminar a priori soluciones infactibles y la evaluación de la función objetivo tiene un costo computacional más elevado.

1.2.2 Métodos de solución para los problemas de ruteo de vehículos.

Diferentes métodos se han propuesto para solucionar el VRP y se pueden agrupar en exactos y aproximados. Varias propuestas de algoritmos exactos son apropiadas en problemas pequeños, pero dado el alto costo computacional no son adecuadas para problemas mayores y en estos casos se usan algoritmos aproximados:

Algoritmos exactos.- Dada la complejidad de los problemas, solo instancias con pocos clientes (hasta 50 aproximadamente) pueden ser resueltas con métodos exactos; normalmente, se resuelve el problema con un esquema de ramificación y poda o acotación, Branch & Bound: el algoritmo construye un árbol de soluciones con cada una de las ramas y verifica cuando una de las soluciones ya no es óptima y la descarta (poda), por ejemplo si una rama hijo tiene un costo mayor o igual a su rama padre, el hijo debe ser eliminado para evitar el consumo inútil de recursos [1, 20].

Algoritmos aproximados.-Autores contemporáneos como Olivera [1] los clasifican a su vez en heurísticas y metaheurísticas.

Heurísticas.- Son procedimientos simples, no aseguran encontrar la solución óptima pero si una muy cercana a través de una exploración limitada del espacio solución. Las soluciones obtenidas son aceptables y tienen una complejidad algorítmica baja, pueden ser mejoradas a través de otros procedimientos sofisticados pero a mayor costo computacional:

- *Algoritmos constructivos.*- van construyendo una ruta de forma secuencial, por ejemplo, los de inserción y ahorros.
- *Algoritmos de búsqueda local.*- son métodos de mejora iterativa en los que generalmente se define una solución inicial o semilla, y una estructura de vecindario; se exploran los vecindarios evaluando su contribución a la función objetivo.
- *Algoritmos de dos fases.*- primero realizan un agrupamiento para después aplicar otro método para la asignación de las rutas a los vehículos.

Metaheurísticas.- Son procedimientos complejos que por lo general emplean heurísticas de búsqueda local y mejora; tienen un costo computacional más elevado pero exploran el espacio solución de una manera más amplia que los algoritmos heurísticos.

- *Algoritmos de enjambre*.- son métodos bioinspirados en el comportamiento de colonias de insectos como hormigas, abejas, termitas.
- *Algoritmos evolutivos*.- imitan el proceso de la evolución natural en el que sobreviven los individuos con mayor capacidad de adaptación, mientras que los más débiles tienden a extinguirse.
- *Sistemas inmunes artificiales*.- se inspiran en los sistemas inmunes naturales aprovechando la habilidad de estos sistemas para determinar patrones que les permiten distinguir la presencia de cuerpos extraños o antígenos de células del cuerpo, y memorizar la estructura de éstos para una rápida respuesta futura.

1.2.3 Complejidad algorítmica

La OC consiste en encontrar la mejor solución en un conjunto discreto (finito o numerable), de aquí es de suma importancia conocer la factibilidad o no de un método (ya sea exacto o no) antes de invertir esfuerzo en una implementación.

El criterio para saber si un problema de OC es difícil o no, es la complejidad temporal del algoritmo, la que básicamente relaciona el tiempo de ejecución con el tamaño del procedimiento [6]. Existen muchas opciones de solución para un problema, sin embargo, debemos escoger la mejor secuencia de pasos y con el menor tiempo de ejecución. Con este fin, para cada problema se determina una medida *N*, que representa el número de datos a procesar por el programa; entonces el tiempo de ejecución de un programa se mide en función de *N*, lo que se designa como *T(N).*

Los VRP pertenecen en su mayoría a la clase NP- Hard (difícil), debido al gran consumo de recursos computacionales para encontrar una solución óptima que crece de forma exponencial en relación al tamaño del problema (cantidad de clientes) y de aquí el origen de la gran motivación académica e investigativa.

Los problemas de decisión se clasifican en clases de complejidad, la clase P contiene problemas de decisión que pueden ser resueltos en forma relativamente rápida mientras que la clase NP contiene problemas difíciles de resolver[1].

Los problemas NP – Completo se cree que son los más difíciles de resolver, y tienen la siguiente propiedad: si se encuentra un algoritmo eficaz polinomial para resolver uno de ellos, todos los demás se resolverán polinomialmente y por lo tanto será P = NP. Los problemas NP – completos pueden definirse como la intersección entre NP y NP - duro.

La clase NP – duro, contiene problemas de decisión tan complejos como los de la clase NP, pero no necesariamente forma parte de la clase de complejidad NP.

1.3 Definición del problema

Actualmente, en Ecuador existen aproximadamente 800 agencias de couriers, y todas utilizan métodos empíricos para planificar sus rutas lo cual no garantiza obtener una planificación óptima, encareciendo sus costos de operación y disminuyendo la calidad del servicio a los clientes. De aquí que, es de suma importancia realizar investigaciones orientadas al desarrollo de sistemas de soporte de decisiones para este tipo de operaciones logísticas, donde se empleen procedimientos heurísticos de alta eficiencia.

[1] *Wikipedia – Complejidad algorítmica - http://es.wikipedia.org/wiki/Complejidad_algor%C3%ADtmica , revisado 5 de enero del 2010.*

El presente estudio está basado en la realidad de una de las empresas courier líder en el mercado local, por su red de puntos de servicio y cobertura nacional, más de 1,000 clientes corporativos, además de los clientes esporádicos que día a día captan en sus centros de atención al cliente. También se puede sumar una flota aproximada de 100 vehículos entre camiones y motos.

Sin embargo, la empresa enfrenta los siguientes problemas:

a. Muchas de las decisiones del área de logística, como por ejemplo el horario de recolección y la forma de realizar la operación, las toma el departamento de ventas, cediendo ante las exigencias del cliente sin considerar los costos logísticos.

b. Son los supervisores de operaciones, quienes según su mejor criterio, deciden cual camión debe atender determinados clientes.

c. No tienen registros de volumen transportado por ruta, cliente, entre otros indicadores de desempeño.

Por lo descrito anteriormente, se define el problema como la ausencia de un sistema de soporte de decisiones que aplique procedimientos heurísticos para la planificación de las operaciones de transporte de una empresa courier.

El presente estudio se enfocará en encontrar una alternativa de solución para el problema de planificación de las rutas de recolección (picking) del courier. En el capítulo cinco se explican detalladamente las operaciones de la empresa y se realizan pruebas con los datos de la empresa en cuestión.

Una vez formulado el modelo matemático, una solución al problema descrito consiste en desarrollar e implementar una Metaheurística basada en Algoritmos Genéticos para el problema capacitado de ruteo de vehículos con ventanas de tiempo suaves (CVRPSTW) para el servicio de recolección de una empresa courier.

1.4 Formulación matemática del problema

1.4.1 Condiciones del sistema a modelar

Existe poca literatura sobre modelos matemáticos del problema CVRPSTW, son pocos los autores que describen los métodos para penalizar los atrasos y las restricciones para permitir el incumplimiento de la hora máxima de llegada al cliente [3] [4].

Cabe especificar que una ruta se define como la secuencia de visita a cada uno de los clientes, saliendo desde un centro de distribución (CD) y regresando al mismo una vez terminado el servicio; una ruta óptima es aquella que satisface todas las condiciones de la operación con el mínimo costo posible (distancia o tiempo). Las restricciones a tomar en cuenta son la capacidad de los camiones y las ventanas de tiempo establecidas por los clientes.

Como todo modelo matemático es una representación limitada de un sistema real, a continuación se describen las condiciones del servicio courier de recolección que rigen la formulación:

a. Flota ilimitada con capacidad homogénea.
b. Cada vehículo tiene una capacidad finita de carga.
c. El vehículo sale del centro de distribución (CD), atiende a un grupo de clientes geográficamente dispersos y regresa al punto de origen.
d. Si el vehículo llega después de la hora pactada, aún así será atendido por el cliente, a costo de una penalización por el tiempo de espera del cliente.
e. El vehículo debe esperar en los casos que llegue antes de la hora de atención, sin incurrir en una penalización por el tiempo de espera.
f. Existe una hora de salida y una hora de retorno al CD.
g. Si el vehículo retorna al CD después de la hora establecida, aún así puede ingresar, sin incurrir en una penalización por el atraso.
h. En cada visita (parada) se realiza una operación únicamente de carga, respetando la capacidad máxima del vehículo.
i. Cada cliente es visitado solo una vez por un vehículo.
j. La demanda de cada cliente es totalmente satisfecha en cada visita.

1.4.2 Formulación de programación matemática del CVRPSTW

A continuación se describe la formulación matemática del CVRPSTW como una extensión del CVRPTW donde se incluye la consideración de una penalidad por posible atraso en la visita de cada cliente.

Sea $G = (V, A)$ un grafo dirigido, donde $V=\{v_o, v_1, \ldots, v_n, v_{n+1}\}$ es el conjunto de vértices y $A = \{(v_i, v_j); v_i, v_j \in V, i \neq j\}$ se denomina conjunto de arcos. Los vértices v_o y v_{n+1} representan al depósito y cada vértice v_i al cliente i el cual tiene asociado una demanda q_i. Los nodos de visita sin considerar el depósito se los denota por el conjunto $V_c = \{v_1, \ldots, v_n\}$. Una ruta es posible si comienza en el vértice v_o y termina en el vértice v_{n+1}.

Se define c_{ij} el costo de viajar del cliente v_i al v_j. Para el depósito v_o y v_{n+1} y para cada cliente v_i se fija una ventana horaria de servicio $[e_{vi}, l_{vi}]$, dentro de la cual debe ser visitado. Además la demanda y el tiempo de servicio en el depósito son iguales a cero.

Denotaremos por $\Delta^+(v_i)$ y $\Delta^-(v_i)$ al conjunto de nodos adyacentes e incidentes al nodo $v_i \in V_c$, respectivamente, es decir: $\Delta^+(v_i) = \{v_j \in V_c \setminus (v_i, v_j) \in A \}$ y $\Delta^-(v_i) = \{v_j \in V_c \setminus (v_j, v_i) \in A \}$.

Los parámetros descritos, junto a otros, se resumen a continuación:

c_{ij}	Costo asociado al arco $(v_i, v_j) \in A$
q_i	Cantidad demandada cliente i
s_{vi}	Tiempo de servicio en cliente i
$[e_{vi}, l_{vi}]$	Ventana horaria de atención del cliente i
K	Flota de vehículos del depósito.
Q	Capacidad de los vehículos.
f_k	Costo fijo de un vehículo k.
p_{v_i}	Penalización por atraso al atender el cliente v_i.
α_i	Costo por unidad de tiempo de atraso al atender el cliente v_i.

Se procede a definir las variables de decisión del modelo:

$$x_{ijk} = \begin{cases} 1 \ \textit{si el vehículo } k \in K \textit{ se desplaza en el arco } (v_i, v_j) \in V \\ 0 \ \textit{si el vehículo } k \in K \ no \textit{ se desplaza en el arco } (v_i, v_j) \in V \end{cases}$$

h_{v_i} =*tiempo en que empieza el servicio en el cliente* v_i

r_{v_i} =*tiempo de atraso del servicio al cliente* v_i, *donde* $r_{v_i} = max\,\{0,\ h_{v_i} - l_{v_i}\}$

Para cada cliente v_i, h_{v_i} denota la hora real de visita e inicio del servicio.

El valor de α_i varía de acuerdo a la importancia del nodo, si es un cliente crítico el valor será muy elevado o infinito forzando a cumplir con la ventana horaria y en cierto modo convirtiendo el problema en VRPHTW.

$$p_{v_i} = \begin{cases} 0 & si\ h_{v_i} \leq l_{v_i} \\ \alpha_i(h_{v_i} - l_{v_i}) & si\ h_{v_i} > l_{v_i} \end{cases}$$

Se considera el problema con la siguiente función objetivo:

$$Minimizar \sum_{k\,\in K} \sum_{(v_i,v_j)\in A} c_{ij}\, x_{ijk} + \sum_{k\,\in K} \sum_{(v_i,v_j)\in A} f_k\, x_{ijk} + \sum_{i=1}^{n+1} p_{v_i} \qquad (1.1)$$

Se observa que la ecuación 1.1 suma los costos de viaje de cada arco que forma parte de la solución, los costos fijos por el uso de cada vehículo y los costos de penalización por atraso en la atención a los clientes.

A continuación se describen las restricciones de la función objetivo:

- Todos los clientes $v_i \in V_c$ deben ser visitados, una sola vez, lo cual se expresa por la ecuación 1.2.

$$\sum_{k\,\in K} \sum_{v_j \in \Delta^-(v_i)} x_{ijk} = 1 \qquad \forall\, v_i \in V_c \qquad (1.2)$$

- Cada uno de los vehículos parte sólo del depósito al iniciar la ruta; la ecuación 1.3 expresa matemáticamente esta restricción.

$$\sum_{v_j \in \Delta^+(v_0)} x_{0jk} = 1 \qquad \forall k \in K \quad (1.3)$$

- Cada ruta debe ser realizada sólo por un vehículo, es decir que si el cliente v_i es servido por el vehículo k, entonces los clientes antecesor y sucesor a v_i en la ruta deben ser servidos por el mismo vehículo k, lo que se expresa por:

$$\sum_{v_j \in \Delta^-(v_i)} x_{ijk} - \sum_{v_j \in \Delta^+(v_i)} x_{jik} = 0 \qquad \forall\, k \in K, v_i \in V_c \quad (1.4)$$

- La suma de la demanda de todos y cada uno de los clientes visitados por un vehículo $k \in K$, no puede superar su capacidad Q.

$$\sum_{v_i \in V_c} q_i \sum_{v_j \in \Delta^+(v_i)} x_{ijk} \leq Q \qquad \forall k \in K \quad (1.5)$$

- El tiempo de inicio de servicio en cada cliente v_i debe ser como mínimo igual a la cota inferior de la ventana de tiempo respectiva y se puede incurrir en un atraso, tal como se ha planteado al inicio de la formulación, lo que se expresa por:

$$e_{v_i} \leq h_{v_i} \leq l_{v_i} + r_{v_i} \qquad \forall v_i \in V \qquad (1.6)$$

- La ecuación 1.7 emplea *M,* constante suficientemente grande; asegura que se cumpla la continuidad del tiempo y actúa como restricción de eliminación de subtours al agregar variables reales h_{vi} para i = 1,…, n+1; es decir, si un vehículo *k* viaja desde v_i $hasta$ v_j, no puede llegar a v_j antes que $h_{vi} + s_{vi} + t_{ij}$. Estas restricciones fueron propuestas para el problema TSP por Miller, Tucker y Zemlin [2].

$$h_{vj} - h_{vi} \geq s_{vi} + t_{ij} - M\left(1 - x_{ijk}\right) \qquad \forall \left(v_i, v_j\right) \in A, k \in K \quad (1.7)$$

Si x_{ijk} =0, esta restricción sería innecesaria, por el contrario si x_{ijk} =1 la ecuación 1.7 se reduce a $h_{v_i} + s_{v_i} + t_{ij} \leq h_{v_j}$.

- La variable x_{ijk} es binaria y h_{vi} , r_{vi} son positivos.

$$x_{ijk} \in \{0, 1\} \quad y \quad h_{v_i}, r_{v_i} \geq 0 \qquad (1.8)$$

Capítulo 2

2. Desarrollo de una Heurística basada en Algoritmos Genéticos para el problema de ruteo con ventanas de tiempo

2.1 Introducción

Según la literatura el VRP es un problema de optimización combinatoria muy complejo; se ha demostrado que es de tipo NP-Duro y su tiempo se resolución por medio de métodos exactos crece exponencialmente según el tamaño del problema.

Dado la dificultad práctica que representa resolver un problema NP-Duro, se utilizan metaheurísticas que son procedimientos informáticos que nos permiten obtener soluciones factibles muy buenas; por lo general no consiguen el óptimo pero sí una aproximación bastante aceptable en un tiempo de cálculo razonable.

Las metaheurísticas se pueden clasificar de diferentes maneras; una manera de hacerlo es categorizarlas como aquellas inspiradas en la Naturaleza y otras no. Entre las más conocidas y aplicadas en la actualidad se tienen: Algoritmos Evolutivos (AE, Evolutionary Algorithms). Búsqueda Tabú (Tabu Search TS), y Colonia de Hormigas (Ant Colony Optimization ACO). Existen investigaciones como la de T. Bäck, D.B. Fogel, and Z. Michalewicz [10] donde los Algoritmos Evolutivos han sido reportados como una de las mejores metaheurísticas.

2.2 Revisión teórica de los Algoritmos Genéticos

2.2.1 Algoritmos Evolutivos (AE)

En las décadas de los setenta y ochenta varios investigadores desarrollaron de manera independiente algoritmos basados en el proceso de evolución de los organismos vivos, los cuales tomaron creciente interés y aplicabilidad en varias disciplinas gracias a los últimos avances tecnológicos en la ciencia de la computación que lograron su implementación a un costo computacional muy eficiente. Bajo este principio se supone que parte de la evaluación constituye la selección natural de las mejores características de diferentes individuos que compiten dentro de un ambiente determinado.

Se ha probado en la literatura al respecto, que los algoritmos evolutivos (Evolutionary Algorithms EAs) son aplicables para solucionar problemas de optimización que muestran mucha complejidad para ser formulados matemáticamente o de extrema dificultad para ser resueltos por técnicas típicas exactas, aún después de haber sido el problema relajado; de aquí que, son muy atractivos por su ventaja de no requerir mayor modelamiento matemático; es así que, pueden abordar todo tipo de variables de decisión (reales, enteras o binarias), tener uno o varios objetivos, y considerar restricciones de cualquier naturaleza, como las no lineales.

En la tabla 2.1 se expone el Algoritmo Evolutivo general, el cual empieza creando una población inicial P_0; en la fase de reproducción se mejora de manera iterativa a los individuos de la población, generando mejores soluciones en cada etapa. Enrique Alba [9] define que cada iteración consta básicamente de dos fases: "*la primera decide quiénes compiten por la reproducción (emparejamiento) y la segunda decide cuáles de entre todos los individuos (nuevos y viejos) van a sobrevivir*".

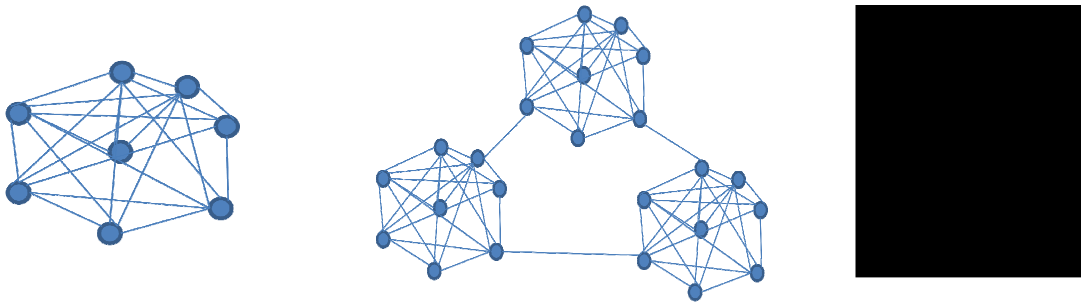
En cada iteración se determinan dos conjuntos: P_t que se define como la población de individuos en la etapa t y C_t el conjunto de los nuevos individuos, llamados hijos, generados a partir de P_t; el tamaño de P_t se denominará por μ y de C_t por λ; es a partir de estos grupos que se determina una nueva población a ser considerada en la siguiente etapa $t+1$. Existen dos maneras de construir la población P_{t+1} ; la primera de ellas es elegir los mejores μ individuos de $P_t \cup C_t$, lo que se denota por $(\mu+\lambda)$-AE; en la segunda manera la población P_{t+1} se conforma reemplazado P_t por los mejores μ elementos del conjunto C_t, lo que denota por (μ, λ)-AE.

```
BEGIN /* Algoritmo Evolutivo General */
        t←0
        Po←generar_poblacion_incial()
        evaluar (Po).
        /* Fase de reproducción y generación de una nueva población*/
        WHILE (No se alcancen las condiciones de finalización) DO
                /* Producir nueva generacion */
                padres←seleccionar_padres(Pt)
                Ct← aplicar_operadores_variación(padres).
                evaluar (Ct).
                P t+1←seleccionar_nueva_población(Pt,Ct).
                t←t+1
        END
END
```
Tabla 2.1: Algoritmo Evolutivo General

Algoritmos Evolutivos Estructurados

En el modelo AE general, cada individuo puede interactuar con cualquier otro de la población, esto se conoce como **Panmixia**. Existe un tipo de AE, que ha tomado reciente importancia, donde la población tiene cierta estructura que restringe la interacción de sus elementos, es decir, un individuo sólo puede combinarse con un grupo definido de otros; a estos AE se los denomina Algoritmos Evolutivos Estructurados (AEe). Existen muchas maneras de estructurar una población, pero las más importantes son: de tipo distribuido y de tipo celular.

Algoritmos Evolutivos distribuidos.- No consideran una población única, en vez de ello agrupan a los individuos en subpoblaciones. Enrique Alba [9] expresa que los AE distribuidos "*demuestran comportarse de forma distinta a la secuencial y, en general, mejoran la búsqueda. Este hecho es lógico, ya que, continuando con la idea de copiar el funcionamiento de la evolución natural, no deberíamos permitir que cualquier individuo se empareje con cualquier otro de la población global (panmixia). Por el contrario, es normal la evolución en grupos separados, dando lugar a especies (demes) o nichos. De esta manera, una población de individuos separada en grupos de reproducción casi-aislada se conoce como politípica, y es la visión natural equivalente* *a la ejecución paralela en islas de individuos típica de los algoritmos evolutivos distribuidos*"[2].



a) AE general b) AE distribuido c) AE celulares

Figura 2.1: Representación de red de las estructuras de las poblaciones

Algoritmos Evolutivos celulares.- Se puede decir que los AE celulares son similares a los AE distribuidos con la diferencia de que en vez de considerar subpoblaciones, se definen vecindades para cada individuo; es decir, cada elemento sólo puede combinarse con sus vecinos más próximos. Para crear la estructura de la población se emplea por lo general una representación bidimensional de la ubicación de los individuos, como en la figura 2.1 c.

Existen varias maneras de estructurar las vecindades, las más comunes son L_n y C_n; en la estructura L_n cada vecindad está compuesta por los n vecinos más próximos según coordenadas verticales y horizontales (norte, sur, oeste y este); mientras que en la C_n se consideran los n-1 vecinos más próximos no sólo en las direcciones horizontales y verticales, sino además diagonales. En la figura 2.1 c se ilustra una vecindad de tipo L_5 o NEWS por sus siglas en inglés (North, East, West, South) donde cada elemento tiene un conjunto de vecinos compuesta por los individuos más cercanos del norte, este, oeste y sur.

[2] *Enrique Alba Torres, "Análisis y diseño de algoritmos genéticos paralelos distribuidos, Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. Página 13*

En un AEc, en la fase de reproducción y mejoramiento, un individuo se combina con un elemento de su vecindad, bajo un criterio definido, a los cuales se les aplican los operadores de variación y si produce un hijo con mejor aptitud entonces se reemplaza en la misma ubicación del elemento.

2.2.2 Algoritmos Genéticos (AG)

Los algoritmos genéticos son un tipo de Algoritmos Evolutivos; Fueron implementados de manera básica y por primera vez por Holland en 1975, mientras que la aplicación de los AG para resolver problemas complejos fue realizada por Jong en 1975 y Goldgerg en 1989.

Los AG utilizan una representación genética de cada solución del problema llamada **cromosoma**; en biología un cromosoma especifica cómo un organismo está estructurado. El proceso de convertir las soluciones en cromosomas se llama *codificación*. Cada cromosoma es un conjunto de individuos llamados **genes**, la ubicación de cada gen se denomina **locus**. Los diferentes valores que puede tomar un gen se conocen como **alelos**.

Como un Algoritmo Evolutivo, el AG empieza con una población inicial P_0, y luego implementa un proceso de mejoramiento; para generar nuevas soluciones, se aplican los operadores genéticos: **combinación** (operación con dos individuos) y **mutación** (operación sobre un individuo); en cada iteración se genera una nueva población P_{t+1} seleccionando a los mejores; se finaliza cuando se cumplen las condiciones de terminación.

En la figura 2.1 se ilustra un AG básico, el cual requiere la aplicación de los siguientes procedimientos: Generación de la población inicial, Codificación, Cruce, Mutación, Selección, Decodificación.

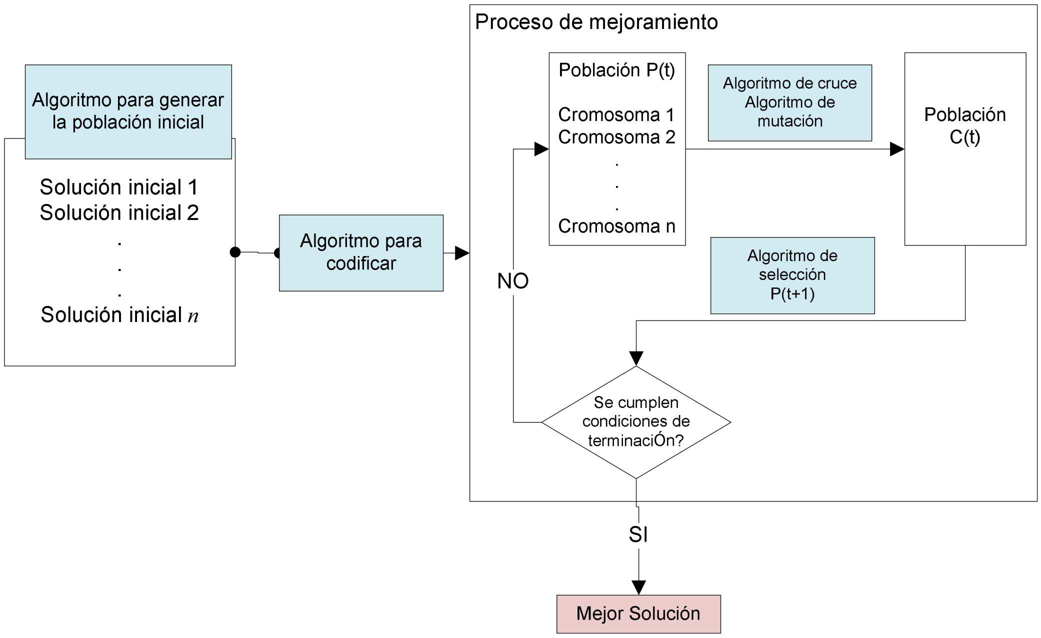


Figura 2.2: Diagrama general de un Algoritmo Genético Básico

Algoritmos genéticos celulares.- El modelo celular imita la evolución natural desde el punto de vista del individuo. Mediante este paradigma se generan poblaciones con una estructura especial, donde cada individuo interactúa sólo con sus vecinos más cercanos; si representamos a la población de un AGc por medio de un grafo, obtendríamos una red donde cada vértice se conecta con el resto mediante una estructura definida. De esta manera, cuando se genera una nueva población cada individuo interactúa solo con su vecindad (neigbordhood) y no con todos los elementos de la población como se realiza en un AE clásico.

2.2.2.1 Ventajas del uso de los Algoritmos Genéticos (AG).

Según Mitsuo Gen, Runwei Cheng y Lin Lin en su libro "Network Models and Optimization Multiobjective Genetic Algorithm Approach" [15], las principales ventajas del uso de los Algoritmos Genéticos son:

1. Adaptabilidad. Los AG no implican el uso de requerimientos matemáticos complejos para los problemas de optimización; pueden utilizar cualquier tipo de

Función Objetivo y cualquier tipo de restricciones lineales o no lineales que incluyan variables discretas o continuas.

2. Robustez. El uso de operadores y procedimientos evolutivos para la evaluación y selección de las nuevas soluciones garantiza efectividad en búsqueda de soluciones globales y no como las metaheurísticas tradicionales que se centran en búsquedas locales. Está probado por muchos estudios que AG es más eficiente y más robusto en localizar la solución óptima y en reducir los esfuerzos computacionales que otras heurísticas convencionales.

3. Flexibilidad. El AG provee una gran flexibilidad para combinarse con otras heurísticas para implementaciones eficientes en la solución de problemas específicos.

2.2.2.2 Estructura de un Algoritmo Genético

En la tabla 2.2 se expone un AG básico, el cual, en cada iteración t, se compone de tres procedimientos principales: uno de búsqueda, reproducción y otro de selección. En la etapa de búsqueda se determinan los individuos de la población P_t que tienen mayor aptitud y por lo tanto serán los que participen en la fase reproductiva, a este conjunto se les denomina "padres".

La reproducción consiste de dos operaciones básicas: combinación y mutación. Mediante la combinación (o cruce), un par de padres mezclan sus genes y producen un nuevo individuo; es así, que se genera el conjunto C_t compuesto de todos estos nuevos elementos (offspring). En la mutación se trata de mejorar la aptitud del nuevo individuo mediante algún procedimiento de modificación de sus genes.

En la fase de de selección, se determinan los nuevos individuos que sobreviven a la siguiente etapa, es decir que componen la población P_{t+1}. El AG termina de iterar cuando se ha alcanzado la condición de parada definida en su estructura.

```
BEGIN /* Algoritmo Genético básico */
        t←0
        Po←generar_poblacion_incial()
        evaluacion(Po).
        /* Fase de reproducción y generación de una nueva población*/
        WHILE (No se alcancen las condiciones de finalización) DO
                /* Producir nueva generacion */
                padres←seleccion_padres(Pt)
                Ct← combinacion(padres).
                Ct←mutacion(Ct).
                evaluacion (Ct).
                P t+1←seleccion_nueva_poblacion(Pt,Ct).
                t←t+1
        END
END
```

Tabla 2.2: Pseudocódigo de un algoritmo genético básico

A continuación se abordan algunas características propias de la estructura de un Algoritmo Genético en la implementación para resolver problemas de optimización.

Representación de los individuos.

Existen varias maneras de representación de los individuos según el problema de Optimización Combinatoria a resolver. En general la codificación utilizada (cromosoma) debe captar las características de una solución (fenotipo). Los principales tipos de representación son:

- Binaria.- donde cada gen del cromosoma es un bit (0 o 1).
- Entera.- donde cada gen del cromosoma es un entero.
- Real.- donde cada gen es un número real.
- Permutación.- donde cada gen es un número entero pero no puede repetirse en el cromosoma.

Aptitud de cada individuo

La aptitud o fitness es una medida de la calidad de cada individuo; en los problemas de optimización combinatoria la aptitud es la evaluación del **fenotipo** de cada individuo en la función objetivo.

Tamaño de la población

El tamaño de la población es uno de los parámetros del modelo. Parece intuitivo que un tamaño muy pequeño no permitiría explorar las soluciones sobre un conjunto diverso, por otro lado un tamaño muy grande podría significar mayor costo computacional y disminuiría la eficiencia de un AG. Con base a su experiencia, **Alander** sugiere que un tamaño de población comprendida entre 1 y 21 es suficiente para atacar con éxito los problemas por él considerados [14]. Lo cierto que es que el tamaño de la población debe de ser suficiente para garantizar la diversidad de las soluciones.

Generación de la población inicial

Generalmente la población inicial se genera de manera aleatoria; se podría además incluir algún tipo de Heurística en la generación de P_0; en las investigaciones existentes se evidencia una convergencia relativamente más rápida de este tipo de implementación, sin embargo, podría ser que conlleve a un resultado de óptimo local [14].

Selección de padres.

En un AG no estructurado, todos los individuos pueden ser seleccionados como padres de manera aleatoria; para ello se calcula la probabilidad de selección, la cual por lo general es proporcional a la aptitud (fitness). El método de la ruleta es el más común, donde la probabilidad se calcula de la siguiente manera [14]:

$$P(h)=\frac{Aptitud\ (h)}{\sum_{i=1}^{N} Aptitud(h_i)} \qquad (2.1)$$

Según este método, los individuos con mayor aptitud se escogerán varias veces por generación, mientras que los de pobre aptitud sólo serán seleccionados de vez en cuando [12].

Una alternativa al esquema de la ruleta, es el método del **ranqueo**, el cual consiste en hacer una selección según el rango k que ocupa el individuo; para ello, se ordenan los individuos de acuerdo a su aptitud y se le asigna una función de probabilidad proporcional a su rango. El procedimiento lineal usa la siguiente función de probabilidad: $P(h) = \alpha + \beta k$ donde α y β son constantes y k es el rango del individuo.

En los AG donde la población tiene una estructura distribuida los padres son seleccionados de manera aleatoria dentro de cada subpoblación; mientras que en los AG con estructura celular cada individuo se combina con otro seleccionado de entre sus vecinos más próximos.

Criterio de parada

De lo revisado en la literatura, se puede expresar que la regla más común para la condición de terminación es la convergencia del algoritmo genético o un número prefijado de iteraciones.

2.2.2.3 Aplicación de Algoritmos Genéticos para resolver el VRPTW

Según la literatura sobre AG, se han desarrollado muchas aplicaciones para resolver el problema de ruteo (VRP), respondiendo cada una de ellas a las consideraciones asumidas según el problema específico, tales como inclusión de ventanas de tiempo, capacidad o demanda dinámica. Joe L. Blanton Jr [10] fue el primero en usar Algoritmos Genéticos para resolver el problema VRPTW en 1993; Luego, en 1995, Thangiagh [16] usó AG para implementar su método denominado GIDEON, el cual asigna primero clientes a las rutas y luego define el ruteo en cada grupo ("Cluster first, route second"). En la primera década del 2000 se han desarrollado algunas investigaciones al respecto tales como la de Kenni Qili Zhu[17] donde propone un algoritmo que prueba ser mejor a GIDEON en todas las 56 instancias de Solomon. Una de las últimas publicaciones (2008) es la de Alba y Dorronsoro [1] donde implementan un Algoritmo Genético celular para resolver no precisamente un VRPTW sino un CVRP, logrando mejores resultados que otras técnicas.

De lo revisado en estas publicaciones se puede concluir que un AG básico no logra buenas aproximaciones a la solución óptima. Toda implementación exitosa de un AG para resolver problemas VRP se convierte en un Algoritmo Genético Híbrido porque requiere algún tipo de adición algorítmica tales como modificación de los operadores genéticos clásicos o el uso de heurísticas para búsqueda local; se pude expresar que estas aplicaciones tienen en común los siguientes elementos:

- Una representación adecuada de los cromosomas (cada cromosoma debe representar una posible solución al VRP).
- Un método para generar la población inicial.
- Una función para evaluar la aptitud de cada individuo y un adecuado criterio de selección.
- Rediseño de los operadores de mutación y cruce.
- Inclusión de búsqueda local para la mejora de las soluciones.
- Definición de los parámetros del algoritmo: Tamaño de la población, probabilidad del uso de operadores, etc.

Entre las investigaciones destacadas están las Homberger y Gehring [23] quienes proponen dos estrategias evolutivas híbridas que aplican operadores de mutación y combinación, utilizando heurísticas clásicas de mejora, tales como el 2-Opt*, modificadas para adaptarlas a las restricciones de tiempo del VRPTW.

2.2.2.3.1 Construcción de la población inicial

La construcción de la población inicial típicamente se la realiza de manera aleatoria; por ejemplo, estrategias aleatorias fueron implementadas por Blanton (1993) y Rahoual et (2001).

Es lógico pensar que el método empleado para la construcción de la población inicial debe garantizar la diversidad de los individuos; por otro lado, este hecho debe contrastarse con las restricciones propias del problema, lo que se puede equivaler a un conjunto de pobres soluciones si todo se lo deja al azar; de aquí que, en algunas aplicaciones para resolver el problema de ruteo se aplican heurísticas para la construcción de las soluciones iniciales; por ejemplo, Zhu[17] utiliza una heurística de inserción para generar una parte de la población inicial, siendo el resto de individuos definidos de manera completamente aleatoria.

2.2.2.3.2 Combinación o cruce (Crossover).

La combinación es una operación que se efectúa entre cada par de padres seleccionados. Muchos autores en la literatura indican que es la operación principal en un AG. En las últimas décadas se han investigado muchos tipos de combinación aplicados a los problemas TSP y VRP, tales como Partially Matched Crossover (PMX), Cycle Crossover (CX) y el Edge Recombination Crossover (ERX), reportando el último mejor rendimiento en problemas de tipo CVRP [7 y 13].

ERX es un tipo de operador de cruce, el cual se aplica a individuos que tienen una representación de tipo permutación (como es el caso del TSP y VRP); la idea principal es preservar lo más posible las adyacencias (edges) de los padres. Para ilustrar esta idea supondremos que tenemos los siguientes padres:

Padre 1:

2	3	4	7	1	10	8	9	6	5

Padre 2:

1	2	3	6	10	9	4	5	8	7

Hijo:

2	1	7	3	4	5	8	6	9	10

Figura 2.3: Ejemplo para la aplicación del operador ERX

Supongamos el ejemplo de la figura 2.3; una vez construida la tabla de adyacencia (Tabla 2.3), el primer gen del hijo se selecciona de entre los primeros genes de los padres, se prefiere aquel con menor número de nodos adyacentes. En nuestro caso el gen inicial es 2 (el primero de P1); luego se elimina el 2 de la tabla y el siguiente nodo seleccionado es aquel con menor número de nodos adyacentes, en caso de un empate se lo rompe al azar; en nuestro ejemplo el 1 y el 3 son los nodos con menor número de adyacencias, ante lo cual aleatoriamente el 1 es seleccionado y se procede a borrarlo de la tabla; el procedimiento termina cuando todos los nodos han sido insertados en el nuevo individuo.

nodo	adyacentes			
1	7	10	2	
2	5	3	1	
3	2	4	6	
4	3	7	9	5
5	6	2	4	8
6	9	5	3	10
7	4	1	8	
8	10	9	5	7
9	8	6	10	4
10	1	8	6	9

Tabla 2.3: Ejemplo de tabla de adyacencia

Otros autores como Áslaug Sóley Bjarnadóttir [11] proponen operadores de combinación adaptados al VRP tales como el UC (Uniform Crossover) donde básicamente un hijo toma las mejores rutas del padre y de la madre.

2.2.2.3.4 Mutación

La idea de mutar es modificar a uno o varios de los genes de un individuo con el objetivo de introducir diversidad en la población. Existes muchas maneras de implementar la permutación, algunos sólo afectan a un grupo de genes sin alterar al resto, mientras que otras más complejas redefinen la estructura del individuo. Por lo general, la mutación puede ser de tres tipos: inserción, inversión o cambio (swap).

Las estrategias evolutivas de Homberger y Gehring 1998 [23] son adaptadas especialmente al problema VRPTW, donde la mutación no consiste en un solo movimiento, sino en un conjunto de operadores que son aplicados en *N* movimientos; por ejemplo, la primera estrategia, que sólo aplica la mutación, se realiza en dos fases:

- En cada movimiento se aplica una de las siguientes heurísticas de mejora: Or-opt, 2-Opt*, e intercambio.
- En la segunda fase de cada movimiento se trata de disminuir el número de rutas con el operador Or-Opt modificado, el cual elimina la ruta con menor número de

vehículos y los inserta en otra ruta considerando criterios para respetar las ventanas de tiempo y no producir atrasos.

2.2.2.3.5 Evaluación y selección

En investigaciones como la de Homberger y Gehring [23] se evidencia que la aplicación de Estrategias Evolutivas a problemas de ruteo requiere algunas consideraciones en la evaluación y selección; por ejemplo, si sólo se considera el costo de la ruta como criterio para evaluar la aptitud del individuo, se ha demostrado que el AG tiende a escoger soluciones con muchas rutas.

Se han desarrollado muchos esfuerzos con éxito para resolver esta dificultad; una forma de hacerlo es considerar una estrategia multiobjetivo como la desarrollada por Kursawe (1992) la cual consta de dos fases para la evaluación y selección:

- En la primera fase se hace un ordenamiento lexicográfico de los individuos candidatos para formar la nueva población, donde el primer criterio es el número de vehículos y el segundo el tamaño de la ruta más pequeña; bajo estas consideraciones se seleccionan los primeros κ individuos.

- En la segunda fase nuevamente los individuos candidatos son ordenados bajo dos criterios, primero se considera el número de vehículos, luego la distancia total recorrida, y se seleccionan los μ-κ individuos necesarios para completar la nueva población.

2.2.2.4 Heurísticas de búsqueda local utilizadas en el VRPTW

La experiencia de las investigaciones de las últimas décadas demuestra que para lograr soluciones efectivas del VRP, a través de AG, es necesaria la inclusión de Algoritmos de búsqueda local [5]; estos procedimientos pueden hacerse de dos maneras: para mejorar cada ruta, lo que equivale aplicar la búsqueda en un problema TSP a la vez o búsqueda local considerando todas las rutas a la vez.

Procedimientos de una ruta

r-Opt

Es una de las heurísticas de búsqueda local más utilizada. Este tipo de método básicamente consiste en remover r arcos de la ruta y buscar varias formas de reconectar los nodos afectados, finalmente se selecciona la mejor ruta; el algoritmo tiene complejidad de orden $O\ (n^r)$, por lo que explorar todo el vecindario resulta muy costoso. En la figura 2.4 se ilustra la aplicación del operador 2-Opt, donde sólo es posible una manera de intercambiar los arcos; se ilustra que la aplicación de 2-Opt invierte el sentido de visita de un segmento de la ruta lo cual puede provocar incumplimiento de las restricciones de tiempo y aumentar el costo total en un VRPTW.

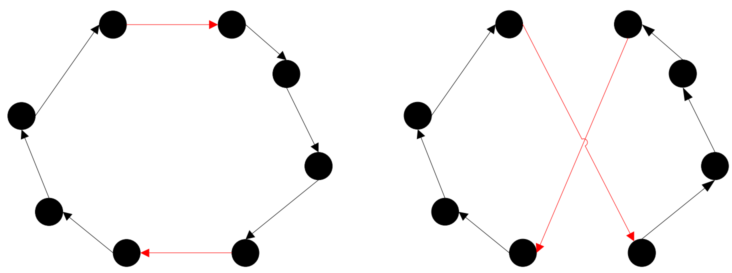


Figura 2.4: Operador 2-Opt. Se ilustra el único intercambio posible para los dos arcos

Operador Or-Opt

El operador Or-Opt traslada una cadena de clientes de longitud *k* de una ubicación a otra; concretamente consiste en mover una secuencia de *k* clientes consecutivos de una ruta a otra posición, de tal manera que permanezcan consecutivos y en el mismo orden. En la práctica, un intercambio de tipo Or-Opt donde $k>3$ es raramente usado, debido a su costo computacional, por lo que primero se realizan movimientos con $k=3$ y luego con un número menor.

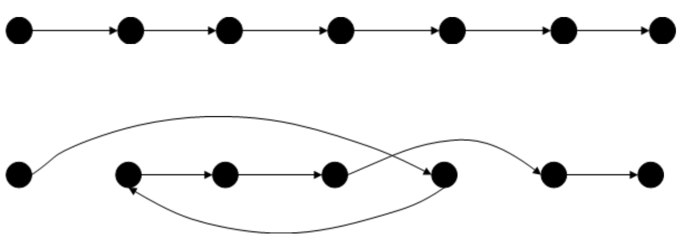


Figura 2.5: Intercambio Or-Opt con k=3

Procedimientos entre rutas

Operador 2-Opt*

2-Opt* es un operador entre rutas, consiste en intercambiar un segmento de una ruta con otro segmento de otra ruta teniendo en consideración el orden (Potvin); Sea el arco de la primera ruta seleccionada (x_i, x_{i+1}) y (y_j, y_{j+1}) el arco de la segunda ruta, entonces se desea mezclar ambas rutas de tal manera que se produzcan los nuevos arcos (x_i, y_{j+1}) y (y_j, x_{i+1}), de esta forma se mantiene el sentido de orientación de ambas rutas. La figura 2.6 ilustra un ejemplo de la aplicación del 2-Opt*. Como caso particular, este operador puede combinar dos rutas en una si el arco (x_i, x_{i+1}) es el primero de su ruta, y el arco (y_j, y_{j+1}) es el último de la otra ruta, o viceversa.

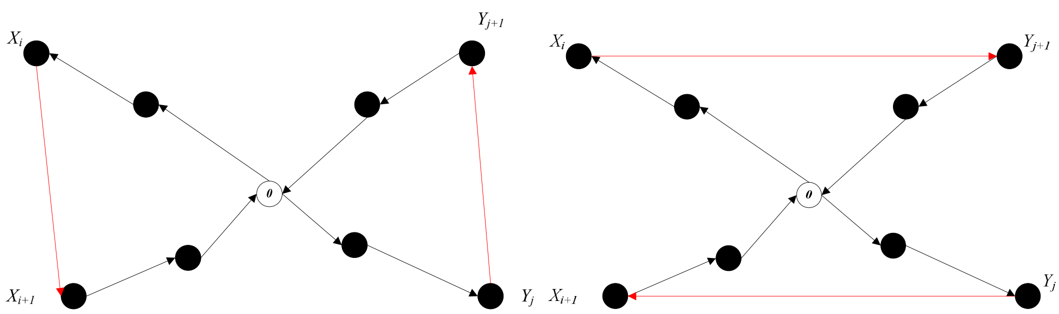


*Figura 2.6: Ejemplo de aplicación del operador 2-Opt**

Cruce (CROSSOVER).

Este operador también consiste en el intercambio de un segmento de ruta con otro segmento de otra ruta manteniendo el orden de visita; se diseñó en el contexto del VRPTW y fue definido por Taillard (1997)[22]; se puede decir que es una generalización del 2-Opt*. Específicamente, se seleccionan un par de arcos de ambas rutas, los cuales son eliminados y posteriormente se reconectan los segmentos de rutas restantes de tal manera que se no se altere el orden de visita. En la figura 2.7 se ilustra un ejemplo de la aplicación del operador de cruce, donde los arcos de (x_1, x_2) y (x_4, x_5) de la ruta *A* junto con los arcos (y_1, y_2) y (y_4, y_5) de la ruta *B* son reemplazados por los arcos (y_1, x_2) y (x_4, y_5) generando la ruta *A'* y por los arcos (x_1, y_2) y (y_4, x_5) en la nueva ruta *B'*.

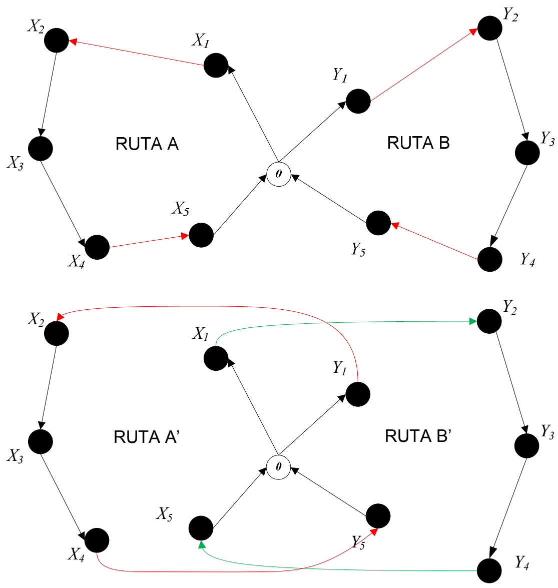


Figura 2.7: Ejemplo de aplicación del operador de cruce

Recolocación

La recolocación (Relocation) consiste en mover un cliente de una ruta a otra. Se selecciona un cliente xi de una ruta A y se lo reubica dentro de otra ruta B. En la figura 2.8 se ilustra un ejemplo aplicando esta operación.

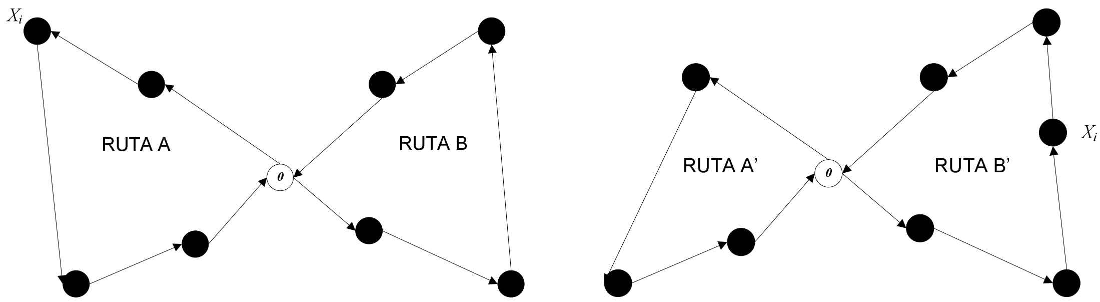


Figura 2.8: Ejemplo de recolocación

Intercambio

(Exchange) En este procedimiento se intercambian clientes asignados a rutas diferentes para generar el vecindario [Osman 93]. En la figura 2.9 se ilustra el intercambio del cliente X_i de la ruta A con el cliente Y_i de la ruta B produciéndo nuevas rutas.

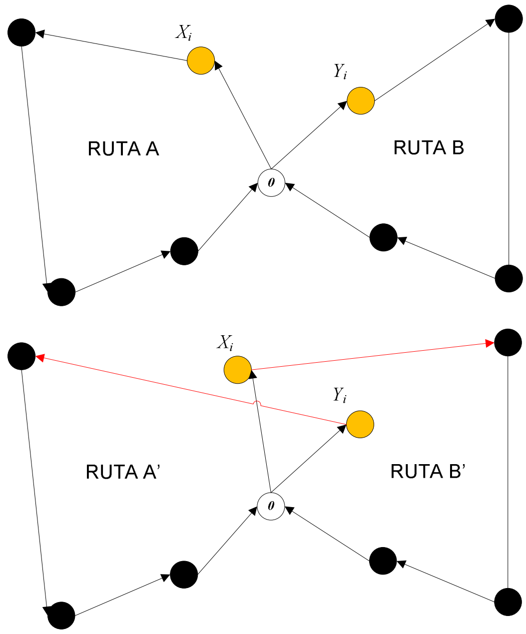


Figura 2.9: Ejemplo de intercambio

2.3 Desarrollo de la Metaheurística para el CVRPTW

Según lo analizado en la revisión teórica, y con el objetivo de que el algoritmo propuesto sea factible de poder desarrollarlo, se fijaron los siguientes postulados para el diseño de la heurística:

- Se define una estrategia evolutiva (μ, λ)-EA, es decir a medida que se va generando la nueva élite, el hijo generado reemplaza a uno de los padres siempre y cuando tenga una mejor aptitud.

- El AG propuesto será sencillo pero válido, por lo tanto no se considera un algoritmo con población estructurada.

- Se considera la mutación como un mecanismo de generación y exploración de vecindades (Osman 1995), que opera en las características de las rutas (nivel fenotipo), basada en la eliminación y reinserción de clientes; para tal fin, la mutación tendrá un conjunto de operadores de búsqueda local.

- Se considera la combinación como un mecanismo de generación de nuevas soluciones (Osman 1995), que opera entre individuos (nivel genotipo); para tal objetivo, se fija un operador de cruce basado en Uniform Crossover [11].

- Para la evaluación y selección se considera tanto la minimización del número de rutas como la minimización del costo total de ruteo.

- Para definir criterios de inserción, en las heurísticas de búsqueda local, se consideran las restricciones de ventanas horarias.

Lo anterior evidencia que la Metaheurística propuesta para resolver el problema CVRPTW es un Algoritmo Genético Híbrido, que consiste en la aplicación de heurísticas de búsqueda local adaptadas a las características propias de las ventanas horarias y guiadas por los principios evolutivos que rigen un AG; En la tabla 2.4 se expone en pseudocódigo el AG a desarrollar.

```
PARAMETROS

POP_SIZE            //  tamaño de la población
ELITE_SIZE          //  número de hijos a generar en cada iteración
PM                  //  probabilidad de mutación
NUM_ITERACIONES     //  número máximo de iteraciones como condición de parada

INPUT
Matriz de datos del problema: coordenadas de cada cliente y del depósito,  ventanas de tiempo, demanda.

OUTPUT
Mejor solución encontrada

BEGIN /* Algoritmo Genético híbrido */
        cargar_datos()
         t←0
         P0←generacion_poblacion_incial()
         /* Fase de reproducción y generación de una nueva población*/
         WHILE (t ≤ NUM_ITERACIONES) DO
                  evaluacion_poblacion (Pt)
                  Pt← mutacion_padres(Pt)
                  Pt←generar_hijos(Pt)
                  t←t+1
         END
        Entregar la mejor solución
END
```

Tabla 2.4: Algoritmo Genético propuesto para resolver el problema CVRPTW

A continuación se describen cada uno de los componentes del AG Híbrido a ser implementado:

- **cargar_datos ( ):** este procedimiento se encarga de leer los datos del problema a resolver y guardarlos en matrices. Una vez inicializado el algoritmo estos datos se convierten en parámetros y son los siguientes:
  - **VÉRTICES [ ]:** número de vértices del problema de ruteo.
  - **VENTANAS_TIEMPO [ ]:** tiempo más temprano y más tardío de atención aceptado por cada cliente.
  - **DEMANDA [ ]:** demanda de cada cliente.
  - **CAPACIDAD:** capacidad de cada vehículo, la cual se considera homogénea según lo expuesto en el capítulo 1.
  - **NUM_VEHÍCULOS:** Número de vehículos que componen la flota.

- **COSTOS [ ] [ ]:** Matriz de costos de desplazamiento entre los vértices.

- **generación_población_inicial** ( ): este procedimiento inicializa cada individuo de la población, es decir construye un conjunto de soluciones iniciales al problema VRPTW. Se intentará primero generar el 100% de los individuos aleatoriamente y según la experimentación se incluirá una heurística de inserción para inicializar una parte de la población.

- **evaluacion_población ():** este procedimiento calcula el fitness de cada individuo que conforma la población P_t y luego los ordena según un orden lexicográfico.

2.3.1 Estrategia de mutación

Según el algoritmo descrito en la tabla 2.4 el operador de mutación se aplica a todos los individuos de la población, con una probabilidad *PM*, antes del proceso reproductivo, de esta manera se garantiza la mejora de todos los elementos de P_t. Para adaptar el operador de mutación al problema CVRPTW se ha desarrollado el procedimiento expuesto en la tabla 2.5, el cual es basado en una de las estrategias evolutivas propuestas por Homberger y Gehring [23], la cual tiene una estructura multiobjetivo que logra mejorar la solución (disminuir el costo de ruteo) y minimizar el número de rutas cada vez que se aplica la mutación sobre un individuo.

De lo expuesto en la tabla 2.5, se observa que la estrategia de mutación consiste en dos fases: primero se realizan *N_MOV* movimientos de mejora aplicando una de las heurísticas del conjunto {**2-Opt*, recolocación, intercambio};** luego se intenta minimizar el número de rutas, para lo cual se aplica el operador Or-Opt con el objetivo de eliminar la ruta que tenga mayor costo por cliente.

Cabe expresar que todos los operadores que se emplean en la mutación deben estar adaptados a las restricciones de tiempo del problema CVRPTW; las heurísticas de búsqueda local consideradas (2-Opt*, recolocación, intercambio y Or-Opt) básicamente consisten en eliminar clientes de las rutas donde se encuentran y reinsertarlos en otras rutas o posiciones diferentes. De aquí que es de suma importancia el criterio de inserción de un cliente en una ruta, el cual se define en la sección 2.3.6.

```
PARAMETROS
   N_MOV    //   número de movimientos de mejora
   PM            //  probabilidad de mutación
INPUT
  Individuo
OUTPUT
  Individuo mutado
BEGIN /* Algoritmo para la mutación */
  r= aleatorio(0,1);
  IF(r<PM)
        Leer solución S correspondiente al individuo a mutar
        FOR i=1 TO  i ≤ N_MOV
        /* Fase de mejora de la solución*/
               Seleccionar un operador del conjunto de operadores de mejora
               Generar la nueva solución S' aplicando el operador seleccionado
               IF ( costo(S') <costo(S))
                   S=S'
               END
        /* Fase de minimización de rutas*/
               Seleccionar la ruta que tenga mayor costo por cliente
               IF (Si es factible eliminar la ruta)
                   Eliminar la ruta aplicando el operador Or-Opt considerando las restricciones de
        tiempo
               END
        END
  END
END
```

Tabla 2.5: Algoritmo para mutación

2.3.2 Estrategia de combinación

Para seleccionar el operador de combinación, se realizaron intentos con el operador ERX obteniendo malos resultados, luego se mejoró el rendimiento diseñando un operador crossover adaptado especialmente al CVRPTW. Considerando que la combinación de dos individuos debe procurar rescatar lo mejor de ambos padres, se definió una estrategia basada en el operador Uniform Crossover (UC) descrito por Áslaug Sóley Bjarnadóttir [11]. El procedimiento para realizar la combinación se expone en la tabla 2.6.

Según el algoritmo descrito en la tabla 2.6 se define el costo por cliente como criterio de calidad para cada circuito, el cual se calcula dividiendo el costo total de la ruta para el número total de clientes visitados. Según este criterio, luego se intenta insertar las mejores rutas en el nuevo individuo, siempre y cuando no produzcan conflictos, es decir que una ruta candidata a ser insertada no tenga elementos comunes con las otras rutas que ya forman parte del hijo. El algoritmo termina examinando si existen clientes no incluidos en alguna ruta, en caso afirmativo son insertados en la posición que menor costo produzca.

```
PARAMETROS
    NUM_CLIENTES // Total de clientes

INPUT
    Padre, Madre
OUTPUT
    Hijo
BEGIN
    Calcular costo por cliente en cada ruta del Padre y Madre.
    Ordenar las rutas de los padres de manera creciente (según costo por cliente).
    NUM_RUTAS_HIJO=min{NUM_RUTAS_PADRE, NUM_RUTAS_MADRE}
    J=0
    WHILE  (J< NUM_RUTAS_HIJO OR (se examinen todas las rutas de los padres)) DO
      Seleccionar la mejor ruta no examinada de los padres.
      IF NOT (existe conflicto con resto de rutas)
        Insertar ruta seleccionada en el hijo
        J←J+1
      END
    END

    FOR  i=1 TO i≤NUM_CLIENTES
      IF (i ∉ {rutas del hijo})
        Insertar cliente i en la ruta del hijo que tenga menor costo
    END

END
```

Tabla 2.6: Algoritmo para la combinación

2.3.3 Representación Cromosómica

Se adoptará la representación de permutación de enteros, donde cada arreglo define el orden de visita de los clientes. Existen 1,2,… n clientes, los cuales se permutan en un solo arreglo; se definen como separadores de rutas los enteros n+1, n+2,….

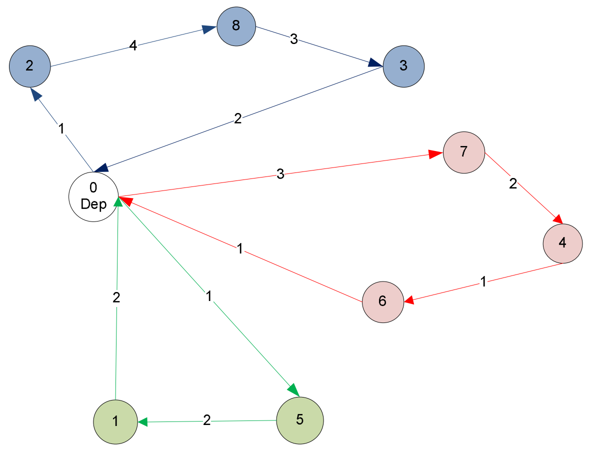


Figura 2.10: Ejemplo de rutas

Por ejemplo, en la figura 2.10 se ilustra una solución a un VRP, la cual necesita tres vehículos, entonces, un cromosoma que representa este ruteo se muestra a continuación; la primera ruta (2→8→3) se encuentra separada de la segunda (7→4→6) y ésta de la tercera (5→1) por un número mayor que *n*.

Cromosoma:

2	8	3	9	7	4	6	10	5	1

Para completar la información requerida en el CVRPTW se define un vector donde se guarda el tiempo de arribo a cada vértice y otro con la penalidad por violación de las ventanas de tiempo.

2.3.4 Cálculo de tiempos y penalidades

Para la determinación de los tiempos, a partir de los cuales se calcularán los costos y criterios a incluir en los operadores de búsqueda local, se utiliza la notación a continuación expuesta y definida en el capítulo uno:

- La matriz de costos $\mathbf{C}_{n\times n} = (c_{ij})$ que calcula la distancia o tiempo desde cada par de vértices (v_i, v_j).
- $[e_i, l_i]$ ventana horaria aceptada por el cliente *i*.
- s_i, el tiempo de servicio para el cliente *i*.

Determinación del tiempo de inicio de servicio

El tiempo de inicio de servicio t_i se calcula por:

$$t_0=e_0 \quad \text{para el depósito} \quad (2.2)$$

$$t_i=max\left\{t_{(i-1)}+s_{(i-1)}+c_{(i-1),i},\ e_i\right\} \quad \text{Para } i\geq 1 \quad (2.3)$$

Es decir si el vehículo arriba antes del inicio del servicio se produce una espera.

Determinación del tiempo de partida más temprano

El tiempo de partida más temprano δ_i [23] se calcular por:

$$\delta_0= e_0 \quad \text{para el depósito} \quad (2.4)$$

$$\delta_i=t_i+s_i \quad \text{Para } i\geq 1 \quad (2.5)$$

Determinación del tiempo de arribo factible máximo

Considerando que el tiempo de arribo a un cliente puede afectar la factibilidad en los tiempos de los sucesores, es necesario redefinir el tiempo de arribo (Solomon 1988) [23] de la siguiente manera:

$$\alpha_0=l_0 \quad \text{para el depósito} \quad (2.6)$$

$$\alpha_i=min\{\alpha_{i+1}-c_{i,i+1}-s_i,\ l_i\} \quad \text{Para } i\geq 1 \quad (2.7)$$

Determinación de la espera y atraso

De la ecuación (2.5) y (2.6) se deduce que el tiempo de espera en el cliente *i* se calcula por:

$$\Delta_0^- = 0 \qquad \text{Para el depósito} \quad (2.8)$$

$$\Delta_i^- = max\{t_i - t_{(i-1)} + s_{(i-1)} + c_{(i-1),i}, 0\} \qquad \text{Para } i \geq 1 \quad (2.9)$$

Por otro lado se podría producir una violación a la restricción del tiempo máximo de servicio, y en este caso se produciría un atraso determinado por la siguiente fórmula:

$$\Delta_i^+ = 0 \qquad \text{Para el depósito} \quad (2.10)$$

$$\Delta_i^+ = max\,\{t_i - l_i, 0\} \qquad \text{Para } i \geq 1 \quad (2.11)$$

2.3.5 Evaluación y Selección

La aptitud de cada individuo (Fitness) se define como la evaluación de la solución en la función objetivo, la cual comprende un componente de costo más una penalización por violación de las ventanas de tiempo.

Dados los siguientes elementos:

- El conjunto de vértices $V = \{v_0, v_1, \dots, v_n\}$ donde $v_0 = 0$ representa el depósito.
- El conjunto $P = \{v_1, \dots, v_n\}$ formado de los *n* clientes a ser visitados.
- $P_1 | P_2 | .. | P_m$ una partición de *P*,
- La ruta $R_k = \{v_0^k, v_1^k, \dots, v_r^k, v_{(r+1)}^k\}$, donde $v_0^k = v_{(r+1)}^k = 0$ y $\{v_1^k, \dots, v_r^k\}$ es una permutación del subconjunto P_i

Entonces el costo de la ruta R_i viene dado por:

$$Costo(R_k) = \sum_{j=0}^{r} (c_{j,(j+1)} + s_j) \qquad (2.12)$$

Y la penalización de la ruta R_i por violar las ventanas de tiempo se define por:

$$pen\_atraso(R_k) = \alpha \sum_{j=1}^{r+1} \Delta_j^+ \qquad (2.13)$$

Donde α un factor de penalización.

Por lo tanto el costo total de la solución correspondiente a un individuo, compuesto de *m* rutas, en el Algoritmo Genético propuesto se calcula por:

$$Costo\ total = \sum_{k=1}^{m} \left(C_k + Costo(R_k) + pen\_atraso(R_k) \right) \qquad (2.14)$$

Donde C_k es un valor correspondiente a cada ruta *k*, que se lo puede interpretar como el costo fijo asumido cada vez que un vehículo entra a operar.

2.3.6 Criterio para la inserción de un cliente en una ruta

Como los operadores de búsqueda local aplicados durante la mutación básicamente consisten en movimientos de eliminación e inserción, es necesario definir un criterio de inserción de un cliente dentro de una ruta. Se ha elegido el criterio usado por Homberger y Gehring [23] el cual se describe a continuación.
Se define el atraso producto de la inserción del cliente *k* dentro de la ruta *R* entre los clientes *i* e *i*+1 como:

$$D_k(i,i+1) = V1_k + V2_k \qquad (2.15)$$

Donde:

$$V1_k = max\ \{\delta_i + c_{i,k} - l_k, 0\} \qquad (2.16)$$

$$V2_k = max\left\{\max\{\delta_i + c_{i,k}, e_k\} + s_k + c_{k,i+1} - \alpha_{i+1}, 0\right\} \qquad (2.17)$$

Se puede observar que $V1_k$ mide el atraso o violación de la ventana de tiempo del cliente *k* al ser insertado después del cliente *i*; mientras que $V2_k$ mide el atraso o

violación de la factibilidad de arribo en el cliente i+1 cuando k es su predecesor inmediato, considerando el tiempo factible máximo del cliente i+1.
En caso que la inserción del cliente k dentro de la ruta R viole la restricción de capacidad, entonces $D_k(i, i+1) = +\infty$.

2.3.7 Descripción de la heurística de construcción de rutas

Para lograr soluciones iniciales factibles, que no produzcan sobre todo retraso en el servicio, se incluye una heurística de construcción, la cual se usará para generar una parte de la población inicial. El algoritmo se describe en la tabla 2.7 y se basa en un procedimiento del vecino más cercano con criterio de inserción de máximo ahorro y sin incurrir en atrasos en el servicio.

Como se observa en la tabla 2.7, para seleccionar el siguiente vértice a insertar en la ruta R_k, se inspeccionan los clientes no incluidos en el ruteo, ordenados según la distancia al centroide de la ruta; es decir, se trata de seleccionar entre los más cercanos. Si no es posible encontrar una inserción factible se crea una nueva ruta con cualquiera de los vértices no incluidos y el procedimiento se repite hasta que todos los clientes sean visitados.

En la implementación del algoritmo de la tabla 2.7, se deberá verificar que no se incumplan los topes máximos de las ventanas de tiempo, es decir que $t_j < \alpha_j$, en caso de que se incumpla esta condición se procederá a intentar la inserción en otra posición.

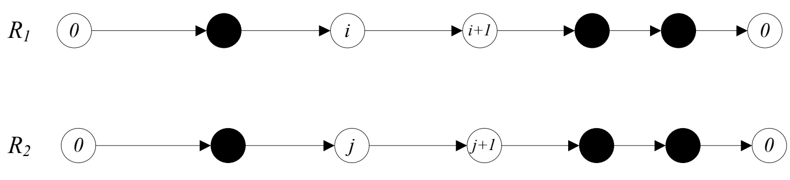
```
PARAMETROS
P        // conjunto de los N clientes a ser visitados
V        //  conjunto de los clientes que no han sido incluidos en una de las rutas
NUM_VEHICULOS    // Total de vehículos disponibles
BEGIN /* heurística de construcción de rutas */
```

```
inicio←aleatorio(N) ;// se selecciona aleatoriamente un vértice de inicio
    k←0;
    V← P;
    R_k.insertar(inicio) ;//se inserta inicio en la ruta k
    WHILE (V≠∅) DO
        R_k.calcular_centroide();
        Ordenar V de manera ascendiente de acuerdo a la distancia al centroide de R_k;
        costo_min←M;
        v←0;
        FOR i=1 TO |V|
            costo_insercion= calcular_costo_insercion(k);
            IF ( no se incumple restricción de capacidad ) AND costo_insercion< costo_min AND
            (no se produce espera) THEN
                costo_min= costo_insercion;
                    v=i;
            END
        END
        IF v≠0 THEN
            R_k.insertar(v);
        ELSE
            IF (k<NUM_VEHICULOS)
                k←k+1;
                R_k.insertar(v);
            ELSE
                Se inserta v en la ruta y posición que tenga menor costo de inserción
            END
        END
        V ← V −{v};
    END
END
```

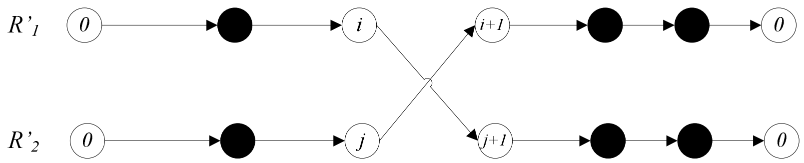
Tabla 2.7: Heurística de construcción

2.3.8 Descripción de la implementación de la heurística de búsqueda local 2-Opt*

La búsqueda local consiste en lograr entre cada par de rutas (que componen una solución) un intercambio de tipo 2-Opt*. En la figura 2.11 se expone una vez más la aplicación del 2-Opt*, donde se ilustra que esta operación equivale a segmentar dos rutas R_1 y R_2 en los puntos *i* y *j* respectivamente, para luego reconectar los segmentos de rutas generados pero de manera inversa.



a) *Antes del intercambio*



b) Luego del intercambio

*Figura 2.11: Implementación del operador 2-Opt**

```
INPUT
        Rutas R1, R2
OUT PUT
       Rutas R1, R2 mejoradas
BEGIN
            mejor_i←0; // punto de corte de la ruta 1
            mejor_j←0; // punto de corte de la ruta 2

            costo_actual←0;
            /* Se aplica el operador 2-Opt* para un par de rutas R1 y R2 */
            R1.calcular_costo();
            R2.calcular_costo();
            costo_actual ← R1.costo+ R2.costo;
            ahorro←0;

            FOR i=1 TO (R1.total_clientes-1)
                FOR j=1 TO (R2.total_clientes-1)
                     IF  (no se incumple restricción de capacidad ) AND (no hay retrasos en los clientes i+1 y
                        j+1 ) THEN
                           costo_nuevo←calcular_costo_nuevas_rutas(R1, R2 , i, j);
                           IF   costo_actual > costo_nuevo THEN
                                 ahorro← costo_actual -costo_nuevo;
                               mejor_i←i;
                               mejor_j←j;
                            END
                     END
                 END
            END
            IF ahorro>0 THEN
                hacer_intercambio(R1, R2 , mejor_i, mejor_j);
            END IF
END
```

*Tabla 2.8 Heurística de búsqueda local 2-Opt**

El algoritmo de implementación se describe en la tabla 2.8. Es importante, antes de seleccionar los puntos de corte, verificar que no se irrespete la restricción de capacidad del modelo y que no se produzcan retrasos en los clientes i+1 y j+1, que son los primeros afectados por el intercambio. El algoritmo requiere un procedimiento para calcular los costos en los segmentos de rutas generados.

Se implementa una función que entregue el costo de ruta desde un nodo *i* hasta el fin de la ruta (se debe tomar en cuenta que el servicio inicia en un tiempo t_i que sería el parámetro de la función.

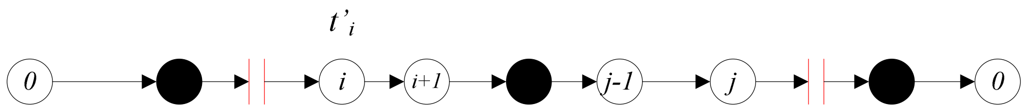


*Figura 2.12: Inserción de un segmento (i, j) en una ruta según 2-Opt**

Suponiendo que se ha segmentado una ruta desde el cliente *i* al cliente *j*, y que existe un nuevo tiempo de inicio de servicio en al nodo *i* denominado t_i', entonces el costo del segmento será:

$$costo\_segmeno(i, j, t_i') = \sum_{l=i}^{j-1} c_{l,l+1} + \sum_{l=i}^{j} \Delta_j^+ \qquad (2.18)$$

2.3.9 Determinación de los parámetros del algoritmo

Los valores de los parámetros del algoritmo, tales como tamaño de la población (POP_SIZE), número de hijos a generar en cada iteración (ELITE_SIZE) , probabilidad de mutación (PM) y número de iteraciones como condición de parada (NUM_ITERACIONES), serán fijados por el método experimental con múltiples combinaciones una vez implementado el procedimiento propuesto.

Capítulo 3

3. Implementación de la Metaheurística y acercamiento conceptual del DSS

3.1 Introducción

En el presente capítulo se exponen los lineamientos planteados para implementar la Metaheurística propuesta en el capítulo 2, así como la documentación del componente de software desarrollado.

A pesar de existir librerías y programas que resuelven problemas de optimización combinatoria usando AG, no se usa ningún software computacional ya desarrollado para la implementación; por el contrario, se prefiere codificar el algoritmo desarrollando un sistema propio con el objetivo de explorar y comprender los detalles implícitos. Para empezar la codificación de la aplicación, se revisó la estructura de algunas librerías realizadas por otros investigadores, principalmente la colección de clases de Java JCell [24] y la encontrada en el foro de "república digital de Colombia" [25], las cuales brindaron una idea inicial de cómo empezar la programación.

El objetivo principal de esta investigación es el desarrollo del AG híbrido (Metaheurística propuesta) y no tiene por fin el desarrollo informático del DSS; sin embargo, se propone en este capítulo una implementación alternativa con el uso de la API Google Maps.

3.2 Definición del lenguaje de codificación.

3.2.1 Necesidad del paradigma orientado a objetos para implementar un AG.

Según Enrique Alba [9] hay que tomar en cuenta las siguientes consideraciones, entre otras, para la aplicación de Algoritmos Genéticos:

- Toda implementación de AG se traduce en sistemas de software complejos que necesitan de un diseño unificado y estructurado.
- Se requiere un sistema modular de software para facilitar la constante monitorización de valores y el estado del algoritmo.
- Todos los componentes del sistema evolutivo, tales como procedimientos y operadores requieren trabajar en cualquier orden entre sí y con la inclusión de heurísticas no evolutivas. Esto requiere que los subsistemas componentes, empleados en cada iteración, presenten una interfaz clara y flexible.

En términos computacionales estos requisitos son satisfechos por la programación orientada a objetos (POO), la cual descompone un sistema de software en subsistemas llamados módulos. En un programa orientado puramente a objetos sus componentes se denominan “clases”, y su diseño se define por las relaciones de sus clases.

3.2.2 Elección del lenguaje de programación para la implementación del AG.

Entre los lenguajes de programación orientados a objetos, usados para implementar los algoritmos genéticos, los más utilizados son C++ y Java. Se ha demostrado computacionalmente que C++ es mucho más rápido que Java, por ser un lenguaje de primer nivel; sin embargo, Java tiene otras bondades que C++ no las posee: nos permite crear y manipular más fácilmente objetos gráficos y sus aplicaciones como al applet corren de manera directa en entornos web y dispositivos móviles.

Existen algunas librerías de Algoritmos Genéticos entre ellos se encuentra GAME, RPL2. GAME y JCell, que al ser muy genéricos su implementación y adaptación a un problema específico resulta extremadamente tediosa; por otro lado RPL2 es más completo pero sin embargo tiene licencia comercial.

Por las razones arriba mencionadas, se eligió al lenguaje de programación Java para la implementación y no se utiliza una librería genérica ya desarrollada, por el contrario, se prefiere hacer una codificación propia.

3.2.3La Programación Orientada a Objetos (POO) y el lenguaje Java.

Un sistema bajo el paradigma de la Programación Orientado a Objetivos (POO) se reduce a la definición y estructuración de sus clases; un Objeto es una instancia de una Clase, por ejemplo si se considera la clase “Humano”, una instancia correspondiente sería “Juan”. Cada clase tiene un conjunto de atributos y métodos. Se puede decir que un atributo es una propiedad que define las características de los objetos; por ejemplo, un atributo de la clase “Humano” puede ser “sexo”, otro “peso” y otro “altura”. Los métodos son las acciones que puede ejecutar un objeto; en términos computacionales serían los procedimientos que operan sobre alguna propiedad del objeto o que permiten interactuar con otras entidades del sistema tales como otras clases o bases de datos.

Existe otra definición importante considerada en la POO llamada “Herencia”, la cual permite que un objeto hijo reutilice la funcionalidad ya definida en un objeto padre; por ejemplo una vez estructurada la clase “Mamífero” se puede definir la clase hija “Perro” quien heredaría todas las características comunes a los mamíferos sin ser necesario volver a definirlas.

Sun Microsystems desarrolló a principios de los años 90 el lenguaje de programación orientado a objetos Java. Es un lenguaje basado en la sintaxis de C y C++, pero usa un modelo más simple y elimina herramientas de bajo nivel. Java permite ser independiente de la plataforma, es decir un programa puede correr en varios sistemas operativos sin necesidad de alterar su código fuente [26]. Bajo este enfoque todo en

Java está dentro de una clase, y el código fuente se guarda en archivos con el mismo nombre que contienen y con extensión ".java".

Gracias a la creación de ciertas especificaciones, se pueden desarrollar aplicaciones Java para dispositivos móviles (Java 2 Platform, Micro Edition), navegador web (Java Script), sistemas de servidor (Servlets y Java Server Pages) y tipo escritorio (Java Runtime Environment JRE) [21]. Sun también ofrece el JDK (Java Development Kit) que incluye el JRE, el compilador de Java y el javadoc usado para generar documentación.

3.3 Descripción del sistema

3.3.1 Descripción general de las librerías

La implementación de la Metaheurística consistió en el desarrollo de dos librerías. La primera, denominada "ag", agrupa los componentes principales de la estrategia evolutiva y comprende las clases: algoritmo genético, problema, población, individuo, y ruta. La segunda librería, comprende el conjunto de operadores genéticos, y las heurísticas de búsqueda local y construcción de rutas.

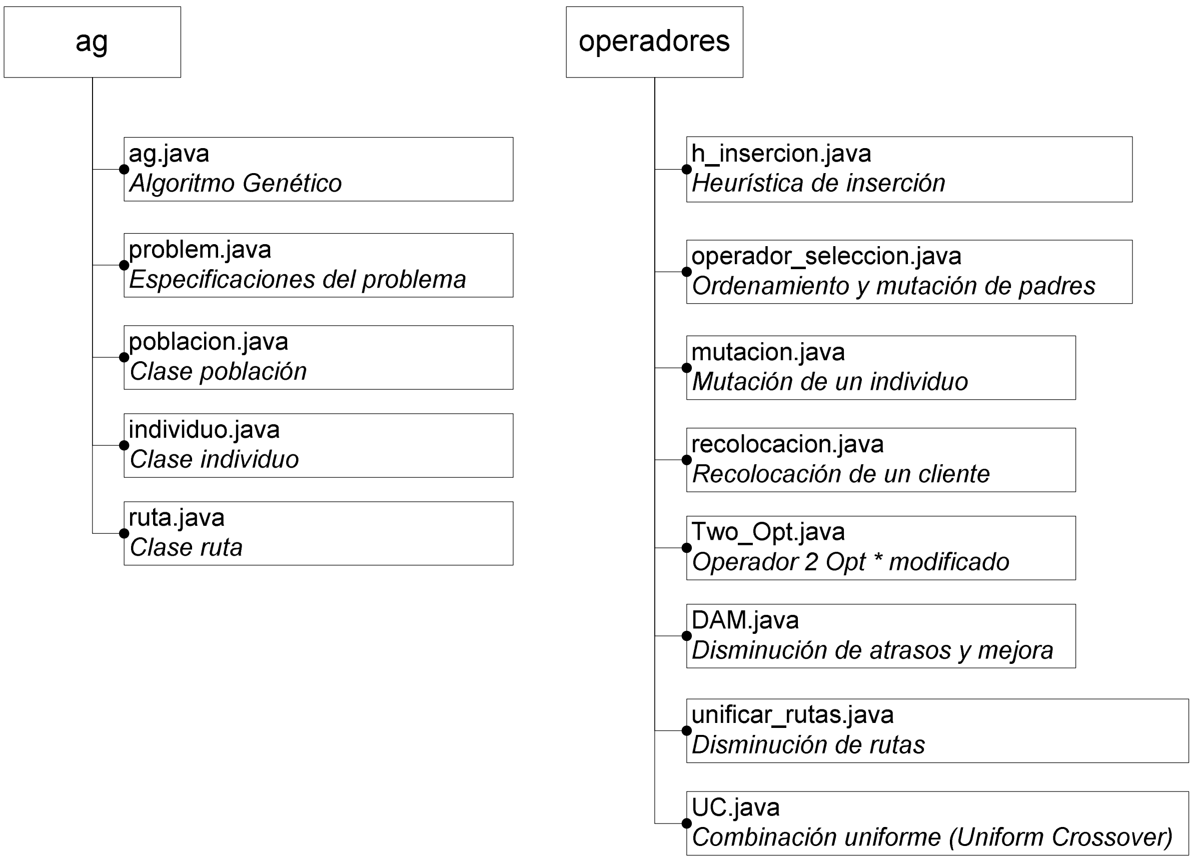


Figura 3.1: Librerías del sistema

El procedimiento principal "main" que ejecuta el programa se encuentra en el archivo "vrptw.java", y tiene por objetivos: crear un objeto de la clase "ag", cargar los datos del problema y ejecutar el algoritmo genético.

3.3.2 Descripción de las clases.

3.3.2.1 Clase ag.java

Descripción: contiene un objeto población y su principal método consiste en ejecutar la Metaheurística según el algoritmo descrito en la sección 2.3. Sus atributos se detallan en la tabla 3.1.

Nombre:	Tipo:	Descripción
tasa_mutacion	double	probabilidad de mutación
tasa_cruzamiento	double	probabilidad de combinación
tam_elite	Int	número de hijos generados en cada iteración
pob	poblacion	Objeto población

Tabla 3.1: Atributos de la clase ag.java.

Métodos:

- public **ag** (problema p): constructor de la clase.
- public void **ejecutar** ( ): ejecuta el algoritmo genético.

3.3.2.2 Clase problem.java

Descripción: Carga los datos y define los parámetros del problema. Sus atributos se detallan en la tabla 3.2.

Métodos:

- Public **problem** (String *url*).- Constructor de la clase problema, también llama a los procedimientos **cargar_datos**(String *url*) y **cargar_matriz_costos** ( ); el parámetro *url* indica la dirección donde se encuentra el archivo de datos.

- Public void **cargar_datos** (String *url*). - Carga los datos del archivo *url* e inicializa los vectores de tiempo, demanda y vértices.

- Public void **cargar_matriz_costos** ( ).- Calcula los costos entre cada par de clientes y llena la matriz respectiva incluido el depósito.

Nombre:	Tipo:	Descripción
vertices[]	array int	Vector que contiene el total de vértices considerados.
tiempo_inicio []	array double	Vector con los tiempos de inicio de la ventana horaria de cada cliente.
tiempo_fin[]	array double	Vector con los tiempos de fin de la ventana horaria de cada cliente.
demanda[]	array double	Vector con la demanda de cada cliente.
tiempo_servicio[]	array double	Vector con los tiempos de servicio de cada cliente.
costo[][]	array double	Matriz [cij] con el costo de desplazamiento del cliente i al cliente j.
TAMANO_POBLACION	int	Tamaño de la población
TAMANO_ELITE	int	Número de hijos a generar en cada iteración.
pen_espera	double	Penalidad en caso de espera
pen_atraso	double	Penalidad en caso de atraso
costo_vehiculo	double	Costo fijo por vehículo
CAPACIDAD	double	Capacidad del vehículo (considerando flota homogénea)
NUM_ITERACIONES	int	Máximo número de iteraciones (Criterio de parada)

Tabla 3.2: Atributos de la clase problem.java

3.3.2.3 Clase poblacion.java

Descripción: Define la estructura de la población junto con los mecanismos que operan sobre ella en cada iteración; en la tabla 3.3 se exponen los atributos de esta clase.

Nombre:	Tipo:	Descripción
cromosomas[]	array *individuo*	Vector de individuos
pop_size	int	tamaño de la población
elite_size	int	número de hijos generados en cada iteración

Tabla 3.3: Atributos de la clase población.java.

Métodos:

- Public **población** ( ).- Constructor de la clase.

- Public void **generar_población** ( ).- Genera la población inicial, usando un objeto de la clase *h_insercion*.

- Public void **evaluar** ( ).- Evalúa la población P_t y usa una instancia de la clase *operador_seleccion* para mutar y ordenar a los individuos de la población *t*.

- Public void **generar_elite** ( ).- Genera los hijos a partir de la población P_t y reemplaza al peor de los padres si el nuevo individuo es mejor; utiliza una instancia de la clase *UC* para realizar la combinación.

- Public int [ ] **selección** ( ).- Selecciona dos de los mejores individuos de la población como padres para generar un hijo.

3.3.2.4 Clase individuo.java

Descripción: Contiene el conjunto de genes que componen el individuo, así como las rutas que comprenden la solución asociada junto con los costos respectivos. Los atributos de esta clase se exponen en la tabla 3.4.

Nombre:	Tipo:	Descripción
alelos[]	Array Integer	Arreglo de genes que componen el cromosoma del individuo
len	int	Longitud del cromosoma
fitness	double	Fitness del individuo
rutas[]	Array *ruta*	Vector con las rutas de la solución
total_costo	double	Costo total de la solución
total_espera	double	Tiempo total de espera
total_atraso	double	Tiempo total de atraso
total_costo_ruta	double	Costo total sólo del ruteo
sobre_capacidad	double	Violación a la restricción de capacidad en unidades

Tabla 3.4: Atributos de la clase individuo.java

Métodos:

- public void **individuo** ( ).- Constructor de la clase.

- public void **inicializar** ( ).- Inicializar el vector *alelos* [ ], inserta los clientes y separadores de rutas en orden ascendente.

- public void **orden_aleatoreo** (Random *r*).- Ordena aleatoriamente los elementos del vector *alelos*[].

- public void **evaluar** ( ).- Evalúa los costos de la solución y calcula el *fitness* del individuo.

- public boolean **eliminar_elemento** (Integer *element*).- Elimina un elemento (*element*) del vector *alelos* [ ].

- public void **hacer_rutas** ( ).- Dado el vector *alelos* [ ] construye las rutas de las solución (decodificación).

- public void **hacer_cromosoma** ( ).- Dadas las rutas de la solución construye el vector *alelos* [ ] (codificación).

- public void **ordenar_rutas** ( ).- Ordena las rutas de acuerdo al costo por cliente de manera ascendente.

3.3.2.5 Clase ruta.java

Descripción: Define la secuencia de visita de los clientes en una ruta, permite calcular los costos del ruteo, los tiempos generados en las visitas y si existe o no violación a las restricciones de tiempo o capacidad. En la tabla 3.5 se exponen los atributos de esta clase.

Nombre:	Tipo:	Descripción
clientes	ArrayList<Integer>	Lista con los clientes, según el orden de visita en la ruta.
tiempo_arribo[ ]	Array double	Vector que contiene el tiempo de arribo a cada cliente.
tiempo_atraso[ ]	Array double	Vector que contiene al atraso (violación a la ventana de tiempo) de cada cliente.
tiempo_partida[ ]	Array double	Vector que contiene el tiempo de partida desde cada cliente (una vez realizado el servicio).
tiempo_inicio[ ]	Array double	Vector que contiene el tiempo de inicio del servicio de cada cliente
costo_total	double	Costo total de la ruta
total_espera	double	Total tiempo de espera del vehículo.
total_atraso	double	Total atraso en la entrega del servicio de la ruta.
costo_ruta	double	Costo de viaje en la ruta
x	double	Coordenada x del centroide de la ruta
y	double	Coordenada y del centroide de la ruta
num_cliente	double	Número de clientes visitados en la ruta
costo_cliente	double	Costo total por cliente en la ruta

Tabla 3.5: Atributos de la clase ruta

Métodos:

- public int **size**() .- Retorna el tamaño de la ruta , incluido el depósito.

- public void **calcular_costos** ( ).- Calcula los costos de la ruta.

- public Integer **get_vertice**(int i).- Devuelve el vértice que se encuentra en la posición i.

- public void **insertar** (Integer *element*).- Inserta un vértice al final de la lista de clientes.

- public void **insertar** (int *posicion*, Integer *element*).- Inserta un cliente (*element*) en una posición específica en la lista de clientes.

- public void **eliminar** (int *posicion*).- Elimina el elemento (*element*) que se encuentra en una posición específica.

- public void **eliminar_elemento**(Integer *element*).- Elimina un elemento (*element*) de la lista de clientes sin especificar la posición.

- Public void **intercambiar** (int *i*, int *j*).- Intercambia los clientes que se encuentran en la posición *i* y *j*.

- public void **calcular_tiempos**().- Calcula el tiempo de arribo, atraso, inicio de servicio, partida y espera de cada cliente y los guarda en los vectores respectivos.

- public void **calcular_centroide**().- Calcula el centroide de la ruta.

- public double[] **calcular_costo_inserción** (int *element*, int *indicador*).- Calcula el costo y la posición de inserción de un nuevo elemento en la ruta; si el parámetro *indicador* es 1 entonces no permite que se genere un atraso, caso contrario si permite atraso; retorna un vector de dos dimensiones, con el costo y la posición de inserción en la ubicación 0 y 1 respectivamente.

- public boolean **verificar_existe** (Integer *element*).- Verifica si existe un elemento en la ruta; retorna verdadero si existe y falso caso contrario.

3.3.2.6 Clase h_inserción.java

Descripción: Implementa la heurística de inserción para generar soluciones iniciales al problema de CVRPTW según el algoritmo expuesto en la sección 2.3.7. En la tabla 3.6 se observan los atributos de esta clase.

Nombre:	Tipo:	Descripción
R	Random	Objeto aleatorio
mas_cercanos[]	Array int	Vector con los clientes má cercanos a una posición específica (x,y)
incluido[]	Array int	Vector que indica si un cliente i se encuentra o no incluido en una ruta.
Ind	individuo	Objeto de tipo individuo que se construye de acuerdo a las rutas generadas.

Tabla 3.6: Atributos de la clase h_insercion.java

Métodos:

- public void **calcular_mas_cercanos**(double *x*, double *y*).- Calcula la distancia al centroide de una ruta (*x*,*y*) de cada uno de los clientes que todavía no han sido asignados a un vehículo; luego los ordena de manera ascendente (según la distancia) y los guarda en el vector *mas_cercanos*[].

- public boolean **insertar_en_ruta(**int *element*, int *k*).- Trata de insertar el nodo *element* en la ruta *k*, sin que se produzca infactibilidad, donde *k* depende de la iteración del algoritmo de inserción; devuelve *true* si se logra insertarlo, caso contrario retorna *false*.

- public boolean **inserta_en_resto_rutas**(int *element*, int *k*, int *modo*).- Trata de insertar el nodo *element* en alguna ruta ya creada y diferente de *k*; si modo es igual a 1 permite la violación de las ventanas de tiempo, y si tiene otro valor no permite la existencia de atrasos; devuelve *true* si logra insertarlo, caso contrario retorna *false*.

- public individuo **ejecutar** ( ).- Ejecuta el algoritmo de inserción para la creación de rutas, según el algoritmo expuesto en la tabla 2.7; devuelve un objeto *individuo* con las rutas creadas.

3.3.2.7 Clase operador_seleccion.java

Descripción: Este operador permite aplicar la mutación a los individuos que componen la población, al mismo tiempo que los ordena de acuerdo al criterio establecido; en la tabla 3.7 se exponen los atributos de esta clase.

Nombre:	**Tipo:**	**Descripción**
r	Random	Objeto aleatorio

Tabla 3.7: Atributo de la clase operador_seleccion.java

Métodos:

- public static Object **ordenar**(individuo *ind*[ ]).- Ordena un vector de objetos de tipo *individuo*, ingresado como parámetro, usando el número de rutas como primer criterio de orden descendente y el costo de la ruta como segundo criterio de orden descendente; retorna el vector ordenado.

- public static Object **mutar_padres**(individuo *padres*[ ]).- Aplica el operador de mutación a cada objeto de tipo *individuo* del vector *padres*[], pasado como parámetro; retorna el vector de individuos mutados y ordenados de acuerdo al método **ordenar**().

3.3.2.8 Clase mutacion.java

Descripción: Dado un individuo aplica la estrategia de mutación descrita en la sección 2.3.1. En la tabla 3.8 se exponen los atributos de esta clase.

Nombre:	Tipo:	Descripción
R	Random	Objeto aleatorio.
Ind	individuo	Objeto de tipo individuo.

Tabla 3.8: Atributos de la clase mutacion.java

Métodos:

- public **mutación** (*individuo in*).- Constructor de la clase mutacion.java, tiene como parámetro el objeto de tipo *individuo* sobre el cual se aplica la estrategia de mutación.

- public *individuo* **ejecutar** ().- Ejecuta el algoritmo de mutación, usando los operadores: recolocación, 2-Opt* modificado, *DAM* (disminución de atrasos y mejora) y unificación de rutas; retorna el individuo mutado.

3.3.2.9 Clase recolocacion.java.

Descripción: Aplica el operador de recolocación descrito en la sección 2.2.2.4. En la tabla 3.9 se exponen los atributos de esta clase.

Nombre:	Tipo:	Descripción
R	Random	Objeto aleatorio.
Ind	individuo	Objeto de tipo individuo.

Tabla 3.9: Atributos de la clase recolocacion.java

Métodos:

- public **recolocación** (individuo *in*).- Constructor de la clase recolocacion.java, tiene como parámetro el objeto de tipo *individuo* sobre el cual se aplica la heurística de mejora.

- public int[] **seleccionar** ().- Selecciona el nodo que será recolocado; retorna un vector de longitud 2, que contiene la ruta y posición del elemento; escoge siempre el cliente que tiene mayor costo dentro de las rutas.

- Public int[] **calcular_inserción** (int *p*, int *k*).- Determina la mejor ruta donde se debe insertar el elemento *p*; el parámetro *k* indica la ruta donde se encuentra el cliente en un inicio; retorna un vector de longitud 2, que contiene la ruta y posición donde se debe recolocar el cliente.

- public *individuo* **ejecutar** ( ).- Ejecuta el algoritmo de recolocación, primero selecciona el individuo a reubicar, lo elimina de la ruta donde se encuentra, y luego lo reinserta en otra ruta con el objetivo de minimizar el costo; retorna el individuo con el cliente recolocado.

3.3.2.10 Clase two_Opt.java

Descripción: implementa el operador 2-Opt* modificado que se detalla en la sección 2.3.8. En la tabla 3.10 se expone el único atributo que tiene esta clase.

Nombre:	Tipo:	Descripción
ind	individuo	Objeto de tipo individuo.

Tabla 3.10: Atributo de la clase two_Opt.java

Métodos:

- public **two_Opt**(individuo *in*).- Constructor de la clase two_Opt.java, tiene como parámetro el objeto de tipo *individuo* sobre el cual se aplica la heurística de mejora.

- public void **ejecutar_2Opt**(int *k*, int *m*).- Intercambia dos secciones de las rutas *k* y *m* de tal manera que el costo conjunto sea el mínimo posible.

- public *individuo* **ejecutar** ( ).- Ejecuta el algoritmo descrito en la tabla 2.8; retorna el individuo modificado sobre el cual se aplicó la estrategia de mejora.

3.3.2.11 Clase DAM.java

Descripción: implementa una estrategia para la disminución de atrasos y mejora (DAM); la cual consiste en seleccionar aquellos clientes con mayor atraso, eliminarlos de las rutas donde se encuentran y reinsertarlos en otra posición de menor costo; es decir, su objetivo es eliminar las infactibilidades de tiempo producidas a medida que se aplica el algoritmo genético. Los atributos de esta clase se exponen en la tabla 3.11.

Nombre:	Tipo:	Descripción
ind	individuo	Objeto de tipo individuo.
clientes[ ] [ ]	Array int	Matriz con los datos de los clientes que tienen atraso.

Tabla 3.11: Atributos de la clase DAM.java

Métodos:

- public **DAM** (individuo *in*).- Constructor de la clase DAM.java, recibe un parámetro de tipo *individuo* sobre el cual se aplica la heurística de mejora.

- public void **seleccionar_atrasos**().- Determina los clientes donde hay atrasos y los ordena de mayor a menor según este criterio guardándolos en la matriz *clientes*[][].

- public individuo **ejecutar** ( ).- Ejecuta la estrategia DAM, selecciona los clientes que tienen atrasos y los reinserta en otra ruta o posición con el objetivo de disminuir la infactibilidad de tiempo; en caso estrictamente necesario crea una nueva ruta; retorna el individuo con la solución mejorada.

3.3.2.12 Clase unificar_rutas.java

Descripción: Tiene por objetivo disminuir el número de rutas, y se utiliza en la estrategia de mutación descrita en la sección 2.3.1.

Nombre:	Tipo:	Descripción
ind	Individuo	Objeto de tipo individuo.

Tabla 3.12: Atributos de la clase unificar_rutas.java

Métodos:

- public **unificar_rutas**(individuo *in*).- Constructor de la clase unificar_rutas.java, recibe un parámetro de tipo *individuo* sobre el cual se aplica la heurística de mejora.
- public int **seleccionar_ruta** ( ).- Selecciona la ruta a eliminar, para tal fin retorna la ruta que tiene mayor costo por cliente.

- public boolean **reinsertar** (Integer *element*, int *ruta_actual*, int *modo*).- Reinserta un elemento en otra ruta diferente a la actual, siempre y cuando el intercambio produzca ahorro de costo; devuelve *true* si logra reinsertar y *false* caso contrario.

- public individuo **ejecutar** ( ).- Selecciona la ruta a eliminar y uno por uno trata de reinsertar los clientes de ésta en otra ruta en la posición que produzca menor costo.

3.3.2.13 Clase UC.java

Descripción: Implementa la estrategia seleccionada para la combinación de individuos Uniform Crossover (UC) descrita en la sección 2.3.2.En la tabla 2.13 se exponen los atributos de esta clase.

Nombre:	Tipo:	Descripción
Padre	individuo	Objeto de tipo individuo.
Madre	individuo	Objeto de tipo individuo.
Hijo	individuo	Objeto de tipo individuo.

Tabla 2.13: Atributos de la clase UC.java

Métodos:

- public **UC** (individuo P_1, individuo P_2).- Constructor de la clase y requiere como parámetros dos objetos de la clase individuo P_1 y P_2, es decir los padres que participan en la combinación.

- public boolean **verificar_conflicto**(ruta *r*).- Verifica si la ruta *r* tienen elementos comunes con las rutas ya creadas e insertadas en el individuo hijo; retorna true si existe conflicto, caso contrario devuelve false.

- public *individuo* **ejecutar** ( ).- Ejecuta el algoritmo descrito en la tabla 2.6.

3.3.3 Relación entre las clases.

Para describir las relaciones entre las clases se usa el diagrama de tipo estático usado en el UML (Unified Modeling Language), estándar de documentación que emplea la programación orientada a objetos.

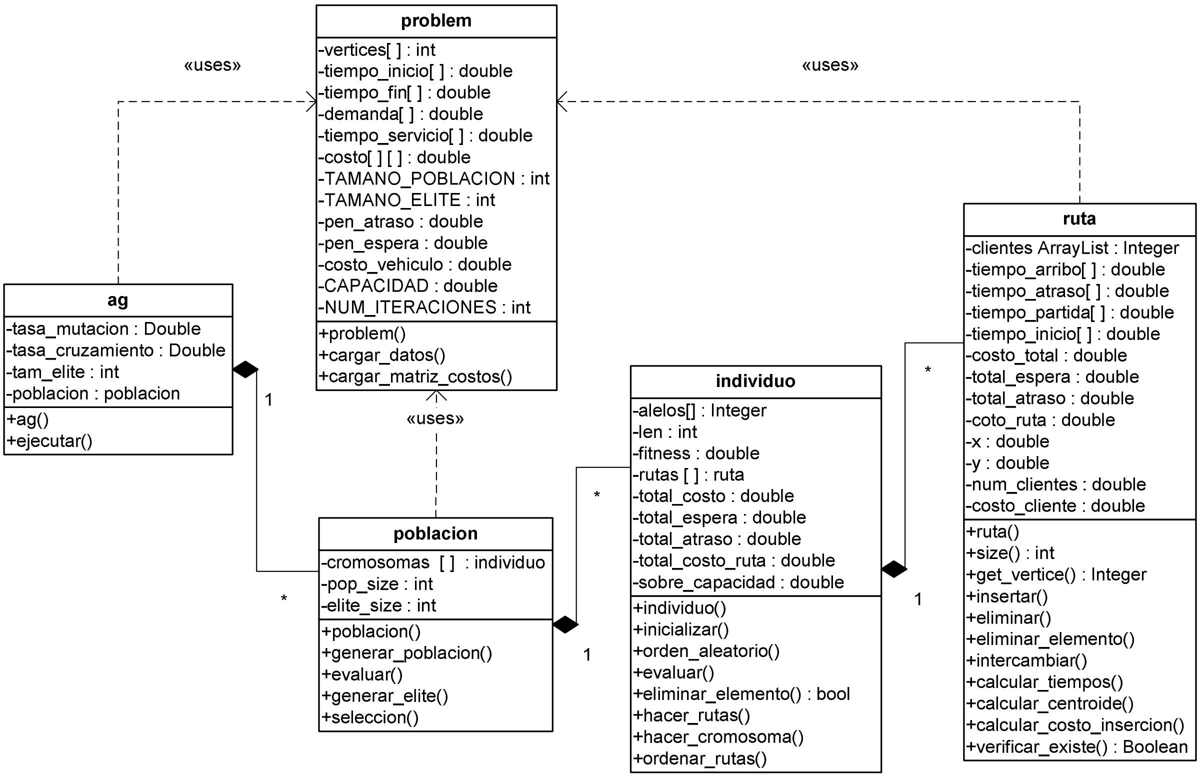


Figura 3.2: Relación entre las clases de la librería ag

Según la figura 3.2 ilustra que los atributos de la clase problem.java definen los parámetros del sistema y son usados por el resto de clases como variables globales, gracias a que son de tipo *public static*. Se observa que un objeto de la clase *ag.java* contiene una instancia de la clase *población.java*, a su vez ésta tiene como atributo un vector de objetos de tipo *individuo*, cuya clase contiene otro vector de tipo *ruta*.

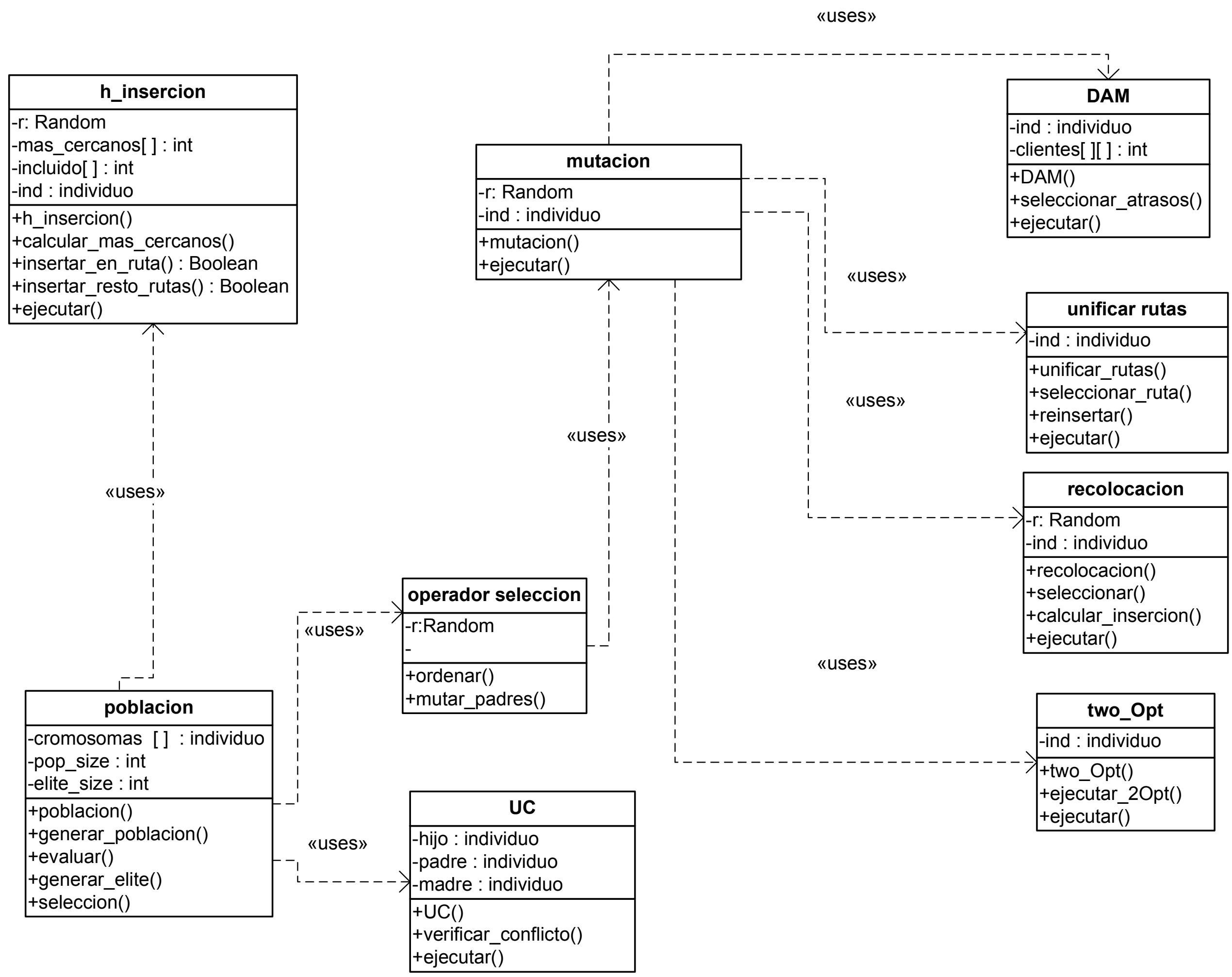


Figura 3.3: Relación de las clases de la librería operadores

Tal como se ilustra en la figura 3.3, la clase *población* usa instancias de la clase *h_inserción* para crear la mitad de la población inicial a través de la heurística de construcción descrita en la sección 2.3.7. La mutación de los individuos se efectúa a través de la clase *mutación* que utiliza instancias de las clases *DAM*, *unificar_rutas*, *recolocación* y *two_Opt*; este operador se aplica en la clase *operador_seleccion* a cada uno de los elementos de la población para luego aplicar el operador de combinación utilizando la clase *UC*.

3.4 Acercamiento a un Sistema de Soporte de Decisiones (DSS) para resolver el problema de ruteo según la Metaheurística desarrollada.

3.4.1 Sistema de información geográfica (SIG)

Un SIG es una herramienta de análisis de información, una integración organizada de hardware, software, procedimientos y datos geográficos, diseñado para capturar, administrar, analizar, modelar y graficar datos y objetos espacialmente referenciados.

Un paquete gráfico (CAD) por sí solo no constituye un SIG, debido al volumen y diversidad de información y los métodos de análisis especializados en un SIG. El sistema requiere también el recurso humano que planifica, desarrolla, administra y opera, caso contrario estará limitado.

El SIG funciona como una base de datos con información geográfica, asociada por un código común a los objetos gráficos de un mapa. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente.

La información de un SIG puede mostrarse en dos formatos: celular o raster y vectorial. El formato raster es la digitalización de los datos por medio de scanner, imágenes de satélite, fotografías, videos, etc.; el formato vectorial se representa por medio de segmentos orientados de rectas o vectores (pares ordenados de coordenadas).

Entre los análisis más importantes que un SIG debe realizar:

- Contigüidad: localizar áreas en una región determinada.
- Coincidencia: análisis de superposición de puntos.
- Enrutamiento: movimiento de un elemento a lo largo de una red.
- Radio de acción: alcance del elemento.
- Apareamiento: acople de información de direcciones con entidades gráficas.
- Análisis digital del terreno: análisis de la información de superficie para modelar fenómenos geográficos.

- Operación sobre mapas: uso de expresiones lógicas y matemáticas para el análisis y modelamiento de atributos geográficos.
- Geometría de coordenadas: operaciones para el manejo de coordenadas terrestres.

El abaratamiento y uso masivo de la tecnología GPS (Global Position System) integrada con equipos móviles (celulares, computadores, etc.), el desarrollo del internet y las redes de comunicación y los estándares OGC, han impulsado la tecnología web mapping y el surgimiento de numerosas aplicaciones que permiten la publicación de información geográfica a través de la web.

El mapeo web (web mapping) es el proceso de diseñar, aplicar, generar y visualizar datos geoespaciales en la world wide web. Dicha tecnología procura entregar los datos en formato GML (geographic markup lenguaje, sublenguaje para el modelamiento, transporte y almacenamiento de información geográfica), de acuerdo con las especificaciones del open geospatial consortium (OGC, grupo de organizaciones públicas y privadas creado en 1994), con el fin de conseguir una interoperación de los datos espaciales. Aunque el GML se presenta como el futuro lenguaje web para la visualización de mapas, en la actualidad todavía no es utilizado por la mayoría de los generadores de mapas.

3.4.2 Una alternativa a un SIG: las APIs de mapas on line

Una alternativa a la implementación de un SIG con licencia comercial puede ser el uso del servicio de mapas On line, como el que ofrece Google Maps, por ejemplo. Cabe mencionar que una tecnología de este tipo por sí sola no es un SIG, pero es válida para construir un DSS (Sistema de Soporte a Decisiones) para planificación de rutas a un bajo costo. A continuación se fundamenta este planteamiento partiendo de ciertas definiciones básicas necesarias para la comprensión de la propuesta.

Se puede expresar que una API (Aplication Programming Interface) es una interfaz de programación de aplicaciones y está compuesta por funciones y procedimientos (o métodos, en la programación orientada a objetos) incluidos en cierta biblioteca y que pueden ser utilizados por otro software en un capa de abstracción[3].

Google Maps es un servidor de aplicaciones de mapas en entorno web; esta tecnología es ofrecida por Google de manera gratuita y permite a los propietarios de websites integrarla a sus páginas web[4]. Expertos en desarrollo de sistemas georeferenciales como Turner[5] coinciden que este servicio no es un SIG, sino que básicamente se trata de una interfaz para el desarrollo de aplicaciones (API) dirigida al manejo de mapas en la web, de aquí que se denomina la API Google Maps.

Actualmente la API Google Maps es muy utilizada en aplicaciones logísticas on line por sus múltiples funcionalidades; por ejemplo, permite dibujar rutas tal como se ilustra en la figura 3.4. Para visualizar estos mapas en una aplicación web se debe obtener una clave proporcionada por Google totalmente gratuita; se utiliza código javascript para adicionar la interacción que necesite la aplicación.

Figura 3.4: Dibujo de una ruta en google maps6.

[3] *Definición de API según Wikipedia. http://es.wikipedia.org/wiki/API. Revisado el 7 de diciembre del 2009.*
[4] *Definición de Google Maps según Wikipedia. http://en.wikipedia.org/wiki/Google_Maps#Google_Maps_API. Revisado el 7 de diciembre del 2009.*
[5] *Andrew Turner. Is GoogleMaps GIS? High Earth Orbit. http://highearthorbit.com/is-googlemaps-gis/. Blog publicado el 12 de junio del 2009. Revisado el 7 de diciembre del 2009.*
[6] *Quikmaps. http://quikmaps.com/new revisado el 9 de diciembre del 2009.*

3.4.3 SIG integrado con DSS para ruteo, panorama local

En Ecuador ninguna empresa courier usa algún sistema de soporte de decisiones para planificar sus operaciones (picking / delivery) ni un SIG para el seguimiento "en línea" de su flota y planificación dinámica; aunque por aspectos de seguridad si utilizan servicios de localización en tiempo real (LBS) suministrados por diferentes operadores locales. Para entender mejor como un SIG puede capturar la información geográfica se define el esquema mostrado en la figura 3.5.

En Ecuador varias empresas usan programas de planificación de rutas, por ejemplo Sab Miller y Coca Cola usan Road Show para planificar su distribución capilar; sin embargo, luego de investigar, sólo una empresa utiliza DSS para ruteo y GIS, ambos desarrollados con personal propio, integrados entre sí con su programa de administración de flota y con su ERP:

La captura de datos para el SIG está tercerizada con un operador local que envía los datos al servidor de la empresa, luego ésta procesa los datos y los integra a los demás sistemas, de tal forma que pueden realizar una planificación semanal de la cantidad de camiones que se necesitarán, planificación diaria de entrega de pedidos, planificación de cada viaje (salida de planta, llegada estimada al cliente y retorno estimado a planta) y en tiempo real recalcula los tiempos (llegada al cliente y retorno a la planta); permite el seguimiento en línea de toda la flota, conocer velocidades, tiempo en ruta, paradas en ruta, aproximaciones, alarmas de "pánico". La integración del SIG con el DSS también permite planificar los viáticos que se entregan al conductor, entre otros beneficios.

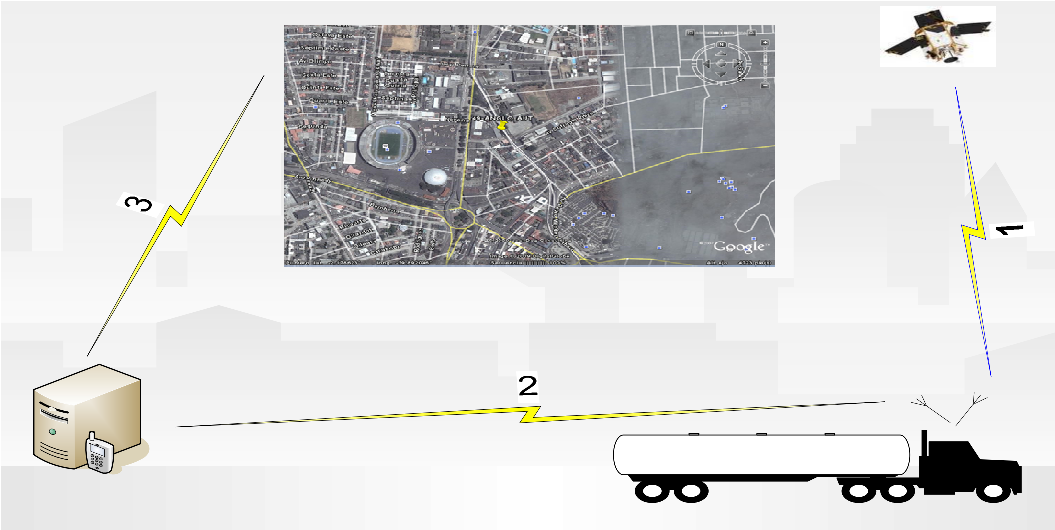


Figura 3.5: Una de las antenas del equipo GPS se conecta con el satélite, otra envía los datos al servidor, a través de un proveedor local, y el servidor procesa los datos

Cabe recalcar que los sistemas antes descritos resuelven un problema de planificación dinámica para distribución primaria con entrega y recolección, es decir, es un problema de ruteo diferente al que atañe a la presente investigación, sin embargo, es válido mencionarlo, y aunque esta empresa no tiene documentados sus estudios y experiencia con el desarrollo de estos sistemas, es la primera en desarrollar e integrar sistemas logísticos de este tipo, con marca y personal ecuatorianos, en lugar de adquirir algún paquete extranjero.

3.4.4 DSS propuesto para planificación de picking en un courier

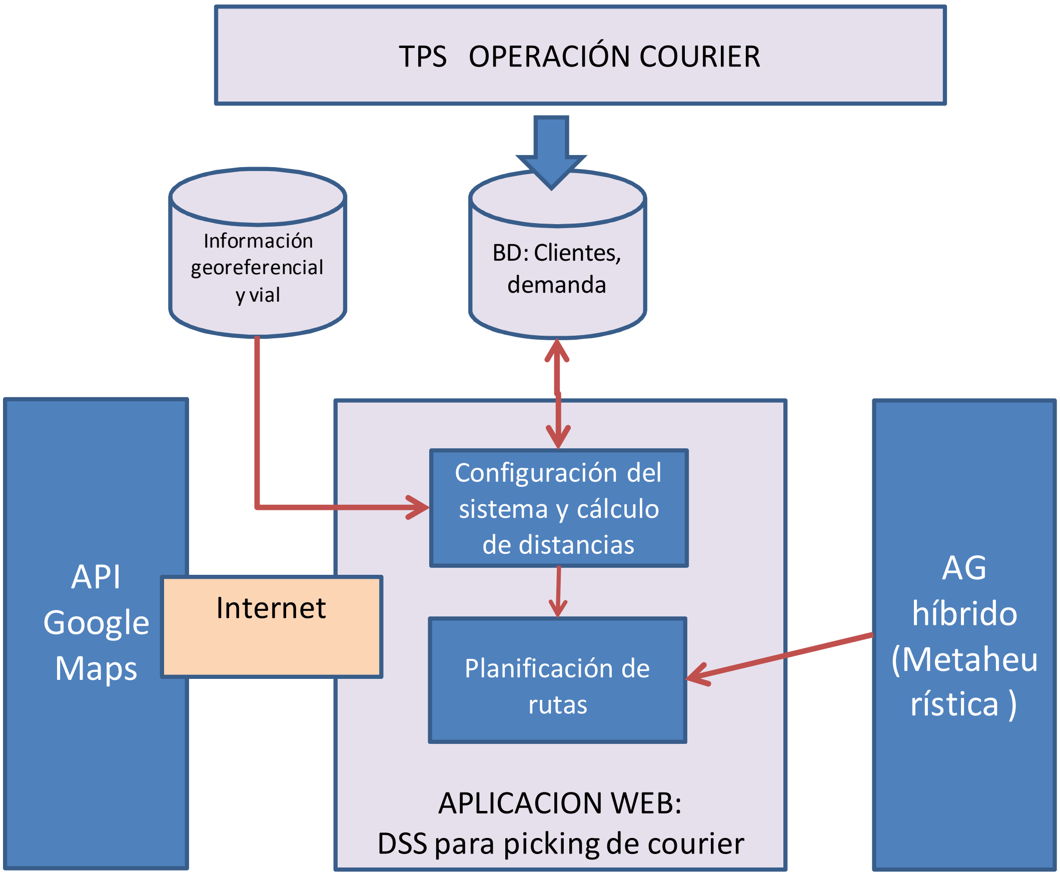


Figura 3.6: Esquema general del DSS para planificación del picking del courier

En la figura 3.6 se expone el sistema propuesto el cual tiene varios componentes; el principal es al Algoritmo Genético Híbrido; se requiere una interfaz entre el DSS con el TPS de la operación Courier que extraiga datos tales como la demanda, dirección de clientes y ventanas horarias de atención.

Para modelar la red geográfica donde se encuentran los puntos de picking, es decir abstraer este sistema a un grafo, se necesita información georeferencial como coordenadas GPS de la dirección de cada cliente, sentido de vías y estimación de la velocidad promedio de los vehículos. Toda esta información deberá ser procesada para estimar el tiempo de viaje entre cada par de clientes, a su vez, este dato es un input en el sistema de planificación de rutas.

Con el objetivo de ilustrar las rutas, resultado de la aplicación del AG híbrido, se utiliza la API Google Maps que mediante código javascript dibuja los diferentes circuitos de picking.

Capítulo 4

4. Validación de la Metaheurística propuesta

4.1 Introducción

En el presente capítulo se exponen los resultados de la validación del AG propuesto para la solución del CVRPSTW. Se desarrollan algunas pruebas con las instancias de Solomon, utilizadas para comprobar la efectividad de este tipo de métodos. De esta manera se intenta demostrar que el paradigma evolutivo, planteado en el capítulo 3, es efectivo para abordar este tipo de problemas de optimización combinatoria.

Un problema a abordar antes de realizar las pruebas es la configuración de los parámetros del AG propuesto, y la definición del criterio de parada del algoritmo, lo cual se trata en la siguiente sección.

4.2 Configuración de parámetros del algoritmo

Para poder verificar la efectividad y validar los métodos de solución de un problema VRPTW se utilizan las instancias de Solomon [27]. Cada una de estas pruebas consiste en un conjunto de 100 clientes con restricciones de tiempo (ventanas horarias de atención), demanda de servicio, coordenadas de ubicación y datos de la flota como capacidad y número de vehículos.

Existen seis tipos de casos: R1, R2, C1, C2, RC1 y RC2. En los grupos R1 y R2 los clientes están distribuidos uniformemente de manera aleatoria. En los conjuntos C1 y C2, los clientes se colocan en grupos; y finalmente en RC1 y RC2 algunos están distribuidos de manera aleatoria y otros están ubicados por grupos. En las pruebas R1, C1 y RC1 se permiten pocos clientes por ruta (entre 5 a 10 aproximadamente); mientras que en los grupos R2, C2 y RC2 se admiten más de 30 clientes por ruta.

Según lo analizado en el capítulo 2 el criterio de selección de un individuo es la aptitud, la cual se mide por el costo de ruteo, más la penalidad por retraso y un costo fijo por vehículo según lo definido en la ecuación 2.14:

$$Costo\ total = \sum_{k=1}^{m} \left(C_k + Costo(R_k) + pen\_atraso(R_k) \right)$$

Es intuitivo pensar que se debe castigar el incumplimiento de las ventanas de tiempo con una alta penalidad; por otro lado, este hecho debe balancearse con el costo que se fije para C_k. Si el costo por vehículo es muy pequeño entonces la minimización del número de rutas podría quedar en un segundo plano, lo cual en términos operativos no es deseable. Bajo este criterio es preferible fijar un C_k alto para obligar al algoritmo a disminuir primero el número de rutas y luego mejorar; las pruebas siguientes se realizan con una penalidad por atraso igual a 10000 y C_k igual a100000.

Costo de la solución (instancia C101)		
N		10
Media		867.03
Error típico		22.70
Mediana		828.94
Desviación estándar		71.77
Rango		219.11
Mínimo		828.94
Máximo		1048.05
Intervalo de confianza (95.0%)		51.34
	Límite inferior	815.69
	Límite superior	918.37

Número de iteraciones necesarias		
N		10
Media		26.60
Error típico		3.04
Mediana		28.50
Desviación estándar		9.61
Rango		35.00
Mínimo		4.00
Máximo		39.00
Intervalo de confianza(95.0%)		6.87
	Límite inferior	19.73
	Límite superior	33.47

Tabla 4.1: Medidas descriptivas de los resultados obtenidos en 10 pruebas con la instancia C101

Con el objetivo de analizar el comportamiento del algoritmo versus el número de iteraciones, se realizaron diez corridas con la instancia C102 cuyos resultados se exponen en la tabla 4.1. Para todas las pruebas expuestas se usó la probabilidad de mutación igual a 0.6 y una probabilidad de combinación de 1, además se fijó el tamaño de la población en 100 individuos.

Según se expone en la tabla 4.1, se observa que en promedio se requieren aproximadamente 27 iteraciones para que el algoritmo alcance una solución aceptable. Es necesario expresar que estas diez soluciones no violan las restricciones de tiempo y que la mejor solución encontrada tiene el costo de 827.30 reportada por Desrochers [23]. Considerando esta muestra, se puede observar que el promedio del costo de la solución tiene como tope máximo 918.37 con una confianza del 95% y que el número de iteraciones necesarias para alcanzar la solución se encuentra aproximadamente entre 20 y 33.

De aquí que se usa máximo 40 iteraciones como criterio de parada del algoritmo en los resultados expuestos a continuación, a excepción de la prueba con R111 debido que se comprobó que instancias de este tipo requieren más esfuerzo computacional para alcanzar una solución aceptable, tema que se trata a continuación.

4.3 Resultados computacionales con las pruebas de Solomon

Para las siguientes pruebas se utilizó un ordenador de tipo notebook con las siguientes características: procesador Intel(R) Core (TM) Duo CPU 1.83 GHz, memoria (RAM) 1015 MB y sistema operativo de 32 bits. Debido que el tiempo de ejecución de un algoritmo puede variar de un computador a otro, en los resultados aquí analizados, el esfuerzo computacional no se lo relaciona con el tiempo, sino más bien con el número de iteraciones necesarias para encontrar la mejor solución en cada prueba.

Aunque la Metaheurística propuesta no se la ha planteado como multiobjetivo, la figura 4.1 ilustra que tanto el número de vehículos y el costo del ruteo tienden a disminuir según avanzan las iteraciones para la prueba con la instancia C101; a la cuarta iteración se alcanza un número mínimo de rutas y a la octava se reporta la mejor solución alcanzada con un costo de 828.94, muy cerca del óptimo de Solomon de 827.30 [28].

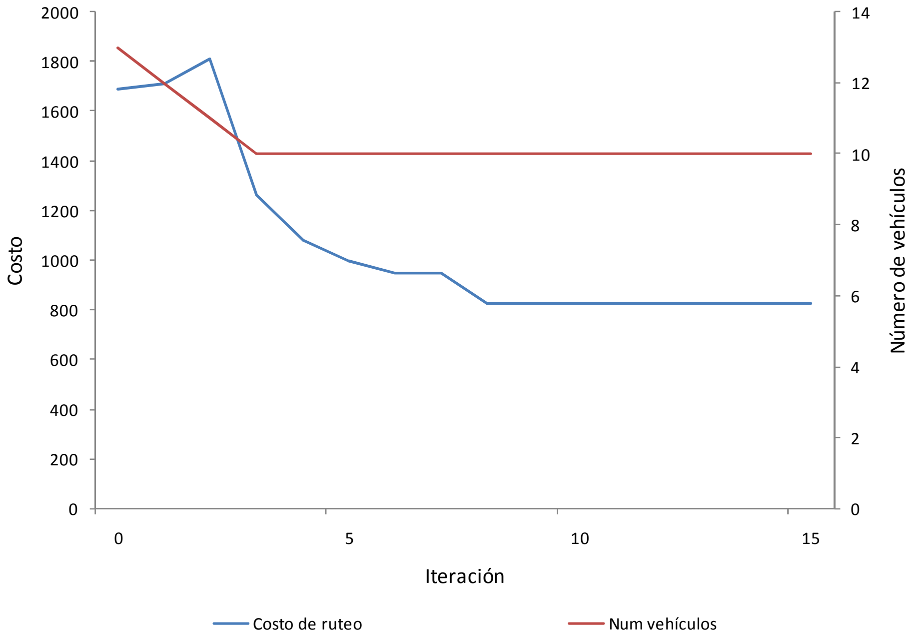


Figura 4.1: Costo de ruteo vs número de vehículos de la solución para la prueba de la instancia C101

En la figura 4.2 se observa que a medida que el número de iteraciones aumenta, el total de atrasos disminuye; sin embargo, el total de espera aumenta al inicio, pero ambas infactibilidades de tiempo llegan a cero en la iteración 8, es decir que en la mejor solución no existen violaciones de las ventanas de tiempo.

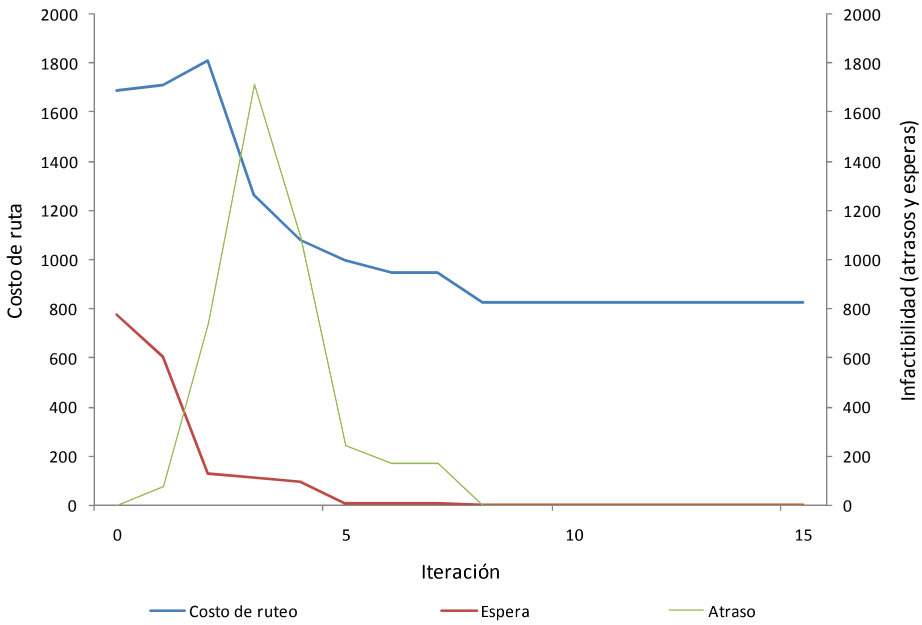


Figura 4.2: Costo de ruteo vs infactibilidad de la solución para la prueba de la instancia C101

Con la instancia R104 la solución converge a un costo de 1174.84, pero se produjo una espera acumulada de 10.88 unidades de tiempo con cero atrasos. Cabe expresar que según el problema definido en el capítulo 1, se consideran ventanas de tiempo suaves y por lo tanto se puede aceptar cierta cantidad de espera, siempre y cuando el servicio al cliente no se vea afectado, es decir que exista retraso en el picking.

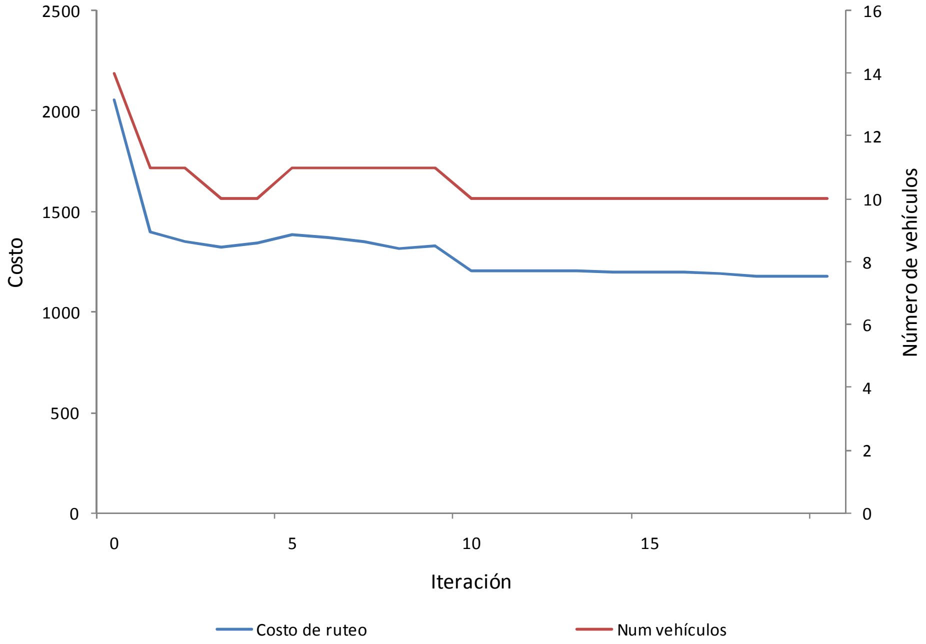


Figura 4.3: Costo de ruteo vs número de vehículos de la solución para la prueba de la instancia R104

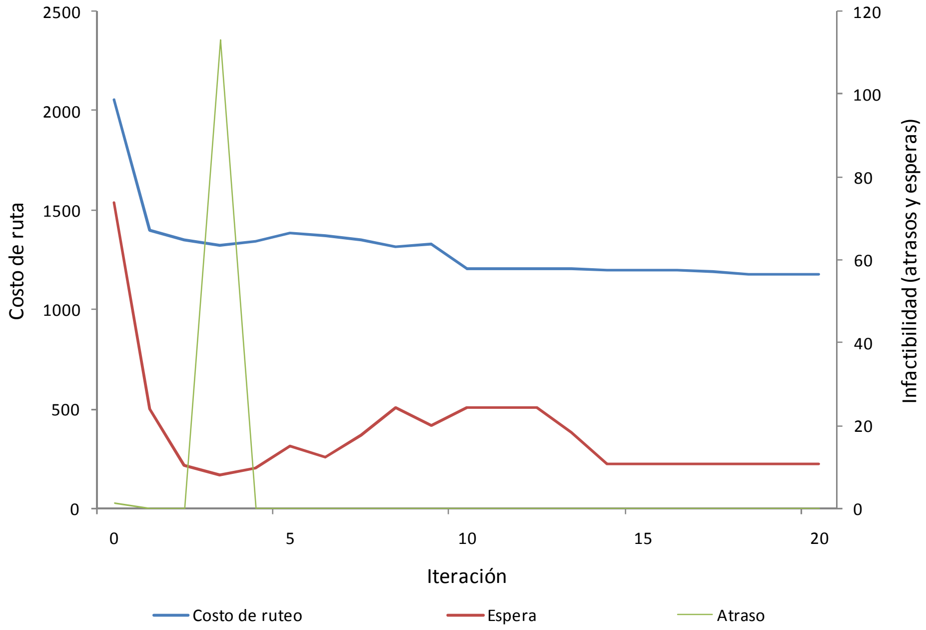


Figura 4.4: Costo de ruteo vs infactibilidad de la solución para la prueba de la instancia R104

Con el objetivo de analizar la salida del sistema usando más iteraciones para las instancias tipo R; se realizó una prueba con los datos del grupo R111 con 160 iteraciones, cuyos resultados se ilustran en la figuras 4.5 y 4.6. Se observa que recién en la iteración 146 se logra cumplir con las restricciones de tiempo (cero atrasos y esperas) y en la 147 se alcanza la mejor solución con un costo de 1316 y 11 rutas. Este resultado, junto con los de la prueba R104 sugieren que se requiere mayor esfuerzo computacional para abordar los problemas con clientes distribuidos aleatoriamente (instancias tipo R) que con los clientes agrupados por zonas geográficas (instancias tipo C); sin embargo, la tendencia de disminución de costos e infactibilidades de tiempo se mantiene a medida que aumenta el número de iteraciones.

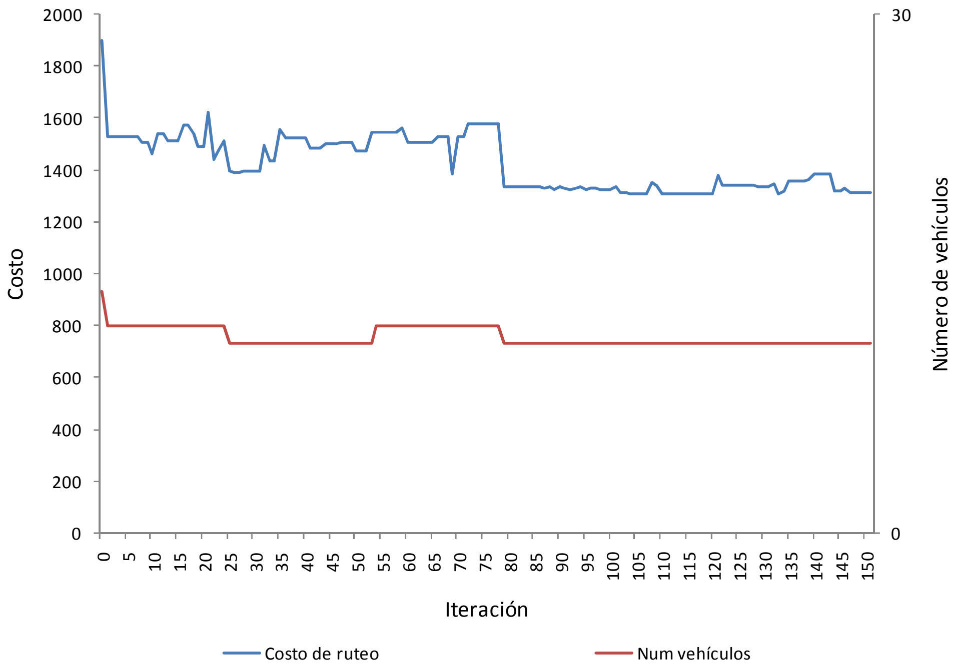


Figura 4.5: Costo de ruteo vs número de vehículos de la solución para la prueba de la instancia R111

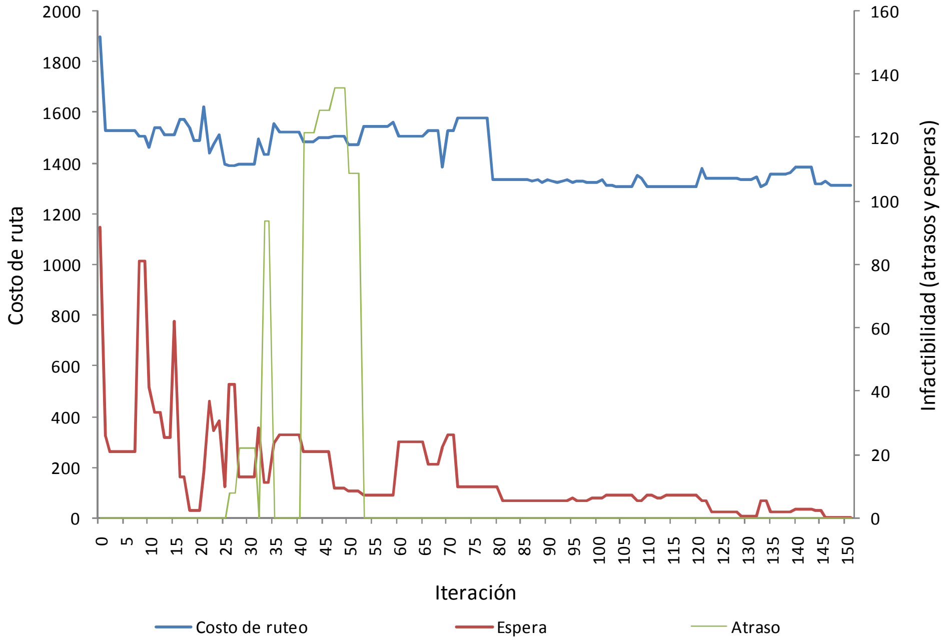


Figura 4.6: Costo de ruteo vs infactibilidad de la solución para la prueba de la instancia R111

En las figuras 4.7 y 4.8 se resume el resultado de la prueba con la instancia RC103; en la iteración 23 se alcanza la mejor solución con un costo de 1424.34, 11 rutas y sin incumplir las restricciones de tiempo.

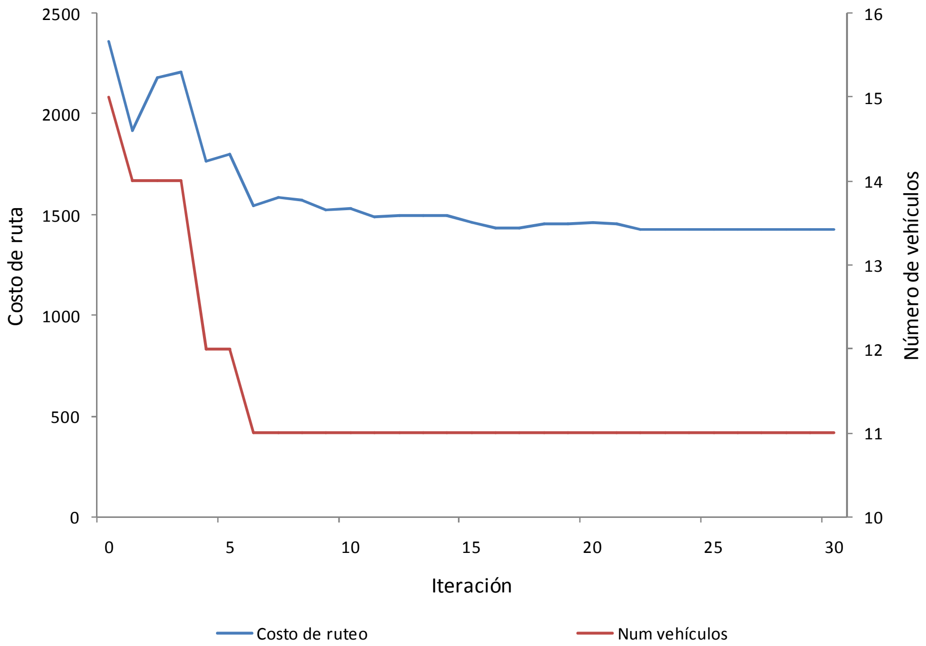


Figura 4.7: Costo de ruteo vs número de vehículos de la solución para la prueba de la instancia RC103

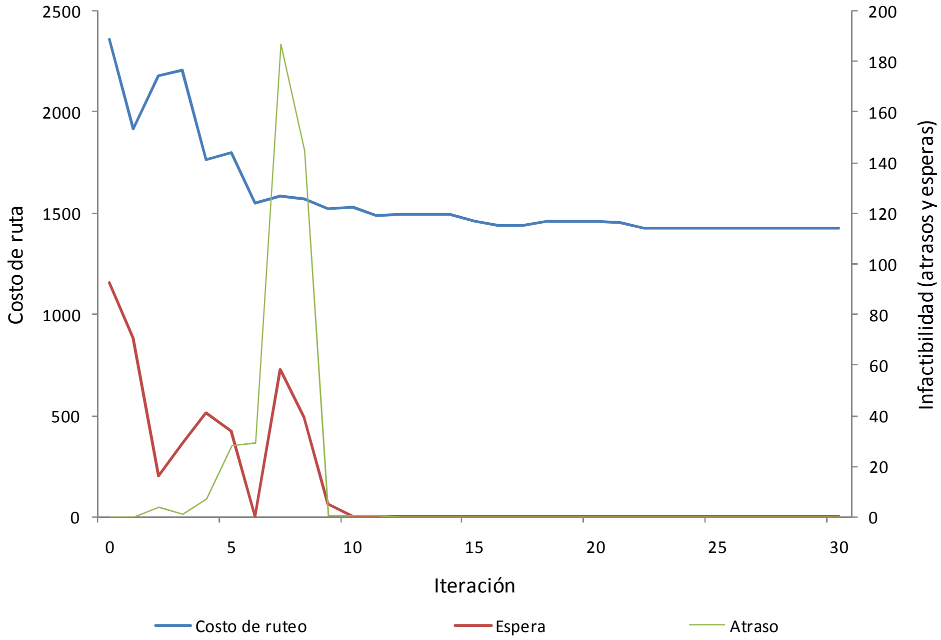


Figura 4.8: Costo de ruteo vs infactibilidad de la solución para la prueba de la instancia RC103

En la tabla 4.2 se resumen los resultados anteriores y se los compara con el óptimo de Solomon; se observa que el AG propuesto produce excelentes resultados en las instancias tipo C, es decir, donde los clientes están agrupados por zonas geográficas; y se puede concluir que en las instancias de tipo R y RC se alcanzan resultados aceptables. Para todas las pruebas realizadas, el algoritmo siempre consigue un mínimo número de rutas en la solución. En la tabla 4.3 se exponen las rutas que componen cada una de las soluciones según la prueba realizada.

	Optimo		Solución AG propuesto	
Instancia	Num rutas	Costo	Num rutas	Costo
C101	10	827.3	10	828.94
C102	10	827.3	10	828.94
R104	10	982.01[7]	10	1174.84
R111	12	1048.7	11	1316
RC103	11	1258	11	1424.34

Tabla 4.2: Comparación de la solución obtenida y del óptimo de Solomon para las pruebas realizadas[8]

[7] *Solución reportada por Rochat y Taillard (1995)*

[8] *El óptimo es tomado de la publicación "Best known solutions Solomon's VRPTW 100 customers problems instances". VRP web. Universidad de Málaga.* *http://neo.lcc.uma.es/radi-aeb/WebVRP*

Instancia	Vehículos	Rutas
C101	10	{43-42-41-40-44-46-45-48-51-50-52-49-47} {5-3-7-8-10-11-9-6-4-2-1-75} {20-24-25-27-29-30-28-26-23-22-21} {67-65-63-62-74-72-61-64-68-66-69} {90-87-86-83-82-84-85-88-89-91} {98-96-95-94-92-93-97-100 -99} {32-33-31-35-37-38-39-36-34} {81-78-76-71-70-73-77-79-80} {13-17-18-19-15-16-14-12} {57-55-54-53-56-58-60-59}
C102	10	{43-33-31-35-37-38-39-36-34} {5-42-41-92-93-97-100-99-95} {90-87-86-83-82-84-85-88-89-91-94-98} {20-24-25-27-10-11-9-6-4-2-1-75} {67-65-63-62-74-72-61-64-68-66-69} {32-3-8-29-30-28-26-23-22-21} {81-78-76-71-70-73-77-79-80} {13-17-18-19-15-16-14-12} {57-55-54-53-56-58-60-59} {7-96-40-44-46-45-48-51-50-52-49-47}
R104	10	{61-16-93-59-99-84-83-60-96-95-94-6-89} {73-41-15-57-87-2-22-74-72-21-40-58} {52-18-7-48-82-8-45-17-86-91-100} {54-24-80-76-79-33-3-68-12-77-50} {92-85-14-44-38-43-42-37-98-5} {31-88-10-62-19-49-36-47-46} {29-34-78-81-9-35-65-71-51} {64-11-63-90-32-66-20-30-70} {26-27-69-1-53-28-97-13} {56-75-23-67-39-55-4-25}

Tabla 4.3: Rutas de las soluciones en las pruebas realizadas

Instancia	Vehículos	Rutas
R111	11	{95-92-42-15-57-87-97-37-98-91-100-59-96} {28-27-69-50-76-79-78-3-68-80-12-77} {52-82-18-7-88-31-10-90-32-70-1} {83-45-61-85-99-84-17-93-5-60-89} {72-73-23-67-22-41-75-74-21-58} {33-30-51-81-9-35-71-66-20} {2-39-53-40-56-4-54-26} {47-36-46-8-6-94-13} {48-19-62-11-64-49-63} {16-14-44-38-86-43} {65-34-29-24-55-25}
RC103	11	{45-1-3-5-46-8-6-7-4-2-100} {91-92-95-62-50-67-84-56-66-90-80} {65-52-11-15-16-9-10-13-47-14-12} {42-39-36-37-38-41-54-61-68-70} {94-71-72-44-40-43-35-93-96-81} {24-18-48-21-23-22-49-19-25-77} {33-27-30-32-28-26-29-31-34} {82-98-69-88-78-79-73-17} {57-83-64-99-53-60-55} {75-97-59-87-86-74-58} {85-89-76-63-51-20}

Tabla 4.3 (continuación): Rutas de las soluciones en las pruebas realizadas

Capítulo 5

5. Caso de estudio:

Servicio de recolección de correo en la ciudad de Guayaquil

5.1 Introducción

El servicio postal, correo o correspondencia (courier), consiste en el transporte de documentos y paquetería. Este método de comunicación, a través de un intermediario, existe aproximadamente desde la invención de la escritura[9].

En Ecuador existen dos gremios que agrupan a las empresas courier: Asociación Ecuatoriana de Correos Privados (ACOPRI) y Asociación Ecuatoriana de Mensajería (ASEME), con 64 afiliados; según Lupe Ortega Peña, presidenta de ACOPRI, el directorio de la Corporación Aduanera Ecuatoriana (CAE) de 2004, eliminó el requisito que obligaba a los couriers a afiliarse a una de las dos asociaciones, previa la obtención de la licencia para desaduanización. Se estima que en el país existen alrededor de 800 empresas de correo[10].

Ninguno de los miembros de las asociaciones courier, que están autorizados legalmente y procuran cumplir normas homogéneas de servicio y operación, cuenta en Ecuador, con un sistema para planificación de rutas; debido que, en parte, no visualizan el

[9] *Wikipedia – Correo postal - http://es.wikipedia.org/wiki/Correo_postal - última modificación 14-dic-09 19:39.*

[10] *Diario Hoy - http://www.hoy.com.ec/noticias-ecuador/se-perfecciona-la-evasion-en-las-aduanas-235585-235585.html - 29-may-06.*

beneficio y por otro lado la escasa oferta de estos sistemas, salvo paquetes extranjeros con elevados costos.

En la actualidad, las organizaciones prefieren tercerizar los servicios de mensajería y distribución cuando no son en sí mismos parte esencial de su negocio pues es más barato administrar un contrato que ejecutar dicha operación por su cuenta y riesgo y por otro lado pueden dedicar más tiempo a su verdadero negocio (financiero, marca, mercadeo o fórmula de producción, por ejemplo). Esto implica que operadores logísticos y couriers como la empresa antes mencionada, tienen una gran cantidad de clientes potenciales.

Por lo anterior, el presente estudio incluye la aplicación de la Metaheurística propuesta en un problema real de una empresa courier ecuatoriana[11].

5.2 Información general de la empresa

La empresa, con matriz en la ciudad de Guayaquil, es una de las más importantes de la industria, debido a su gran cantidad de: clientes corporativos (1,000 aprox.), colaboradores (alrededor de 1,000), flota de vehículos (100 aprox.), 120 puntos de venta y operaciones locales e internacionales.

En su UEN (unidad estratégica de negocios), entre otros, ofrecen los siguientes servicios:

1. Entrega en domicilio (mercancías y documentos).
2. Recolección en domicilio (mercancías y documentos).
3. Mercadeo.
4. Mensajería interna.
5. Servicio courier internacional.

[11] *Todos los datos inherentes a "la empresa" fueron proporcionados por el jefe de operaciones y autorizados por su gerente general.*

Es importante recalcar que la empresa estuvo negociando la adquisición de una licencia de Agemap por el costo aproximado de $30,000 para aplicar a otro de sus productos: entrega de documentos y publicidad masiva, puerta a puerta, con mensajeros a pie y en bicicleta. Lo cual nos indica que la empresa esta volteando la mirada hacia la aplicación de softwares de ruteo para reducir sus costos y mejorar sus niveles de servicio.
Se redacta brevemente la operación de recolección y varios aspectos que la afectan:

1. Debido a la gran oferta de operadores logísticos y empresas courier puntualmente, el área de ventas, con el objetivo de captar clientes, está prácticamente forzada no solo a ofrecer precios competitivos si no a ceder en varios aspectos que afectan el nivel de servicio: el horario de recolección, la frecuencia de recolección durante la semana o si el servicio se realizará previa llamada, son decisiones que usualmente toma el cliente sin consultar con el área de operaciones.

2. Cada cliente requiere servicios personalizados según sus propias operaciones, sin embargo no se realizan reuniones de trabajo previas entre las áreas de operaciones proveedor - cliente, lo que implica que no conocen la necesidad real del cliente y tratar de satisfacer su necesitad termina siendo más complicado y costoso.

3. Los clientes previa llamada, tienen la opción de hasta determinada hora solicitar el servicio para el mismo día, es decir que la ruta puede cambiar antes de la programación o durante la ejecución.

4. No existe una correcta clasificación ABC de clientes, no hay seguimiento de la facturación del cliente, lo que implica que incluso clientes tipo D y C, son atendidos como un cliente tipo A, pues el área de operaciones es presionada para dar el servicio según el cliente lo requiera.

5. La flota de camiones es heterogénea y alquilada (con conductor) en su mayor parte y aunque los mensajeros son propios de la empresa, no tiene ningún conocimiento de los costos reales de una ruta de recolección.

6. Aunque existe un departamento de auditoría operativa, no hay procedimientos claros sobre el alquiler y asignación de los camiones, el supervisor de turno decide, según su criterio, si debe alquilar más camiones, cuál camión debe atender determinada ruta o cuál no debe trabajar.

7. No llevan indicadores logísticos, costeo ABC ni conocen sus costos logísticos, por lo tanto no saben si mejoran o empeoran.

8. La atención de los clientes es monitoreada por un operador de radio manualmente, es decir el mensajero reporta cada vez que visita al cliente y el operador registra en una hoja el cumplimiento del servicio en esa ruta.

9. No existe un sistema de administración de flota adecuado, donde se registre y consulte la operación diaria.

10. Una vez que el cliente ha sido programado para su recolección ya sea diaria, semanal o previa llamada, dependiendo de su ubicación en la ciudad, de la hora y de la demanda del cliente (cantidad de paquetes y peso), un supervisor de operaciones decide según su criterio y conocimiento, en cuál ruta encaja mejor el cliente o si necesita un camión y mensajero adicionales para atender dicha orden.

11. Cada mensajero recibe su hoja de ruta donde indica el orden de los clientes que debe visitar, sin embargo debe estar atento a cualquier cambio en su ruta por cancelaciones o inclusión de un cliente a última hora.

12. Si el cliente aún no tiene lista la mercadería que enviará, el mensajero toma la decisión de esperar o solicita autorización para retirarse y regresar al sitio por segunda ocasión más tarde. En la práctica también puede suceder que el mensajero se queda esperando mientras el camión visita otro cliente.

13. El camión atiende los puntos de servicio asignados en su ruta. Estos puntos, figuran en ruta como una visita (parada) más del camión, sin embargo, requiere

más tiempo de lo planificado, pues debe recolectar todas las ventas del día (paquetes y documentos) y verificar lo que recibe con documentos; ocasionalmente debe esperar, debido que aún están atendiendo algún cliente, sin embargo no puede esperar mucho y alguien debe autorizar si regresa en otro momento o envían otro camión.

14. El camión sale del CD y al terminar la ruta regresa al mismo.

5.3 Descripción del problema específico

Uno de los principales productos de la empresa, es el servicio de recolección y entrega a domicilio para sus clientes corporativos. El problema puntual en el cual nos enfocaremos, es la recolección de paquetería (carga liviana) de sus clientes, ubicados geográficamente en el centro de la ciudad de Guayaquil. Estos clientes normalmente requieren que sus paquetes sean recogidos después de cerrar sus ventas y operaciones del día, es decir entre 16 y 18 horas del día.

El CD está ubicado al norte de la ciudad, posee una flota de camiones heterogénea y tercerizada, que varía desde 2.5 hasta 13 toneladas, cada camión va con su conductor y un mensajero, salvo que por previas indicaciones del cliente se requiera la presencia de más personas para el proceso de estiba.

En el CD, los mensajeros, reciben una hoja de recolección con la secuencia de visita a los clientes, según la ventana de atención respectiva. Las rutas, los clientes y los camiones son asignados según el conocimiento empírico del supervisor de operaciones.

Este servicio de recolección contiene en sí mismo varios problemas, sin embargo, para el estudio, se relajó y se modeló un problema menos riguroso, por lo cual se anotan las siguientes observaciones:

1. Se consideró una flota homogénea de vehículos.
2. Los vehículos salen del CD y regresan al mismo como último punto de arribo, sin embargo en la práctica, debido a retrasos operativos, no siempre salen del CD y después de llegar al CD pueden volver a salir para una visita puntual.

3. Cada vehículo visita un cliente a la vez, aunque en la práctica, debido a la cercanía entre uno y otro cliente (distancia corta o están ubicados en el mismo edificio, en diferentes oficinas o pisos), el conductor del camión visita un cliente mientras el mensajero visita otro.
4. Todos los clientes son programados y visitados para recolección de carga, en la práctica pueden no enviar nada o solo enviar sobres con documentos, lo cual implica que la capacidad del camión estaría subutilizada.
5. El problema de ruteo es solo de recolección, pero en realidad al mismo tiempo se puede realizar una entrega (delivery) que quedó pendiente o que fue solicitada a último momento.
6. El modelo implica ventanas de tiempo suaves y se costean los atrasos, más la empresa no tiene indicadores de gestión para medir el nivel de servicio.
7. El tiempo de servicio y las demandas de los clientes, se consideraron determinísticos, pero en realidad las demandas de los clientes son estocásticas y por lo tanto también lo es el tiempo de servicio.
8. Se consideró una velocidad fija para la zona del centro de Guayaquil, sin embargo, la velocidad es una variable continua y por lo tanto los tiempos en ruta también lo son.

5.4 Datos reales para pruebas con la Metaheurística propuesta

5.4.1 Construcción de indicadores

En abril de 2009 comenzaron las reuniones de trabajo con la empresa y se recibieron los datos registrados en su sistema, sin embargo, debido a que la empresa no contaba con indicadores de gestión, el primer paso fue organizar, revisar y validar los datos y construir dichos indicadores para los meses desde enero hasta abril de 2009, periodo que sirvió de base para la obtención de la información.

2009		ENERO											
		TOTAL PICKING				CUSTOMERS VISITED							
SECTOR	COD SECTOR	TOTAL DOX MES	TOTAL VAL MES	TOTAL MERC MES	TOTAL MES	(1) ENVIOS DIA / CLIENTE	(2) MERC DIA / CLIENTE	(3) STOPS MES	(4) CLIENTES AVG / DIA	(5) NS=1	(6) NSA=1	(7) ICE	(8) ICEM
BANCARIO	**C1**	1277	396	1202	2875	6	5	462	21	0.51	0.63	0.76	0.56
BAHIA	**C2**	1792	99	1317	3208	8	6	396	18	0.24	0.55	0.60	0.56
1 LADO 9 OCT	**C3**	1871	106	1577	3554	9	10	385	18	0.72	0.78	0.65	0.41
2 LADO 9 OCT	**C4**	141	69	725	935	5	12	183	8	0.46	0.68	0.67	0.34
DURAN	N1	885	51	2674	361	12	20	311	14	0.32	0.56	0.79	0.42
KENEDY	N11	1334	35	1443	2812	7	6	421	19	0.35	0.49	0.79	0.60
J.T.M.	N12	561	83	381	4454	11	19	405	18	0.33	0.55	0.64	0.49
VIA DAULE	N2	1979	73	940	2992	5	5	629	29	0.55	0.71	0.79	0.32
VIA DAULE	N3	1225	2	3257	4484	9	15	497	23	0.32	0.58	0.67	0.43
ALBOR, OTROS	N4	1999	167	1427	3593	8	6	437	20	0.53	0.66	0.70	0.53
URDESA	N5	955	17	1603	2575	7	8	346	16	0.23	0.64	0.69	0.58
KENEDY	N6	1099	154	1233	2486	5	4	536	24	0.49	0.68	0.77	0.56
C.J.A.	N7	1425	186	1416	3027	6	5	538	24	0.38	0.64	0.74	0.49
AV AMERICAS	N9	1907	205	2835	4947	9	9	522	24	0.42	0.72	0.71	0.62
SUR	S1	1868	116	2448	4432	14	13	325	15	0.46	0.71	0.79	0.60
	TOTAL	20318	1759	27907	49984	8	9	6.393	291	0.43	0.64	0.72	0.50

(1): # DE ENVÍOS PROMEDIO DE C/ CLIENTE, C/ DÍA (ENVIOS = DOX, VALIJAS Y/ O MERCANCÍAS)
(2): # DE PAQUETES QUE ENVÍA C/ CLIENTE DE MERCANCÍAS, C/ DÍA
(3): TOTAL DE PARADAS QUE HIZO EL CAMIÓN EN EL MES (TOTAL CLIENTES VISITADOS MES)
(4): PROMEDIO DE CLIENTES VISITADOS C /DÍA
(5): NIVEL DE SERVICIO, CLIENTES VISITADOS EN HORARIO ACORDADO
(6): NIVEL DE SERVICIO AJUSTADO, CLIENTES VISITADOS HASTA 25 MIN DESPUÉS DE HORA ACORDADA
(7): INDICE DE CLIENTES QUE EFECTIVAMENTE HACEN ENVÍOS
(8): INDICE DE CLIENTES QUE EFECTIVAMENTE ENVÍAN MERCANCÍAS

Tabla 5.1: Indicadores elaborados para el servicio de recolección de mercancías (carga liviana) para clientes corporativos de la ciudad de Guayaquil, enero 2009

Además de las quince zonas de recolección detalladas en la tabla 5.1, tienen al menos dos vehículos diarios que hacen servicios de "volante", es decir, visitan aquellos clientes que no pudieron ser atendidos normalmente por el exceso de carga, la ruta puede variar cada día, es decir, pueden ir desde el norte, al centro, al sur, al este o al oeste; sin embargo, no fueron tomados en cuenta para la medición por la variabilidad de los datos.

5.4.2 Selección de los clientes

Los clientes seleccionados fueron de las zonas C1, C2, C3, C4, según la tabla 5.1, y otros más, ubicados principalmente en el centro de la ciudad. Los criterios de selección fueron:

1. Las zonas del centro de la ciudad tenían datos consistentes en los cuatro meses evaluados.
2. Todos los clientes están ubicados dentro de una misma zona de la ciudad, lo cual reduce la complejidad para establecer rutas de tránsito, velocidad y cálculo de distancias.
3. Las rutas resultantes de la aplicación de la Metaheurística pueden ser comparables con las rutas que la empresa tiene actualmente.
4. En otras zonas de la ciudad, como por ejemplo Kennedy norte, existe el problema de que muchos clientes están ubicados en un mismo edificio como las dos torres del WTC en la Av. Francisco de Orellana, hay varios clientes en cada piso de las torres; en estos casos el camión se parquea en un único lugar, mientras el mensajero y el conductor pueden visitar uno o varios clientes al mismo tiempo; aunque en el centro también puede darse el caso, es menos frecuente.

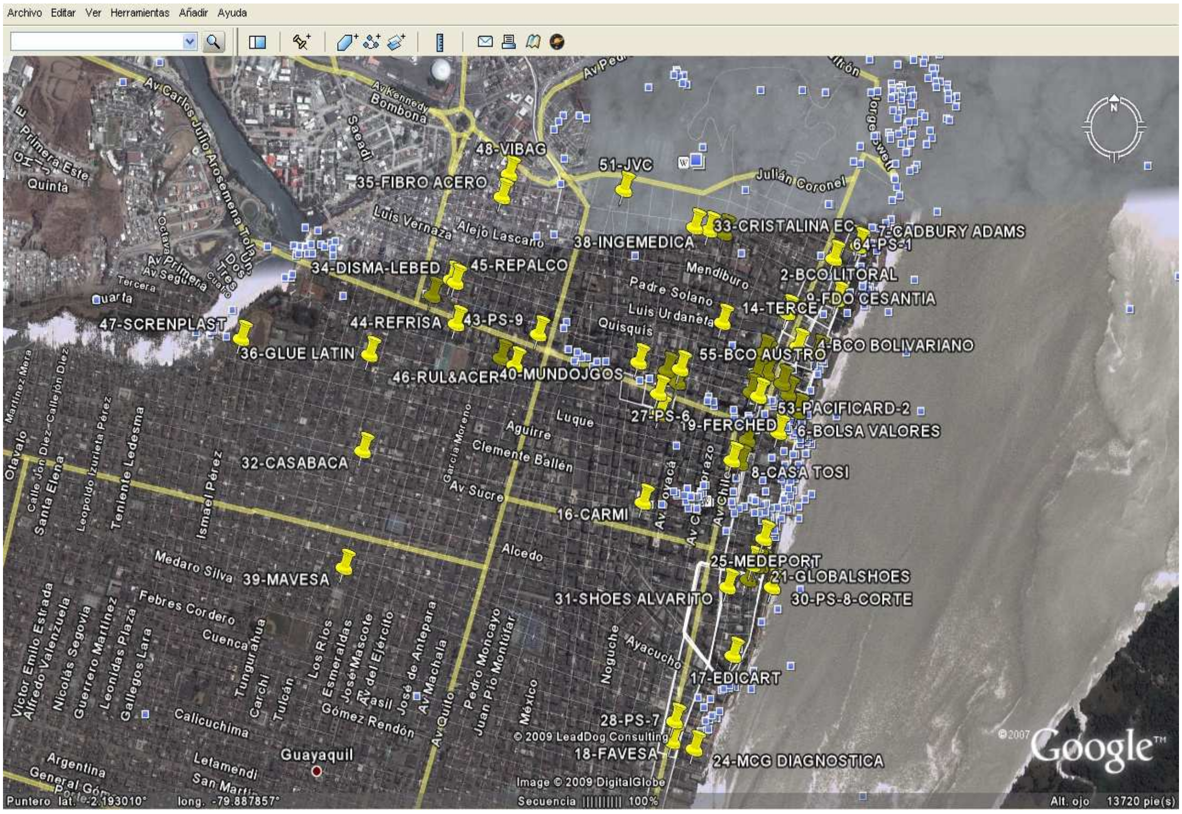


Figura 5.1: Ubicación espacial de los clientes de la empresa. Mapa del centro de la ciudad de Guayaquil de Google Earth

5.4.3 Ubicación geográfica de los clientes

Una vez seleccionados los clientes, es necesario establecer su ubicación geográfica y espacial. Para el efecto, con las direcciones de cada uno de los clientes y con la herramienta Google Earth se pudieron conocer las coordenadas geográficas. El sistema de coordenadas geográficas[12] es un sistema de referencia que usa las coordenadas angulares, latitud y longitud, para expresar la posición de cualquier punto de la superficie terrestre. La latitud mide el ángulo entre cualquier punto y el ecuador[13] (plano perpendicular al eje de rotación que divide al tierra en los hemisferios norte y sur). La longitud mide el ángulo a lo largo del ecuador desde cualquier punto del planeta.

En total son 65 clientes y el depósito, para aquellos clientes ubicados en el mismo edificio o uno junto a otro, la ubicación espacial es la misma, sin embargo, tienen diferentes ventanas horarias y/o demanda, tal es el caso de los nodos clientes 20 y 21

[12] *Wikipedia – Coordenadas geográficas - http://es.wikipedia.org/wiki/Coordenadas_geogr%C3%A1ficas – modificación 29-dic-09 2:12.*

[13] *Wikipedia – Ecuador terrestre - http://es.wikipedia.org/wiki/Ecuador_terrestre - última modificación 20-dic-09 19:26.*

(en tabla 5.2) cuya ubicación es la misma, ambos con demanda y ventanas horarias diferentes.

CLIENTE	DIRECCIÓN	LAT	LONG
0	JTM Y GUILLERMO CUBILLO	-2,138273	-79,914612
1	CORDOVA 1005 Y P. ICAZA	-2,191019	-79,881373
2	MALECON 514 E IMBABURA	-2,188471	-79,878389
3	P. ICAZA 125 Y PICHINCHA P15	-2,191295	-79,880403
4	JUNIN 200 Y PANAMA	-2,190063	-79,879828
5	P. ICAZA MATRIZ BCO PACIFICO P5 PROVEDURIA	-2,191647	-79,880183
6	PICHINCHA 334 Y ELIZALDE ED. EL COMERCIO P1	-2,192947	-79,880548
7	EL FORTIN P. AGUIRRE 104 Y MALECON P4	-2,186674	-79,877584
8	AGUIRRE Y P. CARBO P5	-2,193972	-79,881966
9	VICENTE ROCAFUERTE 748 Y LUIS URDANETA	-2,188930	-79,880224
10	P. CARBO 613 Y LUQUE P3	-2,193201	-79,881699
11	PEDRO CARBO 715 Y AGUIRRE	-2,193972	-79,881966
12	JUNIN 114 Y MALECON TORRES DEL RIO P4 OFC6	-2,190291	-79,879103
13	V.M.R. Y CORDOVA MEZZANINE	-2,190275	-79,881104
14	JUNIN 511 Y ESCOBEDO ESQ. P1	-2,189249	-79,882615
15	GENERAL CORDOBA 1021 Y 9 OC SAN FCO 300	-2,191588	-79,881554
16	SUCRE E/ G. AVILES Y BOYACA	-2,195303	-79,885549
17	MALECON 2309 Y AV. OLMEDO	-2,200456	-79,882223
18	CUENCA 102 Y ELOY ALFARO P6 OFC11	-2,203358	-79,884466
19	LUQUE 632 E/ GARCIA AVILES Y BOYACA	-2,192296	-79,884924
20	COLON 206 Y PICHINCHA EDF COLON OFC204	-2,197430	-79,881351
21	EDF COLON 206 Y PICHINCHA P3 OFC 301	-2,197430	-79,881351
22	COLON 206 Y PICHINCHA EDF COLON OFC204	-2,197430	-79,881351
23	ELOY ALFARO 326 Y LETAMENDI	-2,207754	-79,885385
24	CUENCA 102 Y ELOY ALFARO P6 OFC9	-2,203358	-79,884466
25	SUCRE 113 E/ MALECON Y PICHINCHA	-2,196523	-79,881088
26	PICHINCHA 816 Y COLON PRIMER PISO	-2,197895	-79,881650
27	VELEZ 601 Y GARCIA AVILES	-2,191639	-79,884974
28	FEBRES CORDERO 223 E/ CHILE Y ELOY ALFARO	-2,202674	-79,884385
29	C.C. MALECON 2000 L-8 BL-C	-2,198155	-79,880774
30	C.C. MALECON 2000 L-8 BL-C	-2,198155	-79,880774
31	ELOY ALFARO 205 Y ABDON CALDERON	-2,198146	-79,882421
32	LOS RIOS 1508 E/ SUCRE Y COLON	-2,193570	-79,895906
33	MANUEL GALECIO 233 E/ JIMENA Y BOYACA	-2,186041	-79,883639
34	J. MASCOTE 614 Y 1ERO. DE MAYO	-2,187896	-79,892482
35	AV. DEL EJERCITO 118 Y M. GALECIO	-2,185020	-79,890762
36	TULCAN 1017 Y LUQUE	-2,190329	-79,895656
37	VELEZ 1305 Y JOSE DE ANTEPARA	-2,190442	-79,890801
38	MANUEL GALECIO 231 E/ JIMENA Y BOYACA	-2,186141	-79,883109
39	HUCAVILCA Y LOS RIOS	-2,197495	-79,896597
40	9 DE OCT. 838 E/ RUMICHACA Y G. AVILES	-2,190582	-79,885709
41	MANUEL GALECIO Y BOYACA	-2,186239	-79,882558
42	ESMERALDAS 909 Y 9 DE OCT.	-2,188342	-79,893335
43	AV. QUITO 902 Y 9 DE OCTUBRE	-2,189643	-79,889418
44	HURTADO 513 Y JOSE MASCOTE	-2,189294	-79,892455
45	J. MASCOTE 713 Y 1 ERO. DE MAYO	-2,187896	-79,892482
46	MACHALA 1101 Y VELEZ	-2,190686	-79,890290
47	AGUIRRE 2703 E ISMAEL P. PAZMIĐO	-2,189795	-79,900399
48	AV DEL EJERCITO #120 E/ MANUEL GALECIO Y PIEDRAHITA	-2,184252	-79,890495
49	PRESIDT. LUIS C Y SUFRAGIO LIBRE FRTE. AGUIRRE ABAD	-2,178299	-79,890922
50	9 DE OCT 707 E/ G. AVILES Y BOYACA	-2,190865	-79,884671
51	PIEDRAHITA Y LORENZO DE GARAYCOA ESQ.	-2,184782	-79,886258
52	P. ICAZA Y PICHINCHA BCO. DEL PACIFICO P6	-2,191647	-79,880183
53	SAN FRANCISCO 300 PISO 9 DPTO. DE SISTEMAS	-2,191733	-79,881277
54	VICTOR MANUEL RENDON 415 Y CORDOVA P7	-2,190275	-79,881104
55	BOYACA Y 9 DE OCTUBRE	-2,190812	-79,884170
56	P.CARBO Y 9 DE OCT EDF SN FRCO 300	-2,191733	-79,881277
57	AV. AMERICAS (FTE. MECANOS)	-2,167083	-79,891643
58	EDF. LA PREVISORA PISO 27 OFC. 5	-2,192246	-79,879829
59	EDF. LA PREVISORA PISO 27 OFC. 2702	-2,192246	-79,879829
60	EDF. LA PREVISORA P30 OFC. 3006	-2,192246	-79,879829
61	P. ICAZA Y CORDOVA PISO 3	-2,191019	-79,881373
62	EDF. LA PREVISORA PISO 27 OFC. 6	-2,192246	-79,879829
63	9 DE OCT Y BOYACA P-2	-2,191271	-79,884276
64	PANAMA 306 THOMAS MARTINEZ	-2,187101	-79,878499
65	EDF. LA PREVISORA 1ER PISO	-2,192246	-79,879829
66	AV. J.T.M. KM. 2.5 FTE. A SERVIENTREGA	-2,139938	-79,915874

Tabla 5.2: Coordenadas geográficas (en grados decimales) de los clientes y el depósito

5.4.4 Cálculo de distancias y tiempos

Las distancias entre todos los nodos, desde el 0 hasta el 66, implicó la construcción de 4,489 (67 × 67) rutas, para lo cual:

1. Se levantó la tabla de sentido de las vías del centro de la ciudad de Guayaquil.

SENTIDOS DE LAS VÍAS ESTE - OESTE Y ESTE - OESTE
CENTRO DE GUAYAQUIL

CALLE	VÍA	ORIGEN	DESTINO	DESDE	HASTA
LOJA	UNA	ESTE	OESTE		
JULIAN CORONEL	UNA	ESTE	OESTE		
JUAN MONTALVO	UNA	OESTE	ESTE		
PADRE AGUIRRE	UNA	OESTE	ESTE		
PIEDRAHITA	UNA	OESTE	ESTE	RIOBAMBA	BOYACÁ
PIEDRAHITA	UNA	ESTE	OESTE	RIOBAMBA	ESTERO
T MARTÍNEZ	UNA	OESTE	ESTE	CÓRDOVA	MALECÓN
T MARTÍNEZ	DOBLE	OESTE	ESTE	ESCOBEDO	CÓRDOVA
IMBABURA	DOBLE	OESTE	ESTE		
LUZURRAGA	DOBLE	OESTE	ESTE	ROCAFUERTE	PANAMÁ
LUZURRAGA	UNA	OESTE	ESTE	PANAMÁ	MALECÓN
ROCA - URDANETA	UNA	ESTE	OESTE		
MANUEL GALECIO	UNA	OESTE	ESTE		
ALEJO LASCANO	UNA	ESTE	OESTE		
MENDIBURO	UNA	ESTE	OESTE		
PADRE SOLANO	UNA	OESTE	ESTE		
JUNÍN	UNA	ESTE	OESTE		
VM RENDÓN, 1 MAY	UNA	OESTE	ESTE		
P ICAZA	UNA	ESTE	OESTE	G AVILÉS	CÓRDOVA
P ICAZA	UNA	OESTE	ESTE	P CARBO	MALECÓN
09-oct	DOBLE	OESTE	ESTE	STA ELENA	BOYACÁ
09-oct	UNA	OESTE	ESTE	BOYACÁ	MALECÓN
VELEZ	UNA	ESTE	OESTE		
LUQUE	UNA	OESTE	ESTE		
ELIZALDE	UNA	OESTE	ESTE		
ILLINGWORTH	UNA	ESTE	OESTE		
AGUIRRE	UNA	ESTE	OESTE	PICHINCHA	ESTERO
AGUIRRE	DOBLE	ESTE	OESTE	PICHINCHA	MALECÓN
BALLÉN	UNA	OESTE	ESTE		
10-ago	UNA	OESTE	ESTE		
SUCRE	UNA	ESTE	OESTE		
COLÓN	DOBLE	ESTE	OESTE		
COLÓN	UNA	ESTE	OESTE		
ALCEDO	UNA	OESTE	ESTE		
HURTADO	UNA	ESTE	OESTE		
OLMEDO	UNA	OESTE	ESTE		
CUENCA	UNA	ESTE	OESTE		
FEBRES CORDERO	UNA	ESTE	OESTE		
HUANCAVILCA	UNA	ESTE	OESTE		
P P GÓMEZ	UNA	OESTE	ESTE		
AYACUCHO	UNA	ESTE	OESTE		
CPTN NAJERA	UNA	OESTE	ESTE		
BRASIL	UNA	OESTE	ESTE		
MANABÍ	UNA	OESTE	ESTE		

Tabla 5.3: Sentido de las vías del centro de Guayaquil

CENTRO DE GUAYAQUIL					
CALLE	VÍA	ORIGEN	DESTINO	DESDE	HASTA
MALECÓN	UNA	SUR	NORTE		
PANAMÁ	UNA	NORTE	SUR		
PICHINCHA	UNA	NORTE	SUR		
ROCAFUERTE	UNA	SUR	NORTE		
P CARBO	UNA	SUR	NORTE		
CÓRDOVA	UNA	NORTE	SUR	J CORONEL	09-oct
CHILE	UNA	NORTE	SUR	09-oct	ADELANTE
CHIMBORAZO	UNA	SUR	NORTE		
B MORENO	UNA	SUR	NORTE		
ESCOBEDO	UNA	NORTE	SUR		
BOYACÁ	UNA	NORTE	SUR		
XIMENA	UNA	SUR	NORTE		
RIOBAMBA	UNA	NORTE	SUR		
RUMICHACA	UNA	SUR	NORTE		
L GARAYCOA	UNA	NORTE	SUR		
STA ELENA	UNA	NORTE	SUR		
G AVILÉS	UNA	NORTE	SUR		
06-mar	UNA	NORTE	SUR	VELEZ	SUR
06-mar	UNA	SUR	NORTE	01-may	URDANETA
P MONCAYO	UNA	SUR	NORTE		
QUITO	UNA	SUR	NORTE		
MACHALA	UNA	NORTE	SUR		
ANTEPARA	UNA	NORTE	SUR		
G MORENO	UNA	SUR	NORTE		
AV EJÉRCITO	UNA	SUR	NORTE		
J MASCOTE	UNA	NORTE	SUR		
ESMERALDAS	UNA	SUR	NORTE		
LOS RIOS	UNA	NORTE	SUR		
TULCÁN	DOBLE	NORTE	SUR	09-oct	NORTE
TULCÁN	UNA	NORTE	SUR	09-oct	SUR
CARCHI	UNA	SUR	NORTE		
TUNGURAHUA	UNA	NORTE	SUR		
L. GARCÍA	UNA	SUR	NORTE		
ELOY ALFARO	UNA	SUR	NORTE		

Tabla 5.3: (Continuación) Sentido de las vías del centro de Guayaquil

2. Con la herramienta Google Earth, se calculó la distancia euclidiana, en metros, de esquina a esquina, en cada bloque de edificaciones o de la esquina de una calle hasta la esquina en la cual el camión toma la izquierda o derecha, según el sentido de la calle.

La distancia euclidiana o euclídea es la distancia "ordinaria" entre dos puntos de un espacio vectorial real de dimensión finita (euclídeo)[14].

Google Earth, permite crear rutas y graba la distancia total desde su inicio hasta el final de dicha ruta, sin embargo es una distancia euclidiana, por lo tanto no era factible crear rutas y usar esas distancias; por lo anterior se debió usar el método de esquina a esquina.

Por otro lado, las posiciones calculadas con GPS pueden tener errores de 15 a 100 m, pero esto depende del número de satélites visibles en un momento y

[14] *Wikipedia – Distancia euclidiana - http://es.wikipedia.org/wiki/Distancia_euclidiana - última modificación 25-dic-09 16:06.*

posición determinados[15]; por lo tanto la distancia calculada entre un par de puntos también puede tener errores incluso al medir en línea recta.

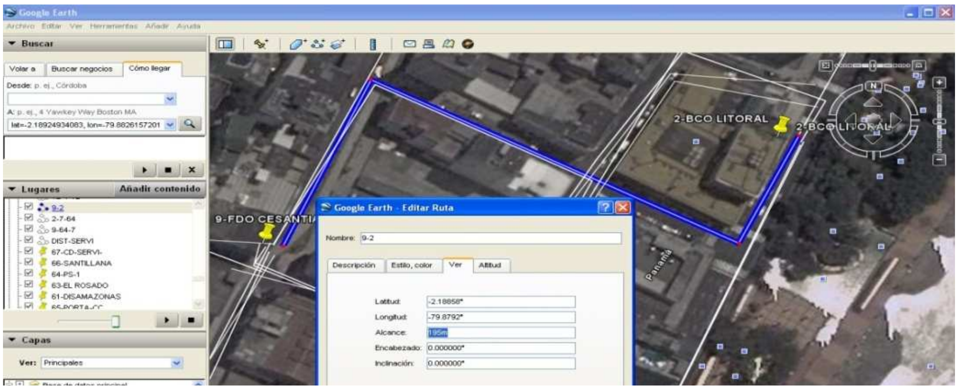


Figura 5.2: Ruta creada en Google Earth, del cliente 9 al 2: el camión avanza por la calle Rocafuerte, gira a la derecha por Luzurraga y toma el Malecón Simón Bolívar hasta llegar al cliente 2, distancia euclidiana = 195 metros

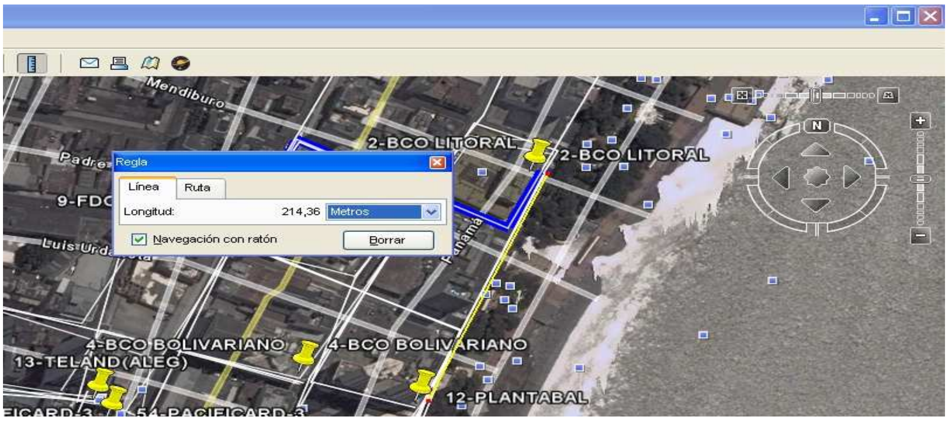


Figura 5.3: Medición de distancia con segmento de recta en Google Earth, El camión avanza por el Malecón Simón Bolívar, distancia euclidiana del cliente 12 al 2 = 214.36 m

3. Se elaboró matriz de distancias (en metros), y luego la matriz de tiempo de viaje entre cada par de nodos, los resultados de estas mediciones se exponen en el anexo A.

4. Se consultó con los conductores de las zonas del centro, concluyendo que una velocidad promedio en esa zona es de 30 Km / Hr. La velocidad máxima

[15] *Wikipedia – Sistema de posicionamiento global - http://es.wikipedia.org/wiki/GPS - última modificación 30-dic-09 22:31.*

permitida para vehículos de carga en el perímetro urbano según las normas de tránsito es de 40 Km / Hr[16].

Como referencia[17], después de analizar las velocidades, en el año 2009, en un sistema de distribución primaria de productos de consumo masivo con doble conductor, se pudo concluir que la velocidad promedio fue de 30 Km / Hr. aproximadamente.

5. Debido a que la zona del centro, casco comercial y bancario por tradición, es una zona muy transitada, de calles cortas y con semáforos en cada esquina, se decidió penalizar la velocidad con un factor del 10 %, considerando entonces 27 Km / Hr para el cálculo de los desplazamientos entre cada par de clientes (anexo A); fórmula: Tiempo = distancia / velocidad.

5.4.5 Ventanas de atención, demanda y tiempo de servicio de los clientes

Ventanas de atención.-

La empresa identifica dos tipos de clientes corporativos, dentro del servicio de recolección:

"Previa llamada - P" y "Diario - D", siendo los primeros aquellos que se visitan el día y la hora solicitado en la semana, previa su llamada, y los diarios son aquellos que se visitan diariamente sin que ello implique un requerimiento previo.

Los clientes P, no tienen una hora programada de recolección pues siempre varía, mientras que los clientes D si tienen una hora acordada de recolección diaria, pero, ninguno tiene una ventana de atención asociada. Al momento de construir los indicadores se crearon las ventanas de atención a partir de la hora programada, 25 minutos antes para el límite inferior y 25 minutos después para el límite superior.

16 *Manual del conductor de la comisión de tránsito del Guayas.*
17 *Transporte por carreteras de Ecuador, con peso aproximado de 25 toneladas, datos de empresa de transporte de carga pesada.*

CLIENTE	DEMAND (Kg.)	LIM INF - horas	LIM SUP - horas	T SERV - horas	CLIENTE	DEMAND (Kg.)	LIM INF - horas	LIM SUP - horas	T SERV - horas
0	0	13.00	22.00	0.00	34	23	16.75	17.58	0.08
1	648	17.50	18.33	0.25	35	10	16.33	17.17	0.08
2	108	19.42	20.25	0.08	36	46	18.83	19.67	0.08
3	390	17.67	18.50	0.17	37	23	15.33	16.17	0.08
4	375	19.08	19.92	0.17	38	61	15.25	16.08	0.08
5	180	15.67	16.50	0.08	39	30	15.67	16.50	0.08
6	144	17.92	18.75	0.08	40	35	17.33	18.17	0.08
7	180	15.83	16.67	0.08	41	54	19.08	19.92	0.08
8	180	18.33	19.17	0.08	42	612	18.58	19.42	0.33
9	185	17.25	18.08	0.08	43	180	18.08	18.92	0.08
10	90	18.50	19.33	0.08	44	32	15.42	16.25	0.08
11	576	18.08	18.92	0.33	45	39	16.67	17.50	0.08
12	150	18.92	19.75	0.08	46	1620	17.58	18.42	0.42
13	115	16.83	17.67	0.08	47	540	16.00	16.83	0.17
14	100	16.08	16.92	0.08	48	149	16.42	17.25	0.08
15	165	18.75	19.58	0.08	49	534	19.08	19.92	0.25
16	22	16.83	17.67	0.08	50	61	17.25	18.08	0.08
17	174	17.00	17.83	0.08	51	2304	17.58	18.42	0.50
18	123	15.75	16.58	0.08	52	46	16.58	17.42	0.08
19	252	18.17	19.00	0.08	53	326	17.00	17.83	0.17
20	18	16.42	17.25	0.08	54	261	16.83	17.67	0.08
21	36	17.50	18.33	0.08	55	54	19.08	19.92	0.08
22	35	16.42	17.25	0.08	56	1058	18.58	19.42	0.33
23	72	15.42	16.25	0.08	57	196	15.75	16.58	0.08
24	156	15.92	16.75	0.08	58	180	17.92	18.75	0.08
25	144	16.50	17.33	0.08	59	144	17.67	18.50	0.08
26	36	16.25	17.08	0.08	60	180	18.08	18.92	0.08
27	1260	18.50	19.33	0.33	61	180	16.75	17.58	0.08
28	434	17.75	18.58	0.33	62	185	17.83	18.67	0.08
29	648	19.08	19.92	0.33	63	261	18.92	19.75	0.08
30	76	17.33	18.17	0.08	64	1836	19.58	20.42	0.33
31	112	17.50	18.33	0.08	65	70	17.17	18.00	0.08
32	191	16.92	17.75	0.08	66	348	15.08	15.92	0.17
33	14	15.25	16.08	0.08					

Tabla 5.4: Demanda, ventanas de atención y tiempo de servicio de cada cliente, teniendo el CD demanda y tiempo de servicio 0

Demanda.-

La empresa normalmente cobra por kilo transportado, sin embargo, la empresa no tiene un registro del peso real transportado. A partir de abril de 2009, luego de solicitar los datos de peso la empresa comenzó a comprar balanzas y pesar la carga en el CD, al descargar el camión luego de terminar su ruta. Luego comenzaron a registrar el peso de cada paquete en su sistema de información pero aún no tienen reportes que permitan consultar los datos ni integrarlos con el sistema de facturación.

Por lo anterior, no se pudo contar con datos reales de peso transportado por cliente; con los datos de cantidad de paquetes por cliente, con la experiencia de los supervisores y con los pesos facturados por código de cliente, se estimó un peso promedio.

Tiempos de servicio.-
Los tiempos de servicio se estimaron desde 5 hasta 30 minutos, dependiendo del cliente.

5.5 Resultados de las pruebas y comparación con los datos de la empresa

Para aplicar la Metaheurística propuesta al problema de recolección de Courier se realizaron 15 corridas del algoritmo; se consideró una flota con capacidad homogénea de 5.5 Toneladas, el cual es un promedio de la capacidad de los vehículos. Para correr el algoritmo, se utilizan los mismos parámetros definidos en el capítulo 4 y se resumen a continuación.

- Tamaño de la población: *POP_SIZE*=100
- Número de hijos generados: *ELITE_SIZE*=80
- Probabilidad de mutación : *PM*=0.60
- Probabilidad de combinación : 1
- Penalidad por atraso : *pen_atraso*=10000
- Costo fijo por ruta: C_k= 100000
- Criterio de parada : 40 iteraciones

Los resultados de estas pruebas se exponen la tabla 5.5; el costo en cada solución es el tiempo de desplazamiento (en horas) acumulado en todas las rutas, más el tiempo de servicio de los clientes. Se observa que la mejor solución encontrada (corrida 13) tiene un costo de 15.21 hrs donde el tiempo total de espera es de 33:00 minutos (0.55 horas) y el tiempo total de atraso de 00:36 segundos (0.0099 horas).

Prueba	No. Rutas	Costo (hrs)	Atraso (hrs)	Espera(hrs)

1	4	22.41	0.660	0.030
2	4	23.04	0.830	0.020
3	4	14.66	1.790	0.320
4	4	22.9	1.510	0.099
5	4	15.42	0.520	0.010
6	4	17.21	0.970	0.020
7	4	17.09	0.950	0.080
8	4	15.43	0.650	0.020
9	4	11.34	0.480	0.320
10	4	27.62	1.270	0.010
11	4	23.26	1.080	0.020
12	4	15.15	0.930	0.010
13	4	15.21	0.550	0.010
14	4	15.07	0.610	0.210
15	4	17.45	0.540	0.010

Tabla 5.5: Resultados de las 15 pruebas realizadas con los datos de picking del courier

La evolución de la espera y atraso acumulados con respecto a las iteraciones de la prueba se ilustran en las figuras 5.4 y 5.5 respectivamente; se observa que la tendencia de la infactibilidad de tiempo tiene un comportamiento similar al de las pruebas realizadas en el capítulo 4.

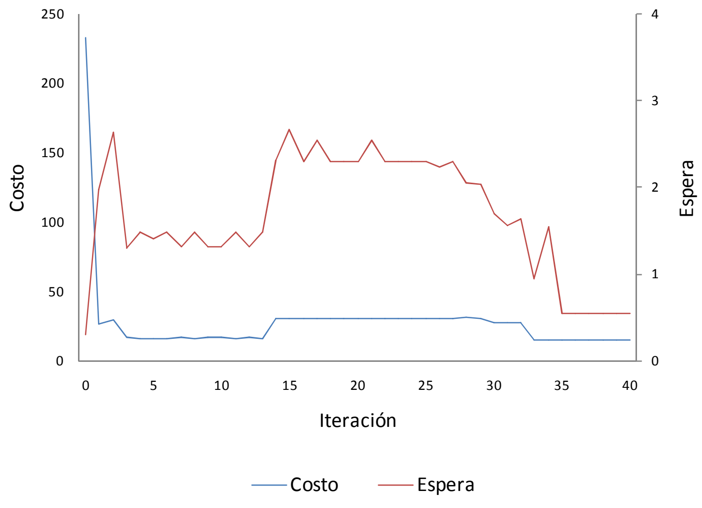


Figura 5.4: Costo de la solución vs el tiempo total de espera en la prueba con los datos de picking del courier

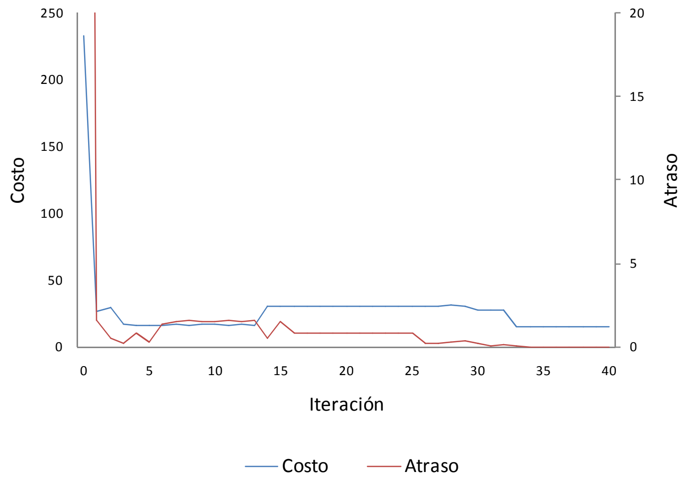


Figura 5.5: Costo de la solución vs el tiempo toral de atraso en la prueba con los datos de picking del courier

Las rutas que componen la solución al problema de picking del Courier se exponen en la tabla 5.6.

RUTA	ORDEN DE VISITAS	CLIENTES
1	0-5-33-66-39-20-45-26-32-48-61-34-17-9-21-27-10-56-4-12-29-0	20
2	0-7-38-37-23-57-24-35-52-25-54-65-51-60-43-62-36-63-55-15-49-0	20
3	0-14-44-18-47-22-13-16-53-40-31-30-59-3-46-6-58-0	16
4	0-1-50-19-28-11-8-42-41-2-64-0	10

Tabla 5.6: Solución al problema de picking del courier

Los clientes y los tiempos se detallan en la tabla 5.7. Se puede observar que el camión solo debe esperar en los clientes 27 y 36, de las rutas 1 y 2, y el incumplimiento del servicio, por atraso, se produce en el cliente 23 de la ruta 2, sin embargo el tiempo de atraso de 36 segundos no es significativo. En la práctica, para el registro de indicadores, se podría acordar que si los tiempos de atraso son menores a 1 minuto se consideran nulos.

Respecto al tiempo de espera, el área de ventas podría realizar una gestión con dichos clientes y cambiar la hora de recolección para reducir el tiempo de espera, por supuesto se debe explicar al cliente el objetivo del requerimiento y probablemente se llegue a ceder en otro aspecto que compense su colaboración.

RUTA	CLIENTE	ARRIBO	INICIO	ESPERA	ATRASO	PARTIDA	SERVICIO
Ruta 1	0					15.34	
	5	15.66	15.66	0.00	0.00	15.74	0.08
	33	15.77	15.77	0.00	0.00	15.85	0.08
	66	16.17	16.17	0.00	0.00	16.34	0.17
	39	16.45	16.45	0.00	0.00	16.53	0.08
	20	16.59	16.59	0.00	0.00	16.67	0.08
	45	16.74	16.74	0.00	0.00	16.82	0.08
	26	16.89	16.89	0.00	0.00	16.97	0.08
	32	17.03	17.03	0.00	0.00	17.11	0.08
	48	17.23	17.23	0.00	0.00	17.31	0.08
	61	17.42	17.42	0.00	0.00	17.50	0.08
	34	17.59	17.59	0.00	0.00	17.67	0.08
	17	17.75	17.75	0.00	0.00	17.83	0.08
	9	17.90	17.90	0.00	0.00	17.98	0.08
	21	18.01	18.01	0.00	0.00	18.09	0.08
	27	18.13	18.50	0.37	0.00	18.83	0.33
	10	18.86	18.86	0.00	0.00	18.94	0.08
	56	18.96	18.96	0.00	0.00	19.29	0.33
	4	19.32	19.32	0.00	0.00	19.49	0.17
	12	19.51	19.51	0.00	0.00	19.59	0.08
	29	19.65	19.65	0.00	0.00	19.98	0.33
	0	20.41					
Ruta 2	0					15.52	
	7	15.83	15.83	0.00	0.00	15.91	0.08
	38	15.96	15.96	0.00	0.00	16.04	0.08
	37	16.04	16.04	0.00	0.00	16.12	0.08
	23	16.26	16.26	0.00	0.01	16.34	0.08
	57	16.55	16.55	0.00	0.00	16.63	0.08
	24	16.70	16.70	0.00	0.00	16.78	0.08
	35	16.98	16.98	0.00	0.00	17.06	0.08
	52	17.12	17.12	0.00	0.00	17.20	0.08
	25	17.27	17.27	0.00	0.00	17.35	0.08
	54	17.43	17.43	0.00	0.00	17.51	0.08
	65	17.52	17.52	0.00	0.00	17.60	0.08
	51	17.66	17.66	0.00	0.00	18.16	0.50
	60	18.18	18.18	0.00	0.00	18.26	0.08
	43	18.33	18.33	0.00	0.00	18.41	0.08
	62	18.48	18.48	0.00	0.00	18.56	0.08
	36	18.65	18.83	0.18	0.00	18.91	0.08
	63	18.97	18.97	0.00	0.00	19.05	0.08
	55	19.09	19.09	0.00	0.00	19.17	0.08
	15	19.18	19.18	0.00	0.00	19.26	0.08
	49	19.36	19.36	0.00	0.00	19.61	0.25
	0	19.86					

RUTA	CLIENTE	ARRIBO	INICIO	ESPERA	ATRASO	PARTIDA	SERVICIO
Ruta 3	0					15.77	
	14	16.08	16.08	0.00	0.00	16.16	0.08
	44	16.23	16.23	0.00	0.00	16.31	0.08
	18	16.39	16.39	0.00	0.00	16.47	0.08
	47	16.6	16.6	0.00	0.00	16.77	0.17
	22	16.81	16.81	0.00	0.00	16.89	0.08
	13	16.93	16.93	0.00	0.00	17.02	0.09
	16	17.06	17.06	0.00	0.00	17.14	0.08
	53	17.14	17.14	0.00	0.00	17.31	0.17
	40	17.37	17.37	0.00	0.00	17.45	0.08
	31	17.55	17.55	0.00	0.00	17.63	0.08
	30	17.63	17.63	0.00	0.00	17.71	0.08
	59	17.74	17.74	0.00	0.00	17.82	0.08
	3	17.83	17.83	0.00	0.00	18.00	0.17
	46	18.06	18.06	0.00	0.00	18.48	0.42
	6	18.54	18.54	0.00	0.00	18.62	0.08
	58	18.62	18.62	0.00	0.00	18.7	0.08
	0	18.87					0.00
Ruta 4	0					17.18	
	1	17.50	17.50	0.00	0.00	17.75	0.25
	50	18.07	18.07	0.00	0.00	18.15	0.08
	19	18.24	18.24	0.00	0.00	18.32	0.08
	28	18.32	18.32	0.00	0.00	18.65	0.33
	11	18.67	18.67	0.00	0.00	19.00	0.33
	8	19.01	19.01	0.00	0.00	19.09	0.08
	42	19.19	19.19	0.00	0.00	19.52	0.33
	41	19.52	19.52	0.00	0.00	19.60	0.08
	2	19.65	19.65	0.00	0.00	19.73	0.08
	64	19.79	19.79	0.00	0.00	20.12	0.33
	0	20.43					0.00

Tabla 5.7: Detalle de rutas de la solución al problema de picking del courier

Entre los indicadores de la empresa, se calculó el nivel de servicio ajustado - NSA, el cual precisamente mide el cumplimiento de la hora programada para visitar al cliente, siendo los resultados:

2009		ENERO		FEBRERO		MARZO		ABRIL	
		CUSTOMERS VISITED		CUSTOMERS VISITED		CUSTOMERS VISITED		CUSTOMERS VISITED	
SECTOR	COD SECTOR	CLIENTES AVG / DIA	NSA=1	CLIENTES AVG / DIA	NSA=1	CLIENTES AVG / DIA	NSA=1	CLIENTES AVG / DIA	NSA=1
BANCARIO	C1	21	0.63	21	0.69	21	0.7	20	0.75
BAHIA	C2	18	0.55	18	0.61	20	0.59	20	0.63
1 LADO 9 OCT	C3	18	0.78	18	0.77	18	0.9	19	0.86
2 LADO 9 OCT	C4	8	0.68	8	0.81	8	0.88	7	0.73

Tabla 5.8: Detalle de indicadores de cumplimiento de la hora programada de vista, desde enero hasta abril

El indicador de la tabla 5.8 muestra, por ejemplo, que en la ruta C1 en el mes de abril, en promedio se visitaron 20 clientes por día y el nivel de servicio fue de 75 %, es decir que en promedio 5 clientes fueron visitados después del horario máximo acordado. Según estos datos, el nivel de servicio promedio del primer cuatrimestre de 2009, en estas 4 rutas, fue de 69.35 %, es decir, 20 clientes no fueron atendidos a tiempo en dicho periodo. La Metaheurística desarrollada mejora la situación actual de la empresa pues reduce la cantidad de clientes no atendidos a tiempo (un cliente), es decir el nivel de servicio sería de 98.46 %.

A continuación se resumen los tiempos de la solución al problema de recolección, sin embargo, no es posible comparar los datos de la tabla 5.9 debido que la empresa no tiene registros de los tiempos reales, excepto el tiempo de atraso analizado en párrafos anteriores:

RUTAS	T ESPERA	T ATRASO	T SERVICIO	T RUTA	TOTAL
1	0:22:12	0:00:00	2:31:48	2:10:12	5:04:12
2	0:10:48	0:00:36	2:11:24	1:57:36	4:20:24
3	0:00:00	0:00:00	1:54:00	1:12:00	3:06:00
4	0:00:00	0:00:00	1:58:12	1:16:48	3:15:00
TOTAL	0:33:00	0:00:36	8:35:24	6:36:36	15:45:36

Tabla 5.9: Resumen de tiempos de la solución al problema de ruteo de picking del courier

Para demostrar la factibilidad de la implementación de un sistema de soporte a decisiones para la planificación de rutas, usando la API de Google Maps, se desarrolló una interfaz web que muestra el detalle de cada una de las rutas; se ilustran algunas salidas en la figura 5.6.

Figura 5.6: Interfaz prototipo de una aplicación DSS para planificación de rutas usando la API de Google Maps[18]

Este prototipo fue diseñado según el esquema expuesto en la sección 3.4.3, y consiste en una aplicación desarrolla en código PHP, el cual es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Los datos de la

[18] *Al momento de la impresión de este documento, el prototipo se encuentra disponible en: www.ecualogistic. com/ruteo.php*

solución del problema de ruteo se extraen de una base de datos MySql, junto con información georeferencial de los clientes. Para poder utilizar mapas de Google online se debe registrar la dirección del sitio web y obtener una clave[19]; luego de lo cual, y mediante código javascript, se puede manipular este tipo de tecnología según la aplicación a desarrollar.

Tal como se expone en la figura 5.6 se pudo dibujar la ubicación exacta de cada cliente y mostrar toda la información de la planificación de rutas. Cabe recalcar que este tipo de soluciones informáticas son más sencillas comparadas con las aplicaciones de escritorio, porque deben ser instaladas en un solo equipo (el servidor), y son independientes del sistema operativo de la máquina cliente; además, poseen la ubicuidad facilitada por el internet, siendo incluso accesibles desde dispositivos móviles, por ejemplo teléfonos celulares que manejan tecnología WAP[20].

[19] *Registro en el API de Google Maps, http://code.google.com/intl/es/apis/maps/signup.html, revisado en enero 5 del 2010.*

[20] ***WAP** son las siglas de **Wireless Application Protocol** (protocolo de aplicaciones inalámbricas), un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos como PDAs, teléfonos móviles, buscapersonas, walkie-talkies y teléfonos inteligentes (smartphones). http://www.masadelante.com/faqs/wap , revisado en enero 5 del 2010.*

5.6 Prueba con ventanas horarias extendidas para todos los clientes

CLIENTE	DEMAND (Kg.)	LIM INF - horas	LIM SUP - horas	T SERV - horas	CLIENTE	DEMAND (Kg.)	LIM INF - horas	LIM SUP - horas	T SERV - horas
0	0	13.00	22.00	0	34	23	15.00	20.50	0.08
1	648	15.00	20.50	0.25	35	10	15.00	20.50	0.08
2	108	15.00	20.50	0.08	36	46	15.00	20.50	0.08
3	390	15.00	20.50	0.17	37	23	15.00	20.50	0.08
4	375	15.00	20.50	0.17	38	61	15.00	20.50	0.08
5	180	15.00	20.50	0.08	39	30	15.00	20.50	0.08
6	144	15.00	20.50	0.08	40	35	15.00	20.50	0.08
7	180	15.00	20.50	0.08	41	54	15.00	20.50	0.08
8	180	15.00	20.50	0.08	42	612	15.00	20.50	0.33
9	185	15.00	20.50	0.08	43	180	15.00	20.50	0.08
10	90	15.00	20.50	0.08	44	32	15.00	20.50	0.08
11	576	15.00	20.50	0.33	45	39	15.00	20.50	0.08
12	150	15.00	20.50	0.08	46	1620	15.00	20.50	0.42
13	115	15.00	20.50	0.08	47	540	15.00	20.50	0.17
14	100	15.00	20.50	0.08	48	149	15.00	20.50	0.08
15	165	15.00	20.50	0.08	49	534	15.00	20.50	0.25
16	22	15.00	20.50	0.08	50	61	15.00	20.50	0.08
17	174	15.00	20.50	0.08	51	2304	15.00	20.50	0.5
18	123	15.00	20.50	0.08	52	46	15.00	20.50	0.08
19	252	15.00	20.50	0.08	53	326	15.00	20.50	0.17
20	18	15.00	20.50	0.08	54	261	15.00	20.50	0.08
21	36	15.00	20.50	0.08	55	54	15.00	20.50	0.08
22	35	15.00	20.50	0.08	56	1058	15.00	20.50	0.33
23	72	15.00	20.50	0.08	57	196	15.00	20.50	0.08
24	156	15.00	20.50	0.08	58	180	15.00	20.50	0.08
25	144	15.00	20.50	0.08	59	144	15.00	20.50	0.08
26	36	15.00	20.50	0.08	60	180	15.00	20.50	0.08
27	1260	15.00	20.50	0.33	61	180	15.00	20.50	0.08
28	434	15.00	20.50	0.33	62	185	15.00	20.50	0.08
29	648	15.00	20.50	0.33	63	261	15.00	20.50	0.08
30	76	15.00	20.50	0.08	64	1836	15.00	20.50	0.33
31	112	15.00	20.50	0.08	65	70	15.00	20.50	0.08
32	191	15.00	20.50	0.08	66	348	15.00	20.50	0.17
33	14	15.00	20.50	0.08					

Tabla 5.10: Nuevas ventanas de atención para los clientes

Aunque, según los resultados analizados en la sección 5.5, el nivel de servicio mejora con la aplicación del sistema desarrollado, es conveniente explorar si existen otras alternativas para reducir los costos del ruteo. Una opción es extender las ventanas de tiempo a todos los clientes para luego limitarlas según los resultados obtenidos. Con este propósito se realizó una corrida del algoritmo, fijando el tiempo de atención más temprana 15:00 y el más tardío de 20:30, manteniendo la demanda y tiempo de servicio anteriores.

En la prueba con las ventanas de tiempo extendidas se logró reducir el costo a 12.74 horas, es decir 16.24 % con relación a los resultados con las ventanas de atención reales, sin espera ni atraso, acumulados. La solución a este problema se expone en la tabla 5.11 y el detalle de las rutas en la tabla 5.12.

Ruta	Orden de visitas	Clientes
1	0-2-56-55-54-53-19-31-26-25-24-17-50-66-59-62-12-11-8-13-63-52-33-36-48-47-0	25
2	0-1-57-27-18-30-21-20-60-65-7-14-41-16-15-51-44-35-34-0	18
3	0-3-61-9-23-28-6-5-43-49-45-40-39-58-0	13
4	0-4-32-29-22-46-37-42-38-10-64-0	10

Tabla 5.11: Solución al problema de picking del Courier con ventanas horarias extendidas

Al comparar los resultados de la tabla 5.13 (ventanas extendidas) con la tabla 5.9 (ventanas normales), se puede apreciar que los tiempos se reducen significativamente. La empresa puede entonces realizar un trabajo conjunto con el área de ventas y con cada uno de sus clientes con el objetivo de mejorar su planificación y el nivel de servicio.

RUTA	CLIENTE	ARRIBO	INICIO	ESPERA	ATRASO	PARTIDA	SERVICIO
Ruta 1	0					14.67	
	2	15.00	15.00	0.00	0.00	15.08	0.08
	56	15.09	15.09	0.00	0.00	15.42	0.33
	55	15.42	15.42	0.00	0.00	15.50	0.08
	54	15.50	15.50	0.00	0.00	15.58	0.08
	53	15.58	15.58	0.00	0.00	15.75	0.17
	19	15.78	15.78	0.00	0.00	15.86	0.08
	31	15.89	15.89	0.00	0.00	15.97	0.08
	26	15.98	15.98	0.00	0.00	16.06	0.08
	25	16.06	16.06	0.00	0.00	16.14	0.08
	24	16.14	16.14	0.00	0.00	16.22	0.08
	17	16.26	16.26	0.00	0.00	16.34	0.08
	50	16.37	16.37	0.00	0.00	16.45	0.08
	66	16.66	16.66	0.00	0.00	16.83	0.17
	59	16.83	16.83	0.00	0.00	16.91	0.08
	62	16.91	16.91	0.00	0.00	16.99	0.08
	12	17.01	17.01	0.00	0.00	17.09	0.08
	11	17.09	17.09	0.00	0.00	17.42	0.33
	8	17.43	17.43	0.00	0.00	17.51	0.08
	13	17.55	17.55	0.00	0.00	17.63	0.08
	63	17.66	17.66	0.00	0.00	17.74	0.08
	52	17.75	17.75	0.00	0.00	17.83	0.08
	33	17.84	17.84	0.00	0.00	17.92	0.08
	36	17.95	17.95	0.00	0.00	18.03	0.08
	48	18.03	18.03	0.00	0.00	18.11	0.08
	47	18.11	18.11	0.00	0.00	18.28	0.17
	0	18.6					
Ruta 3	0					14.68	
	3	15.00	15.00	0.00	0.00	15.17	0.17
	61	15.19	15.19	0.00	0.00	15.27	0.08
	9	15.29	15.29	0.00	0.00	15.37	0.08
	23	15.45	15.45	0.00	0.00	15.53	0.08
	28	15.56	15.56	0.00	0.00	15.89	0.33
	6	15.92	15.92	0.00	0.00	16.00	0.08
	5	16.00	16.00	0.00	0.00	16.08	0.08
	43	16.13	16.13	0.00	0.00	16.21	0.08
	49	16.26	16.26	0.00	0.00	16.51	0.25
	45	16.53	16.53	0.00	0.00	16.61	0.08
	40	16.66	16.66	0.00	0.00	16.74	0.08
	39	16.74	16.74	0.00	0.00	16.82	0.08
	58	16.86	16.86	0.00	0.00	16.94	0.08
	0	17.11					

RUTA	CLIENTE	ARRIBO	INICIO	ESPERA	ATRASO	PARTIDA	SERVICIO
Ruta 2	0					14.68	
	1	15.00	15.00	0.00	0.00	15.25	0.25
	57	15.42	15.42	0.00	0.00	15.50	0.08
	27	15.52	15.52	0.00	0.00	15.85	0.33
	18	15.89	15.89	0.00	0.00	15.97	0.08
	30	15.98	15.98	0.00	0.00	16.06	0.08
	21	16.06	16.06	0.00	0.00	16.14	0.08
	20	16.14	16.14	0.00	0.00	16.22	0.08
	60	16.24	16.24	0.00	0.00	16.32	0.08
	65	16.32	16.32	0.00	0.00	16.40	0.08
	7	16.41	16.41	0.00	0.00	16.49	0.08
	14	16.51	16.51	0.00	0.00	16.59	0.08
	41	16.62	16.62	0.00	0.00	16.70	0.08
	16	16.73	16.73	0.00	0.00	16.81	0.08
	15	16.81	16.81	0.00	0.00	16.89	0.08
	51	16.93	16.93	0.00	0.00	17.43	0.50
	44	17.47	17.47	0.00	0.00	17.55	0.08
	35	17.58	17.58	0.00	0.00	17.66	0.08
	34	17.66	17.66	0.00	0.00	17.74	0.08
	0	18.04					
Ruta 4	0					14.68	
	4	15.00	15.00	0.00	0.00	15.17	0.17
	32	15.24	15.24	0.00	0.00	15.32	0.08
	29	15.36	15.36	0.00	0.00	15.69	0.33
	22	15.73	15.73	0.00	0.00	15.81	0.08
	46	15.87	15.87	0.00	0.00	16.29	0.42
	37	16.31	16.31	0.00	0.00	16.39	0.08
	42	16.40	16.40	0.00	0.00	16.73	0.33
	38	16.74	16.74	0.00	0.00	16.82	0.08
	10	16.86	16.86	0.00	0.00	16.94	0.08
	64	16.95	16.95	0.00	0.00	17.28	0.33
	0	17.59	17.59				

Tabla 5.12: Detalle de rutas de la solución al problema de picking del Courier con ventanas horarias extendidas

RUTAS	T ESPERA	T ATRASO	T SERVICIO	T RUTA	TOTAL
1	0:00:00	0:00:00	2:46:12	1:08:24	3:54:36
2	0:00:00	0:00:00	2:16:48	1:04:48	3:21:36
3	0:00:00	0:00:00	1:33:00	0:52:48	2:25:48
4	0:00:00	0:00:00	1:58:48	0:55:48	2:54:36
TOTAL	0:00:00	0:00:00	8:34:48	4:01:48	12:36:36

Tabla 5.13: Resumen de los tiempos de la solución al problema de ruteo de picking del Courier con ventanas extendidas

Capítulo 6

6. Conclusiones y Recomendaciones

6.1 Conclusiones

- El presente trabajo propone un Sistema de Soporte de Decisiones para planificación de rutas de un Courier, problema que es modelado como uno de ruteo capacitado con ventanas de tiempo suaves (CVRPSTW), y para resolverlo se utiliza un Algoritmo Genético Híbrido de tipo secuencial, es decir usa una población sin ningún tipo de estructura. Durante la implementación de la Metaheurística propuesta, se comprobó que en este tipo de estrategia evolutiva se deben modificar los operadores genéticos (combinación y mutación) para adaptarlos a las restricciones de tiempo y capacidad del problema; caso contrario, un AG básico no podría resolver de forma eficiente un CVRPSTW.

  Incluyendo heurísticas de mejora, tales como 2-Opt*, recolocación, e intercambio, en el operador de mutación, así como utilizando un operador de combinación que tome las mejores rutas de los progenitores, se asegura reducir la infactibilidad de tiempo y costo de la solución, a medida que avanzan las iteraciones, tal como se evidencia en los resultados expuestos en el capítulo 4.

- De acuerdo a las pruebas realizadas con las instancias de Solomon, y analizadas en el capítulo 4, la estrategia desarrollada para resolver el problema CVRPSTW no requiere mayor esfuerzo computacional cuando los datos están agrupados por zonas; se comprobó que en estos casos, el número de iteraciones necesarias para alcanzar una buena solución es menor a 40. Por otro lado, se debe aumentar el número de iteraciones cuando los clientes se encuentran distribuidos de manera completamente aleatoria en una región geográfica. De lo anterior se puede concluir que el AG desarrollado es sensible a la forma de cómo los clientes se encuentren ubicados, siendo más efectivo cuando están agrupados por zonas.

- Al ser un Algoritmo Genético Secuencial, la Metaheurística propuesta explora un espacio solución limitado, el cual se estructura sobre todo por la población inicial, donde la mitad de los individuos se genera por una heurística de construcción y la otra parte de manera aleatoria. Según los resultados obtenidos con las pruebas de Solomon y en el caso de estudio, se concluye que para ampliar el espacio solución es necesario realizar más de una corrida del algoritmo. Lo anterior se evidencia sobre todo cuando las ventanas de tiempo son muy estrechas o los clientes se encuentran ubicados de manera completamente aleatoria.

- Es de suma importancia que este tipo de metaheurísticas evolutivas, usadas para resolver el VRPTW, no sólo considere reducir el costo de desplazamiento; sino además, y en igual grado de importancia, el objetivo de reducir el número de rutas. Por otro lado, reducir el número de vehículos, puede provocar aumento de las infactibilidades de tiempo (esperas y atrasos). Según las pruebas realizadas, y la experiencia durante la codificación algorítmica, cada vez que se aplique una estrategia de reducción de rutas, será necesario aplicar otra para suavizar los incumplimientos de horarios producidos.

- Al realizar el caso de estudio se comprobó que la Metaheurística propuesta tiene un desempeño similar al obtenido en las pruebas de Solomon; esto es: parte de una población de soluciones iniciales y comienza a mejorarlas en cada iteración, reduciendo la infactibilidad de tiempo y el costo de la solución; razón por la cual se puede concluir que es válida para aplicarla en una operación de recolección de un Courier y así obtener una mejor planificación de ruteo. Aunque no se ha mencionado en el presente trabajo, el AG híbrido desarrollado también se puede aplicar en el caso de entrega de carga con ventanas de tiempo.

- Como se definió en el capítulo 2, la población inicial de un AG para resolver el VRPTW se puede estructurar de manera aleatoria o aplicando una heurística de construcción. Se comprobó que una estrategia mixta para la generación de P_0 contribuye a mejorar el espacio de búsqueda inicial y acorta el número de iteraciones necesarias para alcanzar una solución aceptable. Se puede afirmar que en la población inicial algunos individuos tienen cero atrasos, sin embargo, el número de rutas es todavía grande, problema que se supera en las siguientes iteraciones.

- De acuerdo a lo analizado en el capítulo 3, el AG híbrido desarrollado es una parte del Sistema de Soporte de decisiones para la planificación de rutas del Courier; el otro componente importante es el SIG (Sistema de Información Geográfica). Una alternativa a este modelo es el uso de la API de Google Maps o el uso de Google Earth; sin embargo, se dificulta su implementación, porque este tipo de tecnología requiere la adición de una capa con toda la información georeferencial y vial, necesaria para el cálculo de los desplazamientos entre nodos, ya que estos datos no están incorporados en los mapas de ciertas ciudades como Guayaquil.

- Existen pocos estudios de problemas de ruteo con ventanas de tiempo suaves (la mayoría aplican ventanas duras), a pesar de que en los problemas de distribución y recolección reales algunos clientes si aceptan retrasos. La complejidad de estos problemas y de implementar algoritmos para resolverlos, hace que los investigadores se inclinen por el más sencillo (ventanas duras), dejando de lado la posibilidad de tener un modelo más realista y que otros continúen con estudios más profundos al respecto.

- En Ecuador, existen asociaciones de transporte y de empresas courier, pero estas instituciones no tienen por objetivo encontrar soluciones a problemas de transporte, a través de la investigación y desarrollo, en cooperación con centros de estudios; en general buscan normar algunas actividades de sus miembros o adquirir fortaleza con la cantidad de socios para la adjudicación de contratos. Por otro lado, los centros de estudio tampoco realizan acercamientos efectivos, con ofertas prácticas y de bajo costo que logren atraer a los empresarios.

- Un DSS para problemas logísticos como el de ruteo no puede ser una aplicación independiente; al contrario, debe estar integrado al sistema transaccional (TPS) de la operación del Courier o ser un componente de un sistema ERP; en este sentido, es de suma importancia, en la gestión logística, que tenga conexión con un sistema de administración de flota, por ejemplo, y éste a su vez debe estar unido a los sistemas de finanzas, recursos humanos y mercadeo.

- La empresa que proporcionó los datos para el caso de estudio, detallado en el capítulo 5, tiene problemas que acarrean grandes costos de transporte, los cuales se evidenciaron al comparar su nivel de servicio actual de 69.35 % con el nivel de servicio resultado de las pruebas 98.46 %; éstas indican que se deben usar cuatro camiones para el servicio de recolección de los 66 clientes, mientras que la empresa utiliza al menos cinco camiones para atender los mismos clientes y aún así con un nivel de servicio inferior. Los siguientes pasos debe ser: estudiar los resultados con la empresa para afinar detalles y aplicar dichos resultados.

6.2 Recomendaciones

- Las futuras investigaciones, que se basen en el presente trabajo y que tengan por objetivo mejorar sus resultados, deberán usar un Algoritmo Genético con población estructurada, ya sea distribuida o celular, debido que este tipo de técnica reporta ampliar el espacio de búsqueda, por ende aumenta la probabilidad de hallar una mejor solución al problema VRPTW, además de que disminuye el tiempo de ejecución del algoritmo. La ventaja de esta opción es que permite una ejecución paralela la cual ayuda a trabajar con problemas muy complejos, y de mayor tamaño.

- La Metaheurística evolutiva, desarrollada en el presente trabajo, considera un modelo de rutas con flota de capacidad homogénea, pero ésta se puede extender para resolver un problema con capacidad heterogénea; en tal caso, se recomienda usar un vector con las capacidades de cada vehículo el cual sería otro parámetro del algoritmo.

- Para efectos de convergencia y de respetar las ventanas de tiempo al inicio de la ejecución del algoritmo, no siempre será conveniente usar el número real de vehículos disponibles. Se recomienda usar por lo menos el doble de la cantidad mínima de vehículos necesarios, es decir:

  Número de Vehículos = 2 x (Total demanda) / (capacidad de Vehículo)

- Para obtener una mejor solución y ampliar el espacio de búsqueda se deberán realizar entre 5 y 10 corridas de la Metaheurística propuesta; 5 si las ventanas de tiempo de los clientes no son muy estrechas y 10 si las ventanas de tiempo sí lo son. Esto se sustenta de acuerdo a los resultados obtenidos en el caso de estudio con los datos de picking, donde fue necesario realizar más de cinco corridas del algoritmo para obtener una solución mejorada.

- En caso de que se opte por usar la API de Google Maps como sustituto del SIG en el desarrollo de un DSS para el problema de ruteo del Courier, se deberá realizar un levantamiento de datos geográficos y viales de la ciudad donde se desee aplicarlo, en caso de que no exista dicha información; el costo de esta actividad junto al desarrollo informático de la interfaz, necesaria para calcular los tiempos de desplazamiento entre nodos, tendrán que ser incluidos en el proyecto respectivo. En todo caso, esta alternativa deberá contrastarse con la adquisición o compra de un SIG de tipo comercial.

- Se deben realizar más investigaciones de problemas de ruteo, aplicados a casos ecuatorianos; modelos más cercanos a la realidad permitirán mejorar la gestión logística y los niveles de servicio. Por ejemplo la presente investigación puede servir de base para crear un modelo de ruteo dinámico con inclusión de condiciones de tráfico, accidentes e imprevistos durante la entrega. Para estos fines, se requiere un trabajo conjunto con la empresa privada, y conocer sus problemas logísticos en la transportación.

  Es urgente la creación de asociaciones de profesionales en logística, con miembros multidisciplinarios, de reconocida trayectoria laboral en cadenas de abastecimiento e investigadores académicos, que aseguren alcanzar los objetivos científicos y prácticos, a través de la investigación y desarrollo, que puedan ofrecer al mercado soluciones informático-logísticas eficientes y de menor costo. Es necesario que estas agrupaciones integren la gestión logística, los problemas de optimización y las TICs en un solo esfuerzo para resolver los problemas que se plantee.

- Empresas como el caso de estudio, con aproximadamente 1,000 clientes corporativos y con un portafolio de servicios logísticos tan diverso, no debe desarrollar sus operaciones de forma empírica. Está prácticamente obligada a automatizar sus operaciones, adquirir o desarrollar programas logísticos adaptables a los rápidos cambios del entorno. Debe buscar asesoría y utilizar un software de planificación de rutas para atender sus 300 clientes diarios en promedio, solo en la ciudad de Guayaquil.

En un sector tan competitivo como el de los operadores logísticos, donde los precios no son altos y las empresas buscan el margen de contribución de cada producto, con costos de operación crecientes en el mercado ecuatoriano, cada punto que se logre reducir en los costos logísticos será la diferencia.

Bibliografía

[1] Alfredo Olivera, "Heurísticas para Problemas de Ruteo de Vehículos". Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. 2004.

[2] Paolo Toth y Daniele Vigo, "The vehicle routing problem", Universidad de Bologna, Italia.

[3] George Iaonnou, Manolis Kritikos, Gregory Pastracos. "A problem generator-solver heuristic for vehicle routing with soft time windows", Universidad de Economía y negocios, Atenas – Grecia, 2002.

[4] Gonzalo Ignacio Romero Yáñez, "Diseño y desarrollo de un sistema de apoyo a las decisiones de ruteo marítimo para un proveedor de alimentos de la industria del salmón", Universidad de Chile, facultad de ciencias físicas y matemáticas, agosto 2008.

[5] Erwin Delgado, "El problema de la recolección de desechos hospitalarios en la ciudad de Guayaquil, modelación y resolución por medio de una heurística basada en la búsqueda Tabú", Escuela Superior Politécnica del Litoral, Guayaquil – Ecuador, 2007.

[6] Manolo Pinto López, "Teoría de la complejidad algorítmica". Monografías.com. Revisado el 12 de septiembre del 2009. Disponible en http://www.monografias.com/trabajos27/complejidad-algoritmica/complejidad-algoritmica.shtml.

[7] Enrique Alba y Bernabé Dorronsoro. "Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms". Departamento de Ciencias de la Computación de la Universidad de Málaga. 2008.

[8] Enrique Alba y Bernabé Dorronsoro, "Cellular Genetic Algorithms". *Operations research/ Computer Science Interfaces*. Springer. 2008.

[9] Enrique Alba Torres, "Análisis y diseño de algoritmos genéticos paralelos distribuidos, Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga, 1999.

[10] T. Bäck, D.B. Fogel and Z. Michalewicz, editors. "Handbook of Evolutionary Computation. Oxford University Press", 1997.

[11] Áslaug Sóley Bjarnadóttir, "Solving the Vehicle Routing Problem with Genetic Algorithms". Department of Informatics and Mathematical Modelling, IMM Technical University of Denmark, DTU, 2004.

[12] Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga. "Tema 2. Algoritmos Genéticos". Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco.

[13] Zen Hongxin, Zhan Guo Onhui, y Cao Shili. " Directed Edge Recombination Crossover for ATSP". DongGuan University of Technology, GuangDong, China. Springer-Verlag Heidelberg 2006.

[14] Natyhelem Gil Londoño, "Algoritmos Genéticos". Universidad Nacional de Colombia. 2006.

[15] Mitsuo Gen , Runwei Cheng y Lin Lin. "Network Models and Optimization Multiobjective Genetic Algorithm Approach". Springer, 2008.

[16] Thangiagh, Sam R. (1995) Vehicle Routing with Time Windows using Genetic Algorithms. Application Handbook of Genetic Algorithms: New Frontiers, Volume II. Lance Chambers (Ed.), CRS Press, 253-277 (1995).

[17] Kenny Qili Zhu. "A New Genetic Algorithm for VRPTW". National University of Singapore. (2000).

[18] Jarmo T. Alander. On optimal population size of genetic algorithms. In Patrick Dewilde and Joos Vandewalle, editors, CompEuro 1992

[19] Bräysy O., Gendreau M. (2005). “Vehicule routing problem with time windows, part I: Route construction and local search algorithms”. Transportation Science Vol. 39, No. 1, pp 104-118.

[20] Joaquín Antonio Pacheco Bonrostro. “Problemas de rutas con ventanas de tiempo”. Universidad Complutense de Madrid. Dpto. Estadística e Investigación operativa. Facultad de Ciencias Matemáticas.

[21] G. A. P. Kindervarter and M. W. P. Savelsbergh. Local Search in Combinatorial Optimization, chapter Vehicle Routing: handling edge exchanges, pages 337/360, John Wiley & Sons, England, 1997.

[22] E. Taillard, P. Badeau, M. Gendreau, F Guertin, and J. I. Potvin, A Tabu Search Heuristic for the vehicle routing problem with soft time windows, Transportation Science, 31(2), May 1997.

[23] Jorg Homberger, Hermann Gehring. “Two evolutionary metaheuristics for the vehicle routing problem with time windows”. Femuniversitdt hagen, lehrstuhl wirtschaftsinformatik, profustr. 8, D-58084 hagen, Bundesrepublik Deutscland. 1998.

[24] Bernabé Dorronsoro. NEO Networking and Emerging Optimizacion, Software Area, JCell Framework. Librería en Java para aplicaciones de Algoritmos genéticos. Universidad de Málaga. Revisado el 1 de agosto del 2009. Disponible en http://neo.lcc.uma.es/software/jcell/index.php.

[25] Algoritmo Genetico en Java. Foro república digital de Colombia. Revisado el 1 de agosto del 2009. Disponible en http://foro.colombiaunderground.org/index.php?action=printpage;topic=5047.0.

[26] Lenguaje de programación Java, Wikipedia, la enciclopedia libre. Disponible en http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java#Filosof.C3.ADa Revisado 27 noviembre 2009.

[27] Solomon Instances, Informatics and Mathematical Modeling, Technical University of Denmark. Disponible en http://www2.imm.dtu.dk/~jla/solomon.html. Revisado el 4 de diciembre del 2009.

[28] "Best known solutions Solomon's VRPTW 100 customers problems instances". VRP web de Networking and emerging Optmization, Universidad de Málaga. Disponible en http://neo.lcc.uma.es/radi-aeb/WebVRP. Revisado el 20 de diciembre del 2009.

[29] Jorge Hernán Restrepo, Pedro Daniel Medina y Eduardo Ortiz Cruz, Un problema logístico de programación de vehículos con ventanas de tiempo (VRPTW), Universidad Tecnológica de Pereira, septiembre 2008.

[30] Daniel Soto, Wilson Soto, Yoan Pinzón, U. de S. Martín – Grupo de algoritmia y combinatoria ALGOS UN – U. Nacional de Colombia, respectivamente, Una Metaheurística híbrida aplicada a un problema de planificación de rutas, 2008.

[31] R. Andrés Jaqué Pirabán, Métodos aproximados para la solución del problema de enrutamiento de vehículos, Dic. 2008, maestría en ingeniería de sistemas y computación, Universidad Nacional de Colombia.

Anexo A (matriz de tiempos)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0,00	0,32	0,33	0,32	0,32	0,32	0,33	0,31	0,33	0,33	0,34	0,33	0,32	0,31	0,31	0,32
1	0,36	0,00	0,03	0,02	0,03	0,02	0,01	0,04	0,02	0,03	0,02	0,02	0,02	0,03	0,04	0,00
2	0,34	0,02	0,00	0,02	0,01	0,02	0,03	0,01	0,04	0,01	0,04	0,04	0,01	0,02	0,02	0,03
3	0,35	0,03	0,03	0,00	0,02	0,00	0,01	0,04	0,02	0,03	0,02	0,02	0,02	0,03	0,03	0,03
4	0,33	0,01	0,01	0,01	0,00	0,01	0,02	0,02	0,02	0,01	0,03	0,02	0,00	0,01	0,01	0,01
5	0,35	0,03	0,03	0,01	0,02	0,00	0,01	0,03	0,01	0,02	0,02	0,01	0,02	0,02	0,03	0,03
6	0,34	0,04	0,03	0,01	0,03	0,01	0,00	0,04	0,01	0,02	0,01	0,01	0,02	0,03	0,02	0,04
7	0,36	0,04	0,03	0,04	0,03	0,04	0,05	0,00	0,06	0,03	0,06	0,06	0,04	0,04	0,04	0,05
8	0,35	0,03	0,04	0,02	0,03	0,02	0,01	0,05	0,00	0,03	0,00	0,00	0,03	0,02	0,03	0,01
9	0,33	0,01	0,01	0,02	0,01	0,02	0,03	0,02	0,03	0,00	0,04	0,03	0,01	0,01	0,01	0,02
10	0,34	0,02	0,04	0,01	0,03	0,01	0,01	0,04	0,01	0,02	0,00	0,01	0,03	0,02	0,02	0,01
11	0,35	0,03	0,04	0,02	0,03	0,02	0,01	0,05	0,00	0,03	0,00	0,00	0,03	0,02	0,03	0,01
12	0,33	0,01	0,01	0,01	0,00	0,01	0,02	0,02	0,03	0,01	0,03	0,03	0,00	0,01	0,01	0,02
13	0,33	0,00	0,02	0,01	0,01	0,01	0,02	0,03	0,02	0,01	0,03	0,02	0,01	0,00	0,01	0,01
14	0,32	0,01	0,03	0,01	0,02	0,01	0,02	0,03	0,03	0,02	0,02	0,03	0,02	0,01	0,00	0,01
15	0,35	0,03	0,03	0,01	0,02	0,01	0,01	0,04	0,02	0,03	0,02	0,02	0,02	0,03	0,03	0,00
16	0,38	0,05	0,07	0,06	0,07	0,06	0,03	0,07	0,02	0,07	0,02	0,02	0,06	0,05	0,05	0,04
17	0,39	0,06	0,07	0,05	0,06	0,05	0,05	0,08	0,04	0,07	0,05	0,04	0,06	0,06	0,07	0,05
18	0,44	0,09	0,09	0,07	0,08	0,07	0,07	0,09	0,06	0,08	0,06	0,06	0,08	0,08	0,09	0,07
19	0,33	0,02	0,04	0,03	0,03	0,03	0,02	0,04	0,01	0,04	0,01	0,02	0,03	0,02	0,02	0,01
20	0,37	0,05	0,05	0,04	0,04	0,04	0,04	0,06	0,02	0,05	0,03	0,02	0,04	0,05	0,06	0,04
21	0,37	0,05	0,05	0,04	0,04	0,04	0,04	0,06	0,02	0,05	0,03	0,02	0,04	0,05	0,06	0,04
22	0,37	0,05	0,05	0,04	0,04	0,04	0,04	0,06	0,02	0,05	0,03	0,02	0,04	0,05	0,06	0,04
23	0,46	0,11	0,11	0,10	0,10	0,10	0,09	0,12	0,08	0,11	0,08	0,08	0,10	0,11	0,12	0,09
24	0,44	0,09	0,09	0,07	0,08	0,07	0,07	0,09	0,06	0,08	0,06	0,06	0,08	0,08	0,09	0,07
25	0,37	0,04	0,05	0,03	0,04	0,03	0,02	0,06	0,01	0,04	0,02	0,01	0,04	0,04	0,06	0,03
26	0,38	0,06	0,06	0,04	0,05	0,04	0,04	0,06	0,02	0,05	0,03	0,02	0,05	0,05	0,06	0,04
27	0,31	0,02	0,05	0,03	0,04	0,03	0,03	0,05	0,02	0,04	0,02	0,02	0,04	0,02	0,02	0,02
28	0,43	0,08	0,09	0,08	0,08	0,08	0,06	0,10	0,05	0,09	0,06	0,05	0,08	0,08	0,09	0,06
29	0,38	0,06	0,05	0,04	0,05	0,04	0,04	0,06	0,03	0,05	0,03	0,03	0,04	0,05	0,06	0,04
30	0,38	0,06	0,05	0,04	0,05	0,04	0,04	0,06	0,03	0,05	0,03	0,03	0,04	0,05	0,06	0,04
31	0,37	0,06	0,06	0,04	0,05	0,04	0,04	0,06	0,02	0,05	0,03	0,02	0,05	0,05	0,05	0,04
32	0,33	0,10	0,12	0,10	0,11	0,10	0,10	0,12	0,09	0,11	0,09	0,09	0,10	0,10	0,10	0,10
33	0,30	0,03	0,04	0,04	0,04	0,04	0,04	0,03	0,05	0,05	0,05	0,05	0,04	0,03	0,03	0,04
34	0,28	0,06	0,07	0,06	0,07	0,06	0,06	0,08	0,06	0,06	0,06	0,06	0,06	0,05	0,05	0,05
35	0,26	0,08	0,09	0,08	0,08	0,08	0,09	0,07	0,08	0,09	0,08	0,08	0,08	0,07	0,07	0,07
36	0,28	0,07	0,09	0,07	0,08	0,07	0,06	0,09	0,06	0,08	0,06	0,06	0,08	0,07	0,07	0,07
37	0,30	0,06	0,07	0,06	0,06	0,06	0,05	0,08	0,04	0,06	0,04	0,05	0,06	0,05	0,05	0,05
38	0,29	0,03	0,04	0,03	0,04	0,03	0,04	0,03	0,05	0,04	0,05	0,05	0,03	0,03	0,02	0,03
39	0,34	0,11	0,13	0,11	0,12	0,11	0,10	0,13	0,10	0,12	0,10	0,10	0,11	0,11	0,11	0,11
40	0,32	0,02	0,05	0,03	0,04	0,03	0,02	0,04	0,03	0,04	0,04	0,03	0,04	0,02	0,02	0,02
41	0,30	0,03	0,04	0,03	0,04	0,03	0,04	0,03	0,05	0,04	0,05	0,05	0,03	0,03	0,02	0,03
42	0,27	0,07	0,08	0,07	0,07	0,07	0,07	0,09	0,06	0,07	0,06	0,07	0,07	0,06	0,06	0,06
43	0,29	0,04	0,06	0,04	0,05	0,04	0,05	0,06	0,04	0,05	0,04	0,04	0,05	0,04	0,04	0,04
44	0,29	0,07	0,09	0,08	0,08	0,08	0,06	0,10	0,06	0,08	0,06	0,06	0,08	0,07	0,07	0,07
45	0,28	0,06	0,07	0,06	0,07	0,06	0,06	0,08	0,06	0,06	0,06	0,06	0,06	0,05	0,05	0,05
46	0,30	0,05	0,07	0,06	0,06	0,06	0,04	0,08	0,03	0,06	0,03	0,03	0,06	0,05	0,05	0,05
47	0,32	0,11	0,13	0,11	0,12	0,11	0,10	0,13	0,10	0,12	0,10	0,10	0,12	0,11	0,11	0,11
48	0,25	0,09	0,10	0,09	0,09	0,09	0,10	0,08	0,09	0,10	0,09	0,09	0,09	0,08	0,08	0,08
49	0,22	0,09	0,10	0,09	0,09	0,09	0,10	0,08	0,10	0,10	0,11	0,10	0,09	0,08	0,08	0,09
50	0,32	0,02	0,04	0,03	0,03	0,03	0,02	0,04	0,03	0,04	0,03	0,03	0,03	0,01	0,01	0,01
51	0,28	0,05	0,06	0,05	0,05	0,05	0,06	0,04	0,05	0,06	0,05	0,05	0,05	0,04	0,04	0,04
52	0,35	0,03	0,03	0,01	0,02	0,00	0,01	0,03	0,01	0,02	0,02	0,01	0,02	0,02	0,03	0,03
53	0,34	0,02	0,03	0,00	0,02	0,01	0,01	0,04	0,01	0,02	0,02	0,01	0,02	0,01	0,02	0,02
54	0,33	0,00	0,02	0,01	0,01	0,01	0,02	0,03	0,02	0,01	0,03	0,02	0,01	0,00	0,01	0,01
55	0,32	0,02	0,04	0,02	0,03	0,02	0,02	0,03	0,03	0,04	0,03	0,03	0,03	0,01	0,01	0,01
56	0,34	0,02	0,03	0,00	0,02	0,01	0,01	0,04	0,01	0,02	0,02	0,01	0,02	0,01	0,02	0,02
57	0,17	0,15	0,16	0,15	0,16	0,15	0,16	0,15	0,17	0,16	0,17	0,17	0,15	0,15	0,14	0,15
58	0,34	0,02	0,02	0,01	0,01	0,01	0,01	0,03	0,02	0,02	0,03	0,02	0,01	0,02	0,03	0,03
59	0,34	0,02	0,02	0,01	0,01	0,01	0,01	0,03	0,02	0,02	0,03	0,02	0,01	0,02	0,03	0,03
60	0,34	0,02	0,02	0,01	0,01	0,01	0,01	0,03	0,02	0,02	0,03	0,02	0,01	0,02	0,03	0,03
61	0,36	0,00	0,03	0,02	0,03	0,02	0,01	0,04	0,02	0,03	0,02	0,02	0,02	0,03	0,04	0,00
62	0,34	0,02	0,02	0,01	0,01	0,01	0,01	0,03	0,02	0,02	0,03	0,02	0,01	0,02	0,03	0,03
63	0,31	0,02	0,04	0,02	0,03	0,02	0,02	0,03	0,03	0,04	0,03	0,03	0,03	0,01	0,03	0,01
64	0,34	0,03	0,01	0,02	0,02	0,03	0,03	0,01	0,04	0,01	0,05	0,04	0,02	0,02	0,03	0,03
65	0,34	0,02	0,02	0,01	0,01	0,01	0,01	0,03	0,02	0,02	0,03	0,02	0,01	0,02	0,03	0,03
66	0,01	0,31	0,32	0,31	0,31	0,31	0,32	0,30	0,32	0,32	0,33	0,32	0,31	0,30	0,30	0,31

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0,33	0,36	0,37	0,32	0,35	0,35	0,35	0,40	0,37	0,35	0,35	0,31	0,36	0,38	0,38	0,37
1	0,03	0,06	0,08	0,04	0,04	0,04	0,04	0,10	0,08	0,04	0,04	0,03	0,07	0,07	0,07	0,04
2	0,05	0,08	0,10	0,04	0,06	0,06	0,06	0,13	0,10	0,07	0,06	0,03	0,09	0,10	0,10	0,06
3	0,03	0,06	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,03	0,07	0,07	0,07	0,04
4	0,04	0,06	0,08	0,03	0,05	0,05	0,05	0,11	0,08	0,05	0,05	0,03	0,07	0,08	0,08	0,05
5	0,03	0,06	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,03	0,07	0,07	0,07	0,04
6	0,02	0,05	0,06	0,02	0,03	0,03	0,03	0,09	0,06	0,03	0,03	0,02	0,06	0,06	0,06	0,03
7	0,08	0,11	0,13	0,07	0,09	0,09	0,09	0,15	0,13	0,10	0,09	0,06	0,12	0,12	0,12	0,09
8	0,01	0,04	0,06	0,02	0,03	0,03	0,03	0,08	0,06	0,02	0,03	0,02	0,05	0,06	0,06	0,03
9	0,05	0,08	0,09	0,03	0,06	0,06	0,06	0,12	0,09	0,06	0,06	0,03	0,09	0,09	0,09	0,06
10	0,03	0,06	0,07	0,04	0,03	0,03	0,03	0,10	0,07	0,02	0,03	0,01	0,06	0,07	0,07	0,03
11	0,01	0,04	0,06	0,02	0,03	0,03	0,03	0,08	0,06	0,02	0,03	0,02	0,05	0,06	0,06	0,03
12	0,04	0,07	0,08	0,04	0,05	0,05	0,05	0,11	0,08	0,07	0,05	0,03	0,08	0,09	0,09	0,05
13	0,04	0,06	0,08	0,04	0,05	0,05	0,05	0,11	0,08	0,05	0,05	0,03	0,07	0,08	0,08	0,05
14	0,04	0,07	0,08	0,02	0,05	0,05	0,05	0,11	0,08	0,05	0,05	0,02	0,08	0,08	0,08	0,05
15	0,03	0,06	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,03	0,07	0,07	0,09	0,04
16	0,00	0,03	0,05	0,02	0,02	0,02	0,02	0,07	0,05	0,06	0,02	0,04	0,04	0,05	0,05	0,03
17	0,04	0,00	0,06	0,06	0,02	0,02	0,02	0,08	0,06	0,02	0,02	0,08	0,05	0,01	0,01	0,03
18	0,03	0,02	0,00	0,08	0,04	0,04	0,04	0,03	0,00	0,05	0,04	0,07	0,00	0,03	0,03	0,03
19	0,02	0,05	0,06	0,00	0,03	0,03	0,03	0,09	0,06	0,03	0,03	0,02	0,06	0,06	0,06	0,06
20	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,00	0,04	0,03	0,04	0,04	0,02
21	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,00	0,04	0,03	0,04	0,04	0,02
22	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,00	0,04	0,03	0,04	0,04	0,02
23	0,05	0,04	0,03	0,11	0,06	0,06	0,06	0,00	0,03	0,08	0,06	0,10	0,03	0,06	0,06	0,06
24	0,03	0,02	0,00	0,08	0,04	0,04	0,04	0,03	0,00	0,05	0,04	0,07	0,00	0,03	0,03	0,03
25	0,02	0,03	0,04	0,04	0,01	0,01	0,01	0,07	0,04	0,00	0,01	0,06	0,04	0,04	0,04	0,02
26	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,00	0,04	0,03	0,04	0,04	0,02
27	0,03	0,06	0,07	0,01	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,00	0,07	0,07	0,07	0,06
28	0,02	0,02	0,00	0,08	0,04	0,04	0,04	0,03	0,00	0,05	0,04	0,07	0,00	0,04	0,04	0,03
29	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,01	0,05	0,04	0,00	0,00	0,02
30	0,02	0,02	0,04	0,05	0,00	0,00	0,00	0,07	0,04	0,01	0,01	0,05	0,04	0,00	0,00	0,02
31	0,02	0,02	0,04	0,05	0,01	0,01	0,01	0,06	0,04	0,01	0,01	0,04	0,03	0,04	0,04	0,00
32	0,07	0,10	0,08	0,09	0,09	0,09	0,09	0,11	0,08	0,12	0,09	0,09	0,08	0,12	0,12	0,10
33	0,05	0,08	0,08	0,03	0,06	0,06	0,06	0,11	0,08	0,07	0,06	0,03	0,08	0,09	0,09	0,08
34	0,06	0,09	0,10	0,04	0,07	0,07	0,07	0,13	0,10	0,08	0,07	0,05	0,10	0,11	0,11	0,10
35	0,09	0,12	0,12	0,07	0,10	0,10	0,10	0,15	0,12	0,10	0,10	0,07	0,11	0,13	0,13	0,12
36	0,06	0,09	0,11	0,04	0,07	0,07	0,07	0,14	0,11	0,08	0,07	0,06	0,10	0,11	0,11	0,10
37	0,05	0,08	0,09	0,03	0,06	0,06	0,06	0,12	0,09	0,06	0,06	0,05	0,09	0,09	0,09	0,08
38	0,05	0,07	0,08	0,03	0,06	0,06	0,06	0,11	0,08	0,07	0,06	0,03	0,08	0,09	0,09	0,08
39	0,08	0,09	0,07	0,08	0,10	0,10	0,10	0,10	0,07	0,11	0,10	0,10	0,06	0,10	0,10	0,10
40	0,03	0,06	0,08	0,01	0,05	0,05	0,05	0,11	0,08	0,09	0,05	0,01	0,07	0,08	0,08	0,06
41	0,05	0,07	0,08	0,03	0,06	0,06	0,06	0,11	0,08	0,07	0,06	0,03	0,08	0,09	0,09	0,08
42	0,07	0,10	0,11	0,05	0,08	0,08	0,08	0,14	0,11	0,09	0,08	0,06	0,10	0,12	0,12	0,11
43	0,05	0,08	0,08	0,03	0,06	0,06	0,06	0,11	0,08	0,06	0,06	0,03	0,08	0,09	0,09	0,08
44	0,06	0,09	0,11	0,04	0,07	0,07	0,07	0,14	0,11	0,08	0,07	0,07	0,10	0,11	0,11	0,10
45	0,06	0,09	0,10	0,04	0,07	0,07	0,07	0,13	0,10	0,08	0,07	0,05	0,10	0,11	0,11	0,10
46	0,03	0,07	0,07	0,02	0,04	0,04	0,04	0,10	0,07	0,05	0,04	0,04	0,07	0,08	0,08	0,07
47	0,10	0,13	0,15	0,08	0,11	0,11	0,11	0,18	0,15	0,12	0,11	0,10	0,14	0,15	0,15	0,14
48	0,10	0,13	0,13	0,08	0,11	0,11	0,11	0,16	0,13	0,11	0,11	0,08	0,12	0,14	0,14	0,13
49	0,10	0,13	0,14	0,09	0,12	0,12	0,12	0,16	0,14	0,12	0,12	0,08	0,13	0,14	0,14	0,13
50	0,03	0,06	0,06	0,01	0,04	0,04	0,04	0,09	0,06	0,05	0,04	0,01	0,06	0,07	0,07	0,06
51	0,06	0,09	0,09	0,04	0,07	0,07	0,07	0,12	0,09	0,07	0,07	0,04	0,09	0,10	0,10	0,09
52	0,03	0,06	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,03	0,07	0,07	0,07	0,04
53	0,03	0,05	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,02	0,07	0,07	0,07	0,04
54	0,04	0,06	0,08	0,04	0,05	0,05	0,05	0,11	0,08	0,05	0,05	0,03	0,07	0,08	0,08	0,05
55	0,03	0,05	0,06	0,01	0,04	0,04	0,04	0,09	0,06	0,05	0,04	0,00	0,06	0,07	0,07	0,06
56	0,03	0,05	0,07	0,03	0,04	0,04	0,04	0,10	0,07	0,04	0,04	0,02	0,07	0,07	0,07	0,04
57	0,17	0,19	0,20	0,15	0,18	0,18	0,18	0,23	0,20	0,19	0,18	0,14	0,20	0,21	0,21	0,20
58	0,05	0,08	0,09	0,05	0,06	0,06	0,06	0,12	0,09	0,05	0,06	0,04	0,09	0,08	0,08	0,06
59	0,05	0,08	0,09	0,05	0,06	0,06	0,06	0,12	0,09	0,05	0,06	0,04	0,09	0,08	0,08	0,06
60	0,05	0,08	0,09	0,05	0,06	0,06	0,06	0,12	0,09	0,05	0,06	0,04	0,09	0,08	0,08	0,06
61	0,03	0,06	0,08	0,04	0,04	0,04	0,04	0,10	0,08	0,04	0,04	0,03	0,07	0,07	0,07	0,04
62	0,05	0,08	0,09	0,05	0,06	0,06	0,06	0,12	0,09	0,05	0,06	0,04	0,09	0,08	0,08	0,06
63	0,03	0,05	0,06	0,01	0,04	0,04	0,04	0,09	0,06	0,04	0,04	0,00	0,05	0,06	0,06	0,05
64	0,05	0,08	0,10	0,06	0,07	0,07	0,07	0,12	0,10	0,06	0,07	0,05	0,09	0,10	0,10	0,07
65	0,05	0,08	0,09	0,05	0,06	0,06	0,06	0,12	0,09	0,05	0,06	0,04	0,09	0,08	0,08	0,06
66	0,32	0,35	0,36	0,31	0,34	0,34	0,34	0,39	0,36	0,34	0,34	0,30	0,35	0,36	0,36	0,35

	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
0	0,32	0,29	0,31	0,28	0,31	0,29	0,29	0,34	0,31	0,29	0,30	0,29	0,29	0,31	0,29	0,33	0,28
1	0,08	0,06	0,08	0,10	0,09	0,06	0,06	0,09	0,06	0,06	0,07	0,05	0,06	0,08	0,05	0,11	0,10
2	0,10	0,04	0,08	0,08	0,09	0,06	0,04	0,11	0,03	0,05	0,08	0,06	0,07	0,08	0,06	0,11	0,08
3	0,07	0,05	0,07	0,09	0,09	0,05	0,06	0,09	0,06	0,06	0,07	0,05	0,06	0,07	0,05	0,11	0,10
4	0,08	0,03	0,08	0,07	0,09	0,05	0,04	0,10	0,03	0,04	0,07	0,05	0,06	0,08	0,05	0,11	0,08
5	0,07	0,05	0,07	0,09	0,08	0,05	0,05	0,09	0,06	0,06	0,07	0,05	0,06	0,07	0,05	0,11	0,09
6	0,06	0,04	0,07	0,08	0,08	0,04	0,05	0,08	0,05	0,05	0,06	0,04	0,05	0,07	0,04	0,10	0,09
7	0,12	0,06	0,11	0,10	0,12	0,09	0,06	0,14	0,06	0,06	0,10	0,08	0,10	0,11	0,08	0,14	0,10
8	0,06	0,05	0,07	0,09	0,08	0,04	0,05	0,07	0,03	0,05	0,06	0,04	0,05	0,07	0,04	0,10	0,09
9	0,09	0,03	0,07	0,07	0,09	0,05	0,03	0,11	0,02	0,03	0,07	0,05	0,06	0,07	0,05	0,11	0,07
10	0,07	0,05	0,06	0,08	0,07	0,04	0,05	0,09	0,02	0,05	0,06	0,04	0,05	0,06	0,04	0,09	0,09
11	0,06	0,05	0,07	0,09	0,08	0,04	0,05	0,07	0,03	0,05	0,06	0,04	0,05	0,07	0,04	0,10	0,09
12	0,08	0,04	0,08	0,08	0,09	0,06	0,04	0,10	0,03	0,04	0,07	0,05	0,07	0,08	0,05	0,11	0,08
13	0,08	0,03	0,08	0,07	0,09	0,06	0,03	0,10	0,06	0,03	0,08	0,06	0,07	0,08	0,06	0,11	0,07
14	0,08	0,02	0,07	0,06	0,08	0,04	0,02	0,10	0,01	0,03	0,06	0,04	0,05	0,07	0,04	0,10	0,06
15	0,07	0,06	0,07	0,09	0,08	0,05	0,06	0,09	0,06	0,06	0,07	0,05	0,06	0,07	0,05	0,11	0,10
16	0,04	0,07	0,09	0,12	0,07	0,10	0,07	0,06	0,03	0,07	0,08	0,06	0,10	0,09	0,10	0,08	0,12
17	0,08	0,09	0,13	0,13	0,11	0,11	0,10	0,10	0,07	0,10	0,12	0,10	0,12	0,13	0,10	0,13	0,14
18	0,10	0,11	0,15	0,18	0,13	0,16	0,12	0,06	0,08	0,12	0,14	0,10	0,15	0,15	0,16	0,14	0,18
19	0,06	0,04	0,07	0,07	0,08	0,04	0,04	0,08	0,02	0,05	0,06	0,04	0,05	0,07	0,04	0,10	0,07
20	0,06	0,08	0,09	0,12	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,09	0,06	0,11	0,12
21	0,06	0,08	0,09	0,12	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,09	0,06	0,11	0,12
22	0,06	0,08	0,09	0,12	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,09	0,06	0,11	0,12
23	0,13	0,14	0,17	0,20	0,16	0,19	0,14	0,09	0,11	0,14	0,17	0,12	0,18	0,17	0,18	0,16	0,21
24	0,10	0,11	0,15	0,18	0,13	0,16	0,12	0,06	0,08	0,12	0,14	0,10	0,15	0,15	0,16	0,14	0,18
25	0,06	0,08	0,11	0,12	0,09	0,08	0,08	0,08	0,05	0,08	0,10	0,08	0,09	0,11	0,08	0,11	0,12
26	0,06	0,08	0,09	0,12	0,09	0,07	0,08	0,08	0,05	0,09	0,09	0,07	0,08	0,09	0,06	0,11	0,12
27	0,07	0,04	0,05	0,05	0,06	0,03	0,04	0,09	0,01	0,05	0,04	0,02	0,04	0,05	0,02	0,08	0,06
28	0,10	0,11	0,14	0,17	0,13	0,16	0,11	0,06	0,08	0,11	0,14	0,09	0,15	0,14	0,15	0,13	0,18
29	0,07	0,08	0,10	0,12	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,10	0,07	0,12	0,12
30	0,07	0,08	0,10	0,12	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,10	0,07	0,12	0,12
31	0,06	0,07	0,09	0,11	0,09	0,07	0,08	0,08	0,05	0,08	0,09	0,07	0,08	0,09	0,06	0,11	0,12
32	0,00	0,10	0,05	0,08	0,03	0,06	0,10	0,02	0,08	0,10	0,04	0,06	0,05	0,05	0,06	0,04	0,08
33	0,09	0,00	0,08	0,04	0,09	0,05	0,00	0,11	0,04	0,00	0,07	0,05	0,06	0,08	0,05	0,14	0,04
34	0,03	0,05	0,00	0,03	0,02	0,02	0,08	0,05	0,04	0,06	0,01	0,02	0,01	0,00	0,02	0,07	0,03
35	0,06	0,04	0,03	0,00	0,05	0,03	0,05	0,07	0,06	0,05	0,04	0,05	0,03	0,03	0,03	0,10	0,00
36	0,02	0,07	0,02	0,05	0,00	0,03	0,07	0,03	0,05	0,07	0,01	0,03	0,02	0,02	0,03	0,02	0,05
37	0,03	0,05	0,05	0,05	0,03	0,00	0,05	0,05	0,04	0,06	0,04	0,02	0,04	0,05	0,01	0,04	0,05
38	0,09	0,01	0,07	0,04	0,08	0,05	0,00	0,11	0,03	0,00	0,07	0,05	0,06	0,07	0,05	0,13	0,04
39	0,04	0,11	0,08	0,11	0,07	0,10	0,11	0,00	0,09	0,11	0,08	0,07	0,09	0,08	0,09	0,07	0,12
40	0,08	0,04	0,06	0,06	0,07	0,04	0,04	0,09	0,00	0,04	0,05	0,03	0,05	0,06	0,03	0,09	0,07
41	0,09	0,03	0,07	0,04	0,09	0,05	0,01	0,11	0,03	0,00	0,07	0,05	0,06	0,07	0,05	0,14	0,05
42	0,04	0,06	0,01	0,02	0,02	0,02	0,08	0,05	0,05	0,06	0,00	0,02	0,01	0,01	0,03	0,05	0,03
43	0,04	0,04	0,02	0,03	0,04	0,01	0,04	0,06	0,02	0,04	0,02	0,00	0,02	0,02	0,01	0,06	0,03
44	0,03	0,07	0,01	0,03	0,02	0,03	0,07	0,04	0,05	0,07	0,01	0,03	0,00	0,01	0,03	0,04	0,03
45	0,03	0,05	0,00	0,03	0,02	0,02	0,08	0,05	0,04	0,06	0,01	0,02	0,01	0,00	0,02	0,07	0,03
46	0,04	0,05	0,04	0,04	0,04	0,00	0,05	0,05	0,03	0,05	0,04	0,01	0,02	0,04	0,00	0,05	0,05
47	0,06	0,11	0,06	0,09	0,04	0,07	0,11	0,07	0,09	0,11	0,05	0,07	0,06	0,06	0,07	0,00	0,09
48	0,07	0,05	0,02	0,01	0,04	0,04	0,06	0,08	0,07	0,06	0,03	0,06	0,03	0,02	0,04	0,06	0,00
49	0,09	0,05	0,08	0,05	0,08	0,06	0,05	0,11	0,08	0,06	0,07	0,06	0,06	0,08	0,06	0,10	0,05
50	0,07	0,04	0,06	0,06	0,07	0,03	0,04	0,09	0,01	0,04	0,05	0,03	0,04	0,06	0,03	0,09	0,06
51	0,09	0,01	0,05	0,03	0,08	0,06	0,02	0,10	0,03	0,02	0,06	0,04	0,06	0,05	0,05	0,10	0,03
52	0,07	0,05	0,07	0,09	0,08	0,05	0,05	0,09	0,06	0,06	0,07	0,05	0,06	0,07	0,05	0,11	0,09
53	0,07	0,04	0,07	0,08	0,08	0,05	0,04	0,09	0,06	0,05	0,07	0,05	0,06	0,07	0,05	0,10	0,08
54	0,08	0,03	0,08	0,07	0,09	0,06	0,03	0,10	0,06	0,03	0,08	0,06	0,07	0,08	0,06	0,11	0,07
55	0,07	0,03	0,05	0,06	0,06	0,03	0,04	0,09	0,01	0,04	0,05	0,03	0,04	0,05	0,03	0,09	0,06
56	0,07	0,04	0,07	0,08	0,08	0,05	0,04	0,09	0,06	0,05	0,07	0,05	0,06	0,07	0,05	0,10	0,08
57	0,15	0,12	0,14	0,11	0,14	0,12	0,12	0,17	0,14	0,12	0,13	0,12	0,13	0,14	0,12	0,17	0,11
58	0,10	0,05	0,09	0,09	0,10	0,07	0,05	0,11	0,04	0,05	0,08	0,07	0,08	0,09	0,06	0,12	0,09
59	0,10	0,05	0,09	0,09	0,10	0,07	0,05	0,11	0,04	0,05	0,08	0,07	0,08	0,09	0,06	0,12	0,09
60	0,10	0,05	0,09	0,09	0,10	0,07	0,05	0,11	0,04	0,05	0,08	0,07	0,08	0,09	0,06	0,12	0,09
61	0,08	0,06	0,08	0,10	0,09	0,06	0,06	0,09	0,06	0,06	0,07	0,05	0,06	0,08	0,05	0,11	0,10
62	0,10	0,05	0,09	0,09	0,10	0,07	0,05	0,11	0,04	0,05	0,08	0,07	0,08	0,09	0,06	0,12	0,09
63	0,07	0,05	0,05	0,06	0,06	0,03	0,05	0,09	0,01	0,05	0,05	0,03	0,04	0,05	0,02	0,08	0,06
64	0,10	0,04	0,10	0,08	0,11	0,08	0,05	0,11	0,08	0,05	0,09	0,07	0,09	0,10	0,07	0,13	0,09
65	0,10	0,05	0,09	0,09	0,10	0,07	0,05	0,11	0,04	0,05	0,08	0,07	0,08	0,09	0,06	0,12	0,09
66	0,31	0,27	0,30	0,27	0,30	0,28	0,28	0,33	0,30	0,28	0,29	0,28	0,28	0,30	0,28	0,32	0,27

	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
0	0,23	0,32	0,28	0,32	0,32	0,31	0,31	0,32	0,17	0,33	0,33	0,33	0,32	0,33	0,31	0,31	0,33	0,01
1	0,14	0,07	0,07	0,02	0,01	0,03	0,06	0,01	0,19	0,01	0,01	0,01	0,00	0,01	0,06	0,06	0,01	0,35
2	0,12	0,03	0,06	0,02	0,03	0,02	0,03	0,03	0,17	0,02	0,02	0,02	0,02	0,02	0,03	0,02	0,02	0,33
3	0,13	0,06	0,07	0,00	0,04	0,03	0,06	0,04	0,18	0,01	0,01	0,01	0,03	0,01	0,06	0,05	0,01	0,34
4	0,11	0,02	0,05	0,01	0,02	0,01	0,02	0,02	0,16	0,01	0,01	0,01	0,01	0,01	0,02	0,02	0,01	0,32
5	0,13	0,06	0,07	0,00	0,03	0,02	0,06	0,03	0,18	0,00	0,00	0,00	0,03	0,00	0,06	0,05	0,00	0,34
6	0,12	0,05	0,06	0,01	0,01	0,03	0,05	0,01	0,17	0,01	0,01	0,01	0,04	0,01	0,05	0,03	0,01	0,33
7	0,14	0,05	0,07	0,04	0,05	0,04	0,06	0,05	0,19	0,05	0,05	0,05	0,04	0,05	0,06	0,02	0,05	0,35
8	0,13	0,02	0,06	0,02	0,01	0,02	0,03	0,01	0,18	0,02	0,02	0,02	0,03	0,02	0,03	0,04	0,02	0,34
9	0,11	0,02	0,04	0,02	0,02	0,01	0,02	0,02	0,16	0,02	0,02	0,02	0,01	0,02	0,02	0,01	0,02	0,32
10	0,12	0,02	0,06	0,01	0,01	0,02	0,02	0,01	0,17	0,01	0,01	0,01	0,02	0,01	0,02	0,04	0,01	0,33
11	0,13	0,02	0,06	0,02	0,01	0,02	0,03	0,01	0,18	0,02	0,02	0,02	0,03	0,02	0,03	0,04	0,02	0,34
12	0,11	0,02	0,05	0,01	0,02	0,01	0,03	0,02	0,16	0,02	0,02	0,02	0,01	0,02	0,03	0,03	0,02	0,32
13	0,11	0,07	0,04	0,01	0,01	0,00	0,07	0,01	0,16	0,02	0,02	0,02	0,00	0,02	0,07	0,02	0,02	0,32
14	0,10	0,01	0,04	0,01	0,02	0,01	0,01	0,02	0,15	0,02	0,02	0,02	0,01	0,02	0,01	0,03	0,02	0,31
15	0,13	0,06	0,07	0,01	0,00	0,03	0,06	0,00	0,18	0,01	0,01	0,01	0,03	0,01	0,06	0,05	0,01	0,34
16	0,16	0,03	0,08	0,06	0,05	0,05	0,03	0,05	0,21	0,05	0,05	0,05	0,05	0,05	0,03	0,06	0,05	0,37
17	0,17	0,07	0,11	0,06	0,06	0,06	0,08	0,06	0,22	0,05	0,05	0,05	0,06	0,05	0,08	0,09	0,05	0,38
18	0,22	0,08	0,13	0,08	0,07	0,08	0,08	0,07	0,27	0,07	0,07	0,07	0,09	0,07	0,08	0,11	0,07	0,43
19	0,11	0,03	0,06	0,03	0,01	0,02	0,02	0,01	0,16	0,02	0,02	0,02	0,02	0,02	0,03	0,04	0,02	0,32
20	0,15	0,05	0,09	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,05	0,08	0,03	0,36
21	0,15	0,05	0,09	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,05	0,08	0,03	0,36
22	0,15	0,05	0,09	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,05	0,08	0,03	0,36
23	0,24	0,10	0,15	0,11	0,10	0,11	0,11	0,10	0,29	0,09	0,09	0,09	0,11	0,09	0,11	0,14	0,09	0,45
24	0,22	0,08	0,13	0,08	0,07	0,08	0,08	0,07	0,27	0,07	0,07	0,07	0,09	0,07	0,08	0,11	0,07	0,43
25	0,15	0,05	0,09	0,03	0,03	0,04	0,05	0,03	0,21	0,03	0,03	0,03	0,04	0,03	0,05	0,05	0,03	0,36
26	0,16	0,05	0,10	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,05	0,06	0,03	0,37
27	0,09	0,01	0,06	0,03	0,02	0,02	0,01	0,02	0,14	0,02	0,02	0,02	0,02	0,02	0,01	0,04	0,02	0,30
28	0,21	0,07	0,12	0,08	0,07	0,08	0,08	0,07	0,26	0,07	0,07	0,07	0,08	0,07	0,08	0,11	0,07	0,42
29	0,16	0,05	0,10	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,06	0,08	0,03	0,37
30	0,16	0,05	0,10	0,04	0,04	0,05	0,05	0,04	0,21	0,03	0,03	0,03	0,05	0,03	0,06	0,08	0,03	0,37
31	0,15	0,05	0,09	0,04	0,04	0,05	0,05	0,04	0,20	0,03	0,03	0,03	0,05	0,03	0,05	0,06	0,03	0,36
32	0,11	0,08	0,11	0,10	0,10	0,10	0,09	0,10	0,17	0,10	0,10	0,10	0,10	0,10	0,09	0,12	0,10	0,32
33	0,08	0,03	0,01	0,04	0,04	0,03	0,02	0,04	0,13	0,04	0,04	0,04	0,03	0,04	0,03	0,03	0,04	0,29
34	0,06	0,04	0,05	0,06	0,06	0,05	0,04	0,06	0,11	0,06	0,06	0,06	0,06	0,06	0,05	0,07	0,06	0,27
35	0,04	0,06	0,03	0,08	0,07	0,07	0,06	0,07	0,09	0,08	0,08	0,08	0,08	0,08	0,06	0,07	0,08	0,25
36	0,06	0,06	0,07	0,07	0,07	0,07	0,06	0,07	0,11	0,08	0,08	0,08	0,07	0,08	0,06	0,09	0,08	0,27
37	0,08	0,04	0,05	0,06	0,06	0,05	0,04	0,06	0,13	0,06	0,06	0,06	0,06	0,06	0,05	0,07	0,06	0,29
38	0,07	0,03	0,02	0,03	0,04	0,03	0,02	0,04	0,13	0,04	0,04	0,04	0,03	0,04	0,02	0,03	0,04	0,28
39	0,12	0,09	0,10	0,11	0,11	0,11	0,10	0,11	0,18	0,11	0,11	0,11	0,11	0,11	0,10	0,13	0,11	0,33
40	0,10	0,00	0,06	0,03	0,02	0,02	0,01	0,02	0,15	0,02	0,02	0,02	0,02	0,02	0,01	0,04	0,02	0,31
41	0,08	0,03	0,04	0,03	0,04	0,03	0,02	0,04	0,13	0,04	0,04	0,04	0,03	0,04	0,02	0,03	0,04	0,29
42	0,05	0,05	0,06	0,07	0,06	0,06	0,05	0,06	0,10	0,07	0,07	0,07	0,07	0,07	0,05	0,08	0,07	0,26
43	0,07	0,03	0,03	0,04	0,04	0,04	0,03	0,04	0,12	0,05	0,05	0,05	0,04	0,05	0,03	0,06	0,05	0,28
44	0,07	0,06	0,07	0,08	0,07	0,07	0,06	0,07	0,12	0,08	0,08	0,08	0,07	0,08	0,06	0,09	0,08	0,28
45	0,06	0,04	0,05	0,06	0,06	0,05	0,04	0,06	0,11	0,06	0,06	0,06	0,06	0,06	0,05	0,07	0,06	0,27
46	0,08	0,04	0,05	0,06	0,05	0,05	0,04	0,05	0,13	0,06	0,06	0,06	0,05	0,06	0,04	0,07	0,06	0,29
47	0,10	0,10	0,10	0,11	0,11	0,11	0,10	0,11	0,15	0,11	0,11	0,11	0,11	0,11	0,10	0,13	0,11	0,31
48	0,03	0,07	0,04	0,09	0,08	0,08	0,07	0,08	0,09	0,09	0,09	0,09	0,09	0,09	0,07	0,08	0,09	0,24
49	0,00	0,08	0,05	0,09	0,09	0,08	0,08	0,09	0,05	0,10	0,10	0,10	0,09	0,10	0,08	0,08	0,10	0,21
50	0,10	0,00	0,05	0,03	0,01	0,01	0,00	0,01	0,15	0,02	0,02	0,02	0,02	0,02	0,00	0,03	0,02	0,31
51	0,06	0,03	0,00	0,05	0,05	0,04	0,03	0,05	0,11	0,05	0,05	0,05	0,05	0,05	0,04	0,04	0,05	0,27
52	0,13	0,06	0,07	0,00	0,03	0,02	0,06	0,03	0,18	0,00	0,00	0,00	0,03	0,00	0,06	0,05	0,00	0,34
53	0,12	0,06	0,06	0,01	0,00	0,01	0,06	0,00	0,17	0,01	0,01	0,01	0,02	0,01	0,06	0,03	0,01	0,33
54	0,11	0,07	0,04	0,01	0,01	0,00	0,07	0,01	0,16	0,02	0,02	0,02	0,00	0,02	0,07	0,02	0,02	0,32
55	0,10	0,01	0,05	0,02	0,01	0,01	0,00	0,01	0,15	0,02	0,02	0,02	0,02	0,02	0,00	0,03	0,02	0,31
56	0,12	0,06	0,06	0,01	0,00	0,01	0,06	0,00	0,17	0,01	0,01	0,01	0,02	0,01	0,06	0,03	0,01	0,33
57	0,06	0,15	0,12	0,15	0,16	0,15	0,14	0,16	0,00	0,16	0,16	0,16	0,15	0,16	0,14	0,15	0,16	0,16
58	0,12	0,04	0,06	0,01	0,03	0,02	0,04	0,03	0,18	0,00	0,00	0,00	0,02	0,00	0,04	0,05	0,00	0,33
59	0,12	0,04	0,06	0,01	0,03	0,02	0,04	0,03	0,18	0,00	0,00	0,00	0,02	0,00	0,04	0,05	0,00	0,33
60	0,12	0,04	0,06	0,01	0,03	0,02	0,04	0,03	0,18	0,00	0,00	0,00	0,02	0,00	0,04	0,05	0,00	0,33
61	0,14	0,07	0,07	0,02	0,01	0,03	0,06	0,01	0,19	0,01	0,01	0,01	0,00	0,01	0,06	0,06	0,01	0,35
62	0,12	0,04	0,06	0,01	0,03	0,02	0,04	0,03	0,18	0,00	0,00	0,00	0,02	0,00	0,04	0,05	0,00	0,33
63	0,09	0,01	0,06	0,02	0,01	0,01	0,01	0,01	0,15	0,02	0,02	0,02	0,02	0,02	0,00	0,03	0,02	0,30
64	0,12	0,09	0,06	0,03	0,03	0,02	0,08	0,03	0,17	0,03	0,03	0,03	0,03	0,03	0,08	0,00	0,03	0,33
65	0,12	0,04	0,06	0,01	0,03	0,02	0,04	0,03	0,18	0,00	0,00	0,00	0,02	0,00	0,04	0,05	0,00	0,33
66	0,22	0,31	0,27	0,31	0,31	0,30	0,30	0,31	0,16	0,32	0,32	0,32	0,31	0,32	0,30	0,30	0,32	0,00