

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

Pincay Johnny¹, Tomalá James², Polo Vaca³

Resumen. En este documento se propone un Sistema de Soporte de Decisiones (DSS) basado en un Algoritmo Genético Híbrido y codificado en Java, para resolver un problema de ruteo con las siguientes características: recolección de carga, un solo depósito, flota homogénea de vehículos, demanda y tiempo de servicio determinísticos y ventanas suaves de atención, es decir un problema CVRPSTW. La Metaheurística propuesta tiene como objetivo planificar un conjunto de rutas al mínimo costo y cumpliendo las restricciones del problema. La investigación comienza con un contexto general de los problemas de ruteo, métodos de solución y el modelo matemático; profundiza en el desarrollo e implementación de un algoritmo genético de población no estructurada, integrado con heurísticas de búsqueda local. Se muestran los resultados obtenidos en las instancias de Solomon. El problema planteado se basa en la situación actual de una empresa de courier ecuatoriana, para llevar a la práctica la investigación académica; con sus datos, se probó la Metaheurística desarrollada y se demostró que se alcanzan los dos objetivos de la logística integrada: costo eficiente y mejora del servicio al cliente, al reducir la cantidad de vehículos, los tiempos en ruta y los atrasos en las visitas. El estudio también puede servir para futuros trabajos sobre la planificación de rutas e investigaciones de optimización combinatoria, relacionados con algoritmos genéticos.

Palabras Claves: Sistema de soporte de decisiones, Algoritmo Genético Híbrido, CVRPSTW, y problemas de ruteo.

Abstract. This paper proposes a Decision Support System based on a Hybrid Genetic Algorithm and codified on Java programming language, to solve a routing problem with the following constraints: cargo picking, one depot, homogeneous fleet, deterministic demand and service time and soft time windows, that is, a CVRPSTW. This Metaheuristic calls for a planning of the optimal set of routes with the least cost and respecting all constraints. The investigation begins describing a general context of vehicle routing problems (VRP), solution methods and mathematical models; makes deeper in the development and application of a Genetic Algorithm of non structured population integrated with Local Search Methods. The Metaheuristic was tested on Solomon instances and we show the results. The problem is based on the situation of an Ecuadorian Courier enterprise, because it is necessary to carry to the real world the academic investigation; the algorithm was also proved with the data of this company and reached the integrated logistic objectives: efficient costs and customer service improve, well then reduce the quantity of vehicles and visit delays. This paper will also offer a base for future works about planning routes and investigations of combinatorial optimization problems, related with genetic algorithms.

Keywords: Decision Support System, Hybrid Genetic Algorithm and Vehicle Routing Problems.

Recibido: Febrero, 2010

Aceptado: Marzo, 2010

1. INTRODUCCIÓN

Los potenciales ahorros en las operaciones de transporte, 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.

Uno de los problemas más estudiados de OC es el problema del agente viajero (TSP): consiste en

visitar una y solo una vez un conjunto de clientes, partiendo de un punto inicial y finalmente retornando a éste, siendo una ruta de menor costo. 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). El CVRPSTW, es un problema de ruteo con ventanas de tiempo suaves, es decir, las horas de visitas a los clientes pueden incumplirse a costo de una penalización.

Actualmente, en Ecuador existen aproximadamente 800 agencias de couriers, y todas utilizan métodos empíricos para planificar sus rutas lo cual no garantiza obtener rutas óptimas, encareciendo sus costos de operación y disminuyendo la calidad del servicio a los clientes. 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, también se puede sumar una flota aproximada de 100 vehículos entre camiones y motos. Por lo descrito anteriormente, se define el

¹ Pincay Villa Johnny, Master en Control de Operaciones y Gestión Logística, ESPOL. (e-mail: jr.johnmypr@gmail.com)

² Tomalá Robles James, Master en Control de Operaciones y Gestión Logística, ESPOL. (e-mail: jamesstomala@hotmail.com)

³ Vaca Arellano Polo, Director del Departamento de Matemática, Escuela Politécnica de Nacional. (e-mail: polo.vaca@epn.edu.ec)

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.

2. FORMULACIÓN MATEMÁTICA DE UN CVRPSTW

A continuación se expone la formulación matemática del problema de ruteo a resolver, el cual consiste en un problema de ruteo con restricción de capacidad y ventanas suaves de tiempo, denotado por CVRPSTW (por sus siglas en inglés).

Sea $G = (V, A)$ un grafo dirigido, donde $V = \{v_0, v_1, \dots, v_n, v_{n+1}\}$ 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_0 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 $V_c = \{v_1, \dots, v_n\}$. Una ruta es posible si comienza en el vértice v_0 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_0 y v_{n+1} y para cada cliente v_i se fija una ventana horaria de servicio $[e_{v_i}, l_{v_i}]$, dentro de la cual debe ser visitado. Además la demanda y el tiempo de servicio en el depósito son iguales a cero. Otros parámetros del modelo son:

s_{v_i} : Tiempo de servicio en cliente i

K : Flota homogénea de vehículos estacionada en el depósito.

Q : Capacidad de los vehículos.

f_k : Costo fijo de un vehículo k .

p_{v_i} : Penalización por retraso al atender el cliente v_i .

α_i : Costo por unidad de tiempo de retraso en v_i .

Se definen las variables de decisión:

$$x_{ijk} = \begin{cases} 1 & \text{si el vehículo } k \text{ se desplaza en } (v_i, v_j) \\ 0 & \text{si el vehículo } k \text{ se desplaza en } (v_j, v_i) \end{cases}$$

h_{v_i} = tiempo de inicio del servicio en v_i

$r_{v_i} = \text{Max}\{0, h_{v_i} - l_{v_i}\}$, representa el tiempo de atraso en el servicio al cliente v_i . Se considera la siguiente función objetivo

$$\text{Min}Z = \sum_{k \in K} \sum_{(v_i, v_j) \in A} c_{ij} x_{ijk} + \sum_{k \in K} \sum_{(v_i, v_j) \in A} f_k x_{ijk} + \sum_{i=1}^{n+1} p_{v_i} \quad (1)$$

Sujeta a las siguientes restricciones:

a) Todos los clientes serán visitados una sola vez:

$$\sum_{k \in K} \sum_{v_j \in \Delta^-(v_i)} x_{ijk} = 1 \quad \forall v_i \in V_c \quad (2)$$

b) Cada vehículo parte del depósito:

$$\sum_{v_j \in \Delta^+(v_0)} x_{ijk} = 1 \quad \forall k \in K \quad (3)$$

c) Cada ruta se realiza sólo por un vehículo:

$$\sum_{v_j \in \Delta^-(v_i)} x_{ijk} - \sum_{v_j \in \Delta^+(v_i)} x_{ijk} = 0 \quad \forall k \in K, v_i \in V_c \quad (4)$$

d) Se debe respetar la capacidad del vehículo:

$$\sum_{v_j \in V_c} q_i \sum_{v_j \in \Delta^-(v_i)} x_{ijk} \leq Q \quad \forall k \in K \quad (5)$$

e) Se respetan las ventanas de tiempo y a lo mucho se incurre en el atraso r_{v_i} :

$$e_{v_i} \leq h_{v_i} \leq l_{v_i} + r_{v_i} \quad \forall v_i \in V \quad (6)$$

f) Debe existir continuidad en el orden de visitas:

$$h_{v_j} - h_{v_i} \geq s_{v_i} + t_{ij} - M(1 - x_{ijk}) \quad \forall (v_i, v_j) \in A, k \in K \quad (7)$$

g) Tipo de variables:

$$x_{ijk} \in \{0, 1\} \text{ y } h_{v_i}, r_{v_i} \geq 0 \quad (8)$$

En esta formulación se considera:

$$\Delta^+(v_i) = \{v_j \in V_c \mid (v_i, v_j) \in A\}, \text{ y}$$

$$\Delta^-(v_i) = \{v_j \in V_c \mid (v_j, v_i) \in A\}$$

3. LOS ALGORITMOS GENÉTICOS APLICADOS AL VRPTW

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. Uno de estos métodos es el enfoque evolutivo desarrollado por los Algoritmos Genéticos (AG) [1] [8] [9]. Los AG 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 Goldberg en 1989. En un problema de Optimización Combinatoria, cada solución posible tiene una representación genética

llamada cromosoma o individuo, que es un conjunto de elementos denominados genes. Los diferentes valores que puede tomar un gen se conocen como alelos. Un AG empieza con un conjunto inicial de individuos P_0 . En cada iteración se genera una nueva población P_{t+1} aplicando dos operadores genéticos: combinación (operación con dos cromosomas) y mutación (operación sobre un cromosoma); además, se evalúa la aptitud o fitness, que es una medida de la calidad del individuo [12].

Joe L. Blanton Jr [10] fue el primero en usar Algoritmos Genéticos para resolver el problema VRPTW en 1993; Luego, en 1995, Thangiah [13] usó AG para implementar su método denominado GIDEON, el cual asigna primero clientes a las rutas y luego define el orden de visita 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 [14] donde propone un algoritmo que prueba ser mejor a GIDEON en todas las 56 instancias de Solomon. Entre los principales referentes de este proyecto, se encuentra la investigación de Homberger y Gehring [16] quienes proponen dos estrategias evolutivas híbridas que aplican operadores de mutación y combinación, y utilizan heurísticas clásicas de mejora, tales como el 2-Opt*, modificadas para adaptarlas a las restricciones de tiempo del VRPTW.

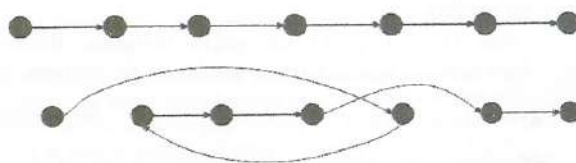
De lo revisado en estas publicaciones se puede concluir que un AG básico no logra buenas aproximaciones a la solución óptima. Toda implementación exitosa de un AG para resolver problemas VRP se convierte en un Algoritmo Genético Híbrido porque requiere algún tipo de adición algorítmica tales como modificación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

4. HEURÍSTICAS PARA LA MEJORA DE SOLUCIONES

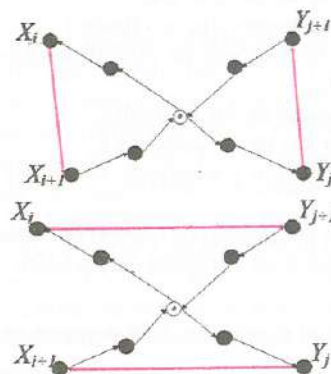
Para que un AG resuelva un problema de tipo VRP, y con el objetivo de mejorar una solución, se utilizan heurísticas de búsqueda local [15], tales como: Or-Opt, 2-Opt*, intercambio y recolocación. El operador Or-Opt se aplica a una ruta y traslada una cadena de clientes de longitud k de una ubicación a otra; por lo general, y debido al costo computacional, se aplica con $k=3$, y luego con un número menor; este caso se ilustra en la figura 1.

FIGURA 1
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Intercambio Or-Opt con $k=3$



2-Opt* es un operador entre rutas, y consiste en intercambiar un segmento de una con otro segmento de otra ruta teniendo en consideración el orden; en la figura 2 se muestra su aplicación, donde básicamente se intercambian los arcos (x_i, x_{i+1}) y (y_j, y_{j+1}) por (x_i, y_{j+1}) y (y_j, x_{i+1}) .

FIGURA 2
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Ejemplo de aplicación del operador 2-Opt*



La recolocación (desplazamiento) consiste en mover un cliente de una ruta a otra, mientras que en el intercambio se procede a intercambiar dos clientes de rutas diferentes.

5. DESCRIPCIÓN DE LA METAHEURÍSTICA DESARROLLADA

El procedimiento propuesto para resolver el CVRPSTW consiste en un Algoritmo Genético Híbrido de tipo secuencial, en otras palabras, no

usa ningún tipo de estructura en la población. Se utiliza una estrategia evolutiva de tipo (μ, λ) -EA, donde a medida que se crea la nueva élite, el hijo generado reemplaza a uno de los padres siempre y cuando tenga una mejor aptitud. El pseudocódigo general se expone en la tabla 1 y no muestra mayor diferencia a un AG básico, sin embargo considera operadores genéticos modificados, estrategias que se describen a continuación.

TABLA 1
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Algoritmo genético propuesto para resolver el problema CVRPSTW

```

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 ← generación_población_inicial()
  /* Fase de reproducción y generación de
  una nueva población */
  WHILE (t ≤ NUM_ITERACIONES) DO
    evaluación_población(Pt)
    Pt ← mutación_padres(Pt)
    Pt ← generar_hijos(Pt)
    t ← t + 1
  END
  Entregar la mejor solución
END

```

5.1 OPERADORES DE MUTACIÓN Y COMBINACIÓN

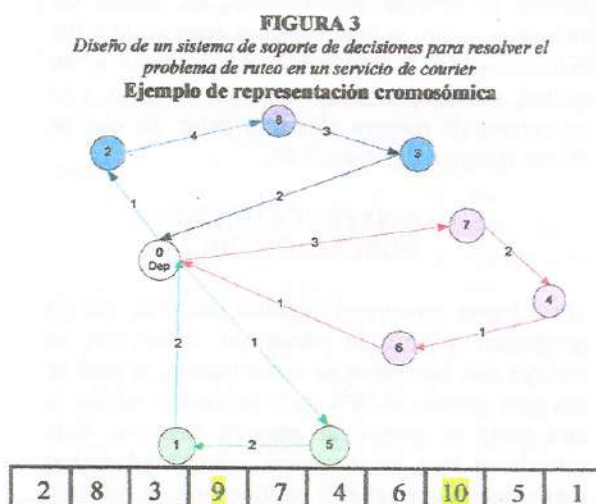
El procedimiento de mutación se basa en una de las estrategias evolutivas propuestas por Homberger y Gehring [16], y consiste en dos fases: primero se realizan tres movimientos de mejora, seleccionado en cada uno de ellos, y de manera aleatoria, una de las heurísticas del conjunto {2-Opt*, recolocación, intercambio}; luego se intenta minimizar el número de rutas, para lo cual se elimina la que tenga mayor costo por cliente y se reinsertan sus elementos en el resto de rutas.

La estrategia de combinación está basada en el operador Uniform Crossover (UC) descrito por Áslaug Sóley Bjarnadóttir [11], la cual consiste en seleccionar las mejores rutas de los padres e insertarlas en el hijo, siempre y cuando no entren en conflicto, es decir, no se incluya un cliente en más de una ruta. Los elementos que no se encuentren insertados, se ubican en la posición menos costosa en alguna ruta; de ser necesario se crea un nuevo circuito siempre y cuando no se supere al número de rutas mínimo de los padres.

Tanto en la mutación y combinación, se utilizan básicamente movimientos de eliminación e inserción; se elige como criterio de inserción el definido por Homberger y Gehring [16] el cual consiste básicamente en insertar un cliente en la posición y ruta que menos atraso produzca.

5.2 REPRESENTACIÓN CROMOSÓMICA

Se adopta la representación de permutación de enteros, donde cada arreglo define el orden de visita de los clientes; existe n clientes, los cuales se permutan en un solo vector; se considera como separador de rutas un entero mayor a n . En la figura 3 se ilustra un ejemplo con tres rutas.



5.3 EVALUACIÓN Y SELECCIÓN

Se definen: el conjunto de vértices $V = \{v_0, v_1, \dots, v_n\}$, donde $v_0 = 0$ representa el depósito; el conjunto $P = \{v_1, \dots, v_n\}$ comprende los n clientes a ser visitados; la matriz de costos $C_{n \times n} = (c_{ij})$ calcula la distancia o tiempo entre cada par de vértices (v_i, v_j) ; $[e_i, l_i]$ representa la ventana horaria aceptada por el cliente i , y s_i su tiempo de servicio.

Dado el tiempo de arribo al nodo i , denotado por t_i , el atraso producido se calcula de la siguiente manera:

$$\Delta_i^+ = \begin{cases} 0 & \text{para el depósito} \\ \max\{t_i - l_i, 0\} & \text{para } i \geq 1 \end{cases} \quad (9)$$

Sea $P_1 | P_2 | \dots | P_m$ una partición de P , entonces se define una ruta como

$R_k = \{v_0^k, v_1^k, \dots, v_r^k, v_{(r+1)}^k\}$, donde $v_0^k = v_{(r+1)}^k = 0$ y $\{v_1^k, \dots, v_r^k\}$ es una permutación del subconjunto P_k . Se calcula el costo de una ruta R_k como:

$$\text{Costo}(R_k) = \sum_{j=0}^r (c_{j, (j+1)} + s_j) \quad (10)$$

El costo por violación de las ventanas de tiempo se denota por $pen_atraso(R_k)$ y se calcula por la siguiente fórmula, donde α es una constante de penalización:

$$pen_atraso(R_k) = \alpha \sum_{j=1}^{r+1} \Delta_j^+ \quad (11)$$

Asumiendo un costo fijo por ruta C_k , una solución compuesta de m rutas tiene el siguiente costo:

$$Costo = \sum_{k=1}^m (C_k + Costo(R_k) + pen_atraso(R_k)) \quad (12)$$

De esta manera, la aptitud de cada individuo (fitness) se define como el costo de la solución respectiva. Para seleccionar al par de padres, durante el proceso reproductivo, se intenta un método sencillo, el cual empieza ordenando a los individuos de la población de acuerdo a su aptitud, es decir, de menor a mayor de costo, y se selecciona de manera aleatoria entre los que se ubican dentro del percentil 40.

5.4 CONSTRUCCIÓN DE LA POBLACIÓN INICIAL

Para lograr soluciones iniciales factibles, que no produzcan sobre todo retraso en el servicio, se incluye una heurística de construcción, la cual se usa para generar el 50% de la población inicial, la otra parte se genera de manera aleatoria. Este método se basa en un procedimiento del vecino más cercano con criterio de inserción de máximo ahorro y sin incurrir en atrasos en el servicio; de esta manera, 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 a la ruta. 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.

6. IMPLEMENTACIÓN Y VALIDACIÓN DE LA METAHEURÍSTICA

A partir de la justificación de Enrique Alba [9] para utilizar Programación Orientada a Objetos (POO) en la implementación de algoritmos genéticos, se selecciona el lenguaje de programación java para la codificación de la Metaheurística; esto se debe por la sencillez de su sintaxis, independencia del sistema operativo, y la ventaja que brindan sus especificaciones, que hacen extensible su uso a dispositivos móviles y aplicaciones web. La implementación de la Metaheurística partió de una revisión de otros trabajos similares [17] [18], y consistió en el desarrollo de dos librerías. La primera agrupa los

componentes principales de la estrategia evolutiva y comprende las clases: algoritmo genético, problema, población, individuo, y ruta. La segunda librería encierra los operadores genéticos, y las heurísticas de búsqueda local y construcción de rutas.

Para validar el programa desarrollado se realizaron pruebas con las instancias de Solomon. En la tabla II se resumen los resultados y se los compara con el óptimo; 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.

TABLA II
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Resultados de pruebas en instancias de Solomon

Instancia	Óptimo		Solución AG propuesto	
	Num rutas	Costo	Num rutas	Costo
C101	10	827.30	10	828.94
C102	10	827.30	10	828.94
R104	10	982.010	10	1,174.84
R111	12	1,048.70	11	1,316.00
RC103	11	1,258.00	11	1,424.34

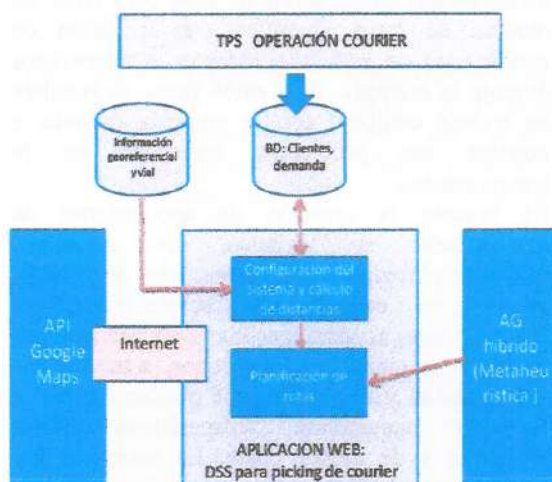
7. DSS PROPUESTO PARA PLANIFICACIÓN DE PICKING EN UN COURIER

Un Sistema de Información Geográfica (SIG) es uno de los principales componentes de un software para la planificación de rutas; una alternativa a la implementación de un SIG puede ser el uso del servicio de mapas a través de la web, como el que ofrece Google Maps, que consiste en 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 [19]. 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.

En la figura 4 se ilustra el esquema general del sistema propuesto, consiste en una aplicación web que utiliza la Metaheurística desarrollada y la API Google Maps. 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 ente cada par de clientes; a su vez, estos datos constituyen entradas en el sistema de planificación de rutas.

FIGURA 4
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Esquema general del DSS para planificación del picking del courier



8. RESULTADOS DEL CASO DE ESTUDIO

Se realizó un caso de estudio con los datos de una de las empresas de Courier más grandes que operan en el país. Se seleccionaron 65 clientes ubicados principalmente en el centro de la ciudad de Guayaquil, que forman parte de la operación de recolección que realiza en horas de la tarde. La ubicación (latitud y longitud) de cada punto de picking se determinó con la herramienta Google Earth, la misma que ayudó al cálculo de la distancia entre cada par de clientes, input necesario para la aplicación de la Metaheurística desarrollada.

Se realizaron 15 corridas del algoritmo, considerando una flota con capacidad homogénea de 5.5 toneladas, que es el promedio de las capacidades de los vehículos. La mejor solución encontrada tiene un costo de 15.21 hrs, donde el tiempo total de espera es de 33:00 minutos (0.55 hrs), el tiempo total de atraso de 00:36 segundos (0.0099 hrs) que corresponde a un solo cliente, y utiliza en total 4 vehículos; los resultados se exponen en la tabla 3.

TABLA 3
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Rutas obtenidas en el caso de estudio

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

Se evidencia que en uno de los 65 clientes se incurre en un atraso (de apenas 0.0099 hrs) durante la recolección, lo que equivale al 1.5%; este resultado contrasta con el indicador real de servicio de la empresa, dato obtenido en el primer cuatrimestre del 2009, donde 12% de los clientes (de estas zonas) no son atendidos a tiempo.

Para demostrar la factibilidad de la implementación de un DSS para la planificación de rutas, usando la API Google Maps, se desarrolló una interfaz web que muestra el detalle de cada una de las rutas; una salida, a manera de ejemplo, se ilustra en la figura 5.

FIGURA 5
Diseño de un sistema de soporte de decisiones para resolver el problema de ruteo en un servicio de courier
Prototipo de una aplicación DSS para planificación de rutas



Tal como se expone en la figura 5, 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.

9. CONCLUSIONES Y RECOMENDACIONES

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.

Según los resultados obtenidos, se concluye que para ampliar el espacio solución es necesario realizar más de una corrida del algoritmo.

Este tipo de metaheurísticas evolutivas, usadas para resolver el CVRPSTW, no sólo deben reducir el costo de desplazamiento; sino además, y en igual grado de importancia, deberán minimizar el número de rutas; por otro lado, reducir el número de vehículos, puede provocar aumento de las infactibilidades de tiempo (esperas y atrasos).

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. Al optar por 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; lo anterior puede provocar dificultad en la implementación de este tipo de solución.

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 informáticas-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 planteen.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. OLIVERA, A. (2004). "*Heurísticas para Problemas de Ruteo de Vehículos*". Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.
- [2]. TOTH, P.; VIGO, D., "*The vehicle routing problem*", Universidad de Bologna, Italia.
- [3]. IANNONOU, G.; KRITIKOS, M.; PASTRACOS, G. (2002). "*A problem generator-solver heuristic for vehicle routing with soft time windows*". Universidad de Economía y negocios, Atenas – Grecia,
- [4]. ROMERO YÁÑEZ, G. (2008). "*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.
- [5]. DELGADO, E. (2007). "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.
- [6]. PINTO LÓPEZ, M. (2009). "*Teoría de la complejidad algorítmica*". Monografías.com. Revisado el 12 de septiembre. Disponible en <http://www.monografias.com/trabajos27/complejidad-algoritmica/complejidad-algoritmica.shtml>.
- [7]. ALBA, E.; DORRONSORO, B. (2008). "*Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms*", Departamento de Ciencias de la Computación de la Universidad de Málaga.
- [8]. ALBA, E.; DORRONSORO, B. (2008). "*Cellular Genetic Algorithms*". Operations research/ Computer Science Interfaces. Springer.
- [9]. ALBA TORRES, E. (1999). "*Análisis y diseño de algoritmos genéticos paralelos distribuidos*", Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga.
- [10]. BÄCK, T.; FOGEL, D.B.; MICHALEWICZ, Z. (1997). "*Handbook of Evolutionary Computation*, Oxford University Press".
- [11]. ÁSLAUG SÓLEY BJARNADÓTTIR, (2004). "*Solving the Vehicle Routing Problem with Genetic Algorithms*", Informatics and Mathematical Modelling - IMM, Technical University of Denmark, DTU.
- [12]. NATYHELEM GIL LONDOÑO, (2006). "*Algoritmos Genéticos*". Universidad Nacional de Colombia.
- [13]. THANGIAGH, SAM R. (1995) "*Vehicle Routing with Time Windows using Genetic Algorithms*". Application Handbook of Genetic Algorithms: New Frontiers, Volume II. Lance Chambers.
- [14]. KENNY QILI ZHU, (2000). "*A New Genetic Algorithm for VRPTW*", National University of Singapore.
- [15]. JOAQUÍN ANTONIO PACHECO BONROSTRO. "*Problemas de rutas con ventanas de tiempo*". Universidad Complutense de Madrid. Dpto. Estadística e Investigación operativa. Facultad de Ciencias Matemáticas.
- [16]. HOMBERGER, J.; GEHRING, H. (1998). "*Two evolutionary metaheuristics for the vehicle routing problem with time windows*". Fernuniversität hagen, lehrstuhl wirtschaftsinformatik, profustr. 8, D-58084 hagen, Bundesrepublik Deutschland.