

# Comparison of the Branch and Bound and Evolutionary Algorithms to Solve the Classic Vehicle Routing Problem Known as VRP

Marcos Moya Navarro, PhD  
Universidad Latina de Costa Rica, marcos.moya@ulatina.net

*Abstract– The routing model solution is an algorithmic challenge that seeks to optimize delivery routes to minimize the associated costs. Linear programming is a classical method proposed to find the minimum cost routes. Alternatively, heuristic and metaheuristic methodologies have been proposed to solve the routing problem.*

*The objective of this work is to compare both the average time to find a solution and the average total distance traveled obtained by using an evolutionary genetic algorithm and a linear programming model. Five route configurations with ten, twenty, thirty, forty and fifty customers or nodes to visit were studied.*

*Results pointed out that as the number of customers to visit is duplicated from ten to twenty, the average time to find the best solution is approximately ten times bigger when using a linear programming model. Moreover, when the number of customers doubles from twenty to forty, the average time to find the solution arises something about sixty-eight times the previous time.*

*Conversely, the evolutionary genetic algorithm showed that the average time response to find a solution decreased significantly as the number of customers included in the model increased. Regarding the total distance traveled on the route, the linear programming model found a solution 96.26% lower than that found with the evolutionary method for a route with ten customers to visit, and 48.48% lower on a route with 50 clients.*

*Recommendations based on results indicate that is convenient to find a solution through the linear programming model when fifty or less clients are included in the model.*

*On the contrary, for more than fifty clients, it is more convenient to solve the vehicle routing problem by means of evolutionary genetic algorithms that needs significantly less time to find a problem solution and they also find solutions closer and closer than those found by the linear programming model as the numbers of customers increase.*

*Keywords—Traveling Agent, Genetic Algorithm, Heuristic, Metaheuristic, Optimization*

**Digital Object Identifier:** (only for full papers, inserted by LACCEI).  
**ISSN, ISBN:** (to be inserted by LACCEI).  
**DO NOT REMOVE**

# Comparación de los Algoritmos de Ramificación y Acotamiento y Evolucionario Para Resolver el Problema Clásico de Ruteo De Vehículos Conocido Como VRP

Comparison of the Branch and Bound and Evolutionary Algorithms to Solve the Classic Vehicle Routing Problem Known as VRP

Marcos Moya Navarro, PhD  
Universidad Latina de Costa Rica, marcos.moya@ulatina.net

**Resumen**— La solución del modelo de enrutamiento es un desafío algorítmico que busca optimizar las rutas de entrega para minimizar los costos asociados. La programación lineal es un método clásico propuesto para encontrar las rutas de costo mínimo. Alternativamente, se han propuesto metodologías heurísticas y metaheurísticas para resolver el problema de enrutamiento.

El objetivo de este trabajo es comparar tanto el tiempo medio para encontrar una solución como la distancia total media recorrida obtenida mediante el uso de un algoritmo genético evolutivo y un modelo de programación lineal. Se estudiaron cinco configuraciones de rutas con diez, veinte, treinta, cuarenta y cincuenta clientes o nodos a visitar.

Los resultados señalaron que a medida que se duplica el número de clientes a visitar de diez a veinte, el tiempo promedio para encontrar la mejor solución es aproximadamente diez veces mayor cuando se utiliza un modelo de programación lineal. Además, cuando el número de clientes se duplica de veinte a cuarenta, el tiempo medio para encontrar la solución aumenta sesenta y ocho veces el tiempo anterior. Por el contrario, el algoritmo genético evolutivo mostró que el tiempo promedio de respuesta para encontrar una solución disminuyó significativamente a medida que aumentaba el número de clientes incluidos en el modelo. Con respecto a la distancia total recorrida en la ruta, el modelo de programación lineal encontró una solución 96,26% menor que la encontrada con el método evolutivo para una ruta con diez clientes, y 48,48% menor en una ruta con 50 clientes.

Las recomendaciones basadas en los resultados encontrados indican que es conveniente encontrar una solución a través del modelo de programación lineal cuando se incluyen cincuenta o menos clientes en el modelo.

Por el contrario, para más de cincuenta clientes es conveniente resolver el problema de enrutamiento mediante algoritmos genéticos evolutivos que necesitan significativamente menos tiempo para encontrar una solución al problema y también encuentran soluciones cada vez más cercanas que las encontradas por el modelo de programación lineal a medida que aumenta el número de clientes.

**Palabras Clave**— Agente Viajero, Algoritmo Genético, Heurístico, Metaheurístico, Optimización

**Abstract**— The routing model solution is an algorithmic challenge that seeks to optimize delivery routes to minimize the associated costs. Linear programming is a classical method proposed to find the minimum cost routes. Alternatively, heuristic and metaheuristic methodologies have been proposed to solve the routing problem.

The objective of this work is to compare both the average time to find a solution and the average total distance traveled obtained by using an evolutionary genetic algorithm and a linear programming model. Five route configurations with ten, twenty, thirty, forty and fifty customers or nodes to visit were studied.

Results pointed out that as the number of customers to visit is duplicated from ten to twenty, the average time to find the best solution is approximately ten times bigger when using a linear programming model. Moreover, when the number of customers doubles from twenty to forty, the average time to find the solution arises something about sixty-eight times the previous time.

Conversely, the evolutionary genetic algorithm showed that the average time response to find a solution decreased significantly as the number of customers included in the model increased. Regarding the total distance traveled on the route, the linear programming model found a solution 96.26% lower than that found with the evolutionary method for a route with ten customers to visit, and 48.48% lower on a route with 50 clients.

Recommendations based on results indicate that is convenient to find a solution through the linear programming model when fifty or less clients are included in the model.

On the contrary, for more than fifty clients, it is more convenient to solve the vehicle routing problem by means of evolutionary genetic algorithms that needs significantly less time to find a problem solution and they also find solutions closer and closer than those found by the linear programming model as the numbers of customers increase.

**Keywords**—Traveling Agent, Genetic Algorithm, Heuristic, Metaheuristic, Optimization

## I. INTRODUCCION

El problema de enrutamiento de vehículos es una función clave en la logística comercial. Tiene el objetivo de encontrar la mejor ruta para llegar a un destino específico. La Ref. [1] indica que en el ámbito de competencia que caracteriza al siglo XXI, la logística industrial es usada por las compañías

**Digital Object Identifier:** (only for full papers, inserted by LACCEI).  
**ISSN, ISBN:** (to be inserted by LACCEI).  
**DO NOT REMOVE**

con el fin de generar ventajas competitivas, por lo que son de vital importancia los procesos de aprovisionamiento y/o distribución, en el cual el establecimiento de las rutas para vehículos de manera óptima ha generado un gran interés de investigación.

La Ref. [2] establece que el ruteo de vehículos con modelado matemático es un problema difícil de implementar en las empresas debido a la gran cantidad de variables y restricciones que se generan en la solución del problema, razón por la cual, desde un punto de vista práctico, una importante cantidad de las empresas no lo utilizan.

La Ref. [3] indica que L. Fisher y R. Jaikumar presentaron una heurística para obtener una asignación de clientes a vehículos resolviendo un problema de asignación generalizada con una función objetivo que aproxima el costo de entrega, superando a las mejores heurísticas existentes en una muestra de problemas de prueba estándar, siempre encontrando una solución factible si existe, algo que ninguna otra heurística existente puede garantizar.

La Ref. [4] proporciona un algoritmo de ramificación y acotamiento para abordar un nuevo problema de enrutamiento de un solo vehículo que tiene en cuenta el peso del vehículo con y sin carga. Los autores indican que las autopistas en China utilizan el esquema de peaje por peso, en el que los peajes de las autopistas se cobran en función del peso y la distancia recorrida del vehículo, pero que la mayoría de los modelos de generación de rutas para vehículos asumen que el costo de viaje es equivalente a la distancia del viaje o bien a alguna constante; y que, como resultado, dichos modelos no se pueden aplicar en la práctica al sistema de transporte por autopista chino.

La Ref. [5] presenta un problema de enrutamiento de vehículos capacitados (CVRP) con demandas estocásticas difusas, en donde se utilizaron variables aleatorias difusas discretas para representar las demandas de los clientes. Los autores indican que el objetivo de CVRP con demandas estocásticas difusas es obtener un conjunto de rutas que se originen y terminen en el nodo de origen y mientras atraviesan la ruta, se satisfacen las demandas de todos los clientes presentes en la red. Para la solución del problema se utilizó el algoritmo de ramificación y acotamiento en un ejemplo numérico con cuatro clientes para presentar la metodología propuesta.

La Ref. [6] indica que, a través del estudio del problema de asignación y optimización de una flota de transporte, se han desarrollado diferentes algoritmos, con metodologías que se relacionan con la investigación de operaciones y la programación lineal entera. Los autores establecen que los modelos exactos, intentan descartar familias enteras de posibles soluciones, tratando así de acelerar la búsqueda y llegar a la óptima. Indican además que el método de

ramificación y acotamiento, propuesto por Land y Doig en 1960, cuando es aplicado para el problema del agente viajero siempre encuentra la solución óptima, por lo que podría ser necesario enumerar y comparar todas las posibles soluciones. De ahí la necesidad de utilizar heurísticas durante el proceso.

La Ref. [7] indica que en el método de ramificación y acotamiento el conjunto de soluciones que se va obteniendo usualmente se dispone en forma de árbol, donde cada nodo representa un problema de programación lineal, poseyendo cada hijo una restricción más que su predecesor o padre. Esta restricción lo que pretende es forzar a que una de las variables obtenidas en la resolución del problema padre sea menor o igual que la parte entera de la solución óptima obtenida para dicha variable, formándose así dos nuevos problemas objeto de estudio. Los autores indican que el procedimiento se dará por finalizado en el momento que ya no queden más elementos objeto de estudio en la lista de problemas a resolver.

La Ref. [8] establece que el problema de ruteo de vehículos con productos de desecho es uno de los problemas a los que se enfrentan las grandes ciudades, por lo que la optimización de las rutas de los vehículos es necesaria para minimizar la distancia y el tiempo de viaje. Los autores indican que investigaron dos modelos de optimización robustos, que están limitados por capacidad del vehículo y ventanas de tiempo. Los dos modelos de optimización robustos se resolvieron utilizando el enfoque exacto y la heurística. Indican además que el enfoque exacto se resolvió usando el método de ramificación y acotamiento con LINGO, mientras que el enfoque heurístico se resolvió con el algoritmo del vecino más cercano.

La Ref. [9] considera el problema de ruteo de vehículos para varios agentes entre los objetos de dos tipos. Se propone el análisis comparativo del algoritmo exacto para la resolución de este problema y los algoritmos heurísticos. El algoritmo exacto se basó en el método de ramificación y acotamiento. Alternativamente, el algoritmo de colonia de hormigas y el algoritmo genético fueron elegidos como aproximados. Se analizó el tiempo de ejecución de los algoritmos y se compararon los resultados en términos de proximidad a la solución óptima.

La Ref. [10] presenta un estudio comparativo de la capacidad de las formulaciones de programación entera mixta (MILP). Los autores indican que, al seleccionar la mejor formulación de programación lineal entera mixta, la cuestión importante es determinar cómo evaluar el rendimiento de cada formulación candidata en términos de criterios seleccionados. Los autores consideraron el problema de "seleccionar la formulación MILP más apropiada para un determinado tipo de tomador de decisiones" como un problema de toma de decisiones multicriterio y presentaron una metodología

integrada de toma de decisiones denominada AHP-TOPSIS para seleccionar la formulación más adecuada.

La Ref. [11] compara los algoritmos genéticos, Random (R), Nearest Neighbor (NN) (Shi, Zhao y Gong, 2009) y Greedy Randomized Adaptive Search Procedure (GRASP) (Vural, 2007 para estudiar el problema de generación de rutas para vehículos con entrega y recogida simultáneas (VRP/SDP), y el problema de generación de rutas para vehículos con entrega y recogida simultáneas con ventanas de tiempo (VRP/SDP/TW). El autor indica los tres modelos genéticos se comparan entre sí utilizando un conjunto de datos hipotéticos para una variedad de tamaños de problemas. Los resultados mostraron que el modelo GA/GRASP generalmente supera a los otros modelos para el problema de VRP/SDP. Sin embargo, los tres modelos muestran el mismo rendimiento para el problema VRP/SDP/TW.

La Ref. [12] indica que los algoritmos metaheurísticos se seleccionan para resolver el problema de enrutamiento de vehículos, donde el algoritmo genético (GA) se implementa como el algoritmo metaheurístico principal. Los autores indican que los algoritmos genéticos pertenecen a la familia de algoritmos evolutivos (EA), que funciona con un mecanismo de "supervivencia del más apto", por lo que presentan la idea de implementar diferentes operadores genéticos, modificados para su uso con el VRP, y realizan experimentos para determinar la mejor combinación de operadores genéticos para resolver el VRP y encontrar soluciones óptimas para ejemplos de la vida real a gran escala del VRP.

La Ref. [13] indica que el problema de enrutamiento de vehículos múltiples con ventana de tiempo (MVRTW) es una variante de VRP, que se adapta a las especificaciones realistas del sistema, como la capacidad de múltiples vehículos, la restricción de tiempo y la restricción de la red (unidireccional, prohibición de movimiento de giro, etc.). Para resolver el MVRTW los autores proponen un enfoque mejorado que combina un sistema de información geográfica (GIS) con un algoritmo genético paralelo (PGA). Para fortalecer la capacidad de búsqueda, en PGA se utilizó un mecanismo de generación adaptativo de la población inicial y los operadores evolutivos. El enfoque sugerido demostró ser eficiente por un caso práctico de Changchun.

La Ref. [14] indica que se aborda un problema de enrutamiento eléctrico (e-VRP) de entrega de última milla operado por vehículos eléctricos innovadores. Los vehículos estudiados son modulares, consisten en un sistema con una cabina para el conductor y uno o más módulos para la mercancía. Los autores comparan técnicas de búsqueda local y su combinación con el esquema evolutivo. Se realizaron pruebas experimentales de este estudio para demostrar la relevancia del método de descendencia de vecindad variable evolutiva.

La Ref. [15] indica que los algoritmos de estimación de distribución (EDA) son algoritmos evolutivos que utilizan modelos probabilísticos para conducir una búsqueda más eficiente de soluciones óptimas. Los autores investigaron la aplicación de los EDA a una versión del problema de enrutamiento de vehículos en el que las soluciones deben satisfacer una serie de restricciones que involucran a los clientes, la flota de vehículos y los artículos a entregar. Se compararon dos representaciones diferentes de las soluciones y aplicaron EDA que usan tres modelos probabilísticos con diferentes características. Los resultados mostraron que la combinación de una representación entera con un modelo probabilístico basado en árboles produce los mejores resultados y es capaz de resolver problemas de enrutamiento de vehículos que contienen más de miles de rutas prometedoras.

La Ref. [16] propone un nuevo algoritmo metaheurístico para resolver el problema de ruteo de vehículos VRP con el fin de lograr una solución óptima. El algoritmo propuesto es un algoritmo metaheurístico de optimización de comparación parcial (PCO). Para demostrar que PCO es una buena metaheurística para resolver VRP, se seleccionaron varias instancias de VRP simétrico de la biblioteca de VRP para evaluar su rendimiento. Los resultados numéricos obtenidos del cálculo indicaron que el método de optimización propuesto podría alcanzar resultados casi similares a las soluciones más conocidas dentro de un tiempo de cálculo razonable. Mostró que PCO era una buena metaheurística para resolver VRP.

La Ref. [17] establece que las técnicas de búsqueda local se han aplicado para resolver muchos problemas combinatorios. La calidad de la búsqueda local utilizada en el algoritmo memético puede afectar significativamente el rendimiento del algoritmo. Los autores consideran cuatro búsquedas locales diferentes y las utilizan para resolver el problema de enrutamiento de vehículos con backhauls o viajes de vuelta (VRPB) como parte de un algoritmo memético propuesto. Extensas pruebas muestran que el método de la mejora de la adyacencia propuesto superó a los otros métodos considerados en términos de tiempo computacional y la calidad de las soluciones obtenidas.

#### *A. Solución del Ruteo de Vehículos Por Medio de Programación Lineal Entera Binaria*

El objetivo de este trabajo es comparar tanto el tiempo medio transcurrido para encontrar una solución óptima como la distancia total media recorrida por un solo vehículo para visitar todos los clientes en el clúster. Se asume que el vehículo tiene capacidad para transportar la mercancía de los clientes que va a visitar. Para el proceso de formar los clústeres en función de la capacidad de los camiones de la flota véase la Ref. [3]. En la Fase I se resuelve el problema

de ruteo mediante un modelo de programación lineal entera binaria usando el algoritmo de ramificación y acotamiento para encontrar la solución. Se estudiaron cinco tamaños de clústeres, conformados por diez, veinte, treinta, cuarenta y cincuenta clientes o nodos por visitar. Se utilizó el modelo de programación entera binaria, propuesto por Miller, Tucker y Zemlin [18].

La Ref. [19] indica que para la resolución de la forma básica de un modelo de ruteo de vehículos (VRP) se requiere una matriz de distancias, un número de vehículos y un depósito. Indica el autor que la matriz de distancias es un tabulado que registra la longitud entre cada uno de los nodos del modelo, incluido el nodo depósito. La tabla I resume el número de restricciones y el número de variables incluidos en el modelo de programación entera binaria formulado para cada uno de los cinco tamaños de clústeres incluidos en este trabajo. Para el clúster cuya configuración se compone de un depósito y diecinueve clientes o ubicaciones por visitar, la formulación del modelo de programación entera binaria requerirá de una matriz de distancias con 400 datos.

TABLA I  
NUMERO DE RESTRICCIONES Y VARIABLES  
SEGÚN LA CANTIDAD DE CLIENTES O NODOS A VISITAR

Tamaño del Cluster (Nodos)	Número de Restricciones	Número de Variables
10	101	109
20	401	419
30	901	929
40	1601	1639
50	2501	2549

### B. Metodología de Solución del Algoritmo de Ramificación y Acotamiento

La Ref. [20] indica que el algoritmo Ramificación y Acotamiento (B&B) es una metodología que realiza una búsqueda completa de una solución óptima dentro de todo el espacio de búsqueda, pero con la ventaja de ir reduciéndolo a medida que encuentra puntos de infactibilidad (sondaje) y puntos para los cuales se determina una buena solución de la función objetivo. Los autores establecen que el algoritmo B&B se basa en dos estrategias: dividir y explorar, por lo que el óptimo global es la mejor solución de las encontradas en cada uno de los subespacios. La Fig. 1 resume la metodología propuesta por los autores para la solución del algoritmo de ramificación y acotamiento.

### C. Metodología de Solución del Algoritmo Genético Evolucionario

La Ref. [21] indica que un algoritmo genético está conformado por una población de cromosomas, típicamente 100, 1000 o más. Cada cromosoma representa una posible

solución buena o mala al problema. Al enfrentar cada cromosoma al problema se obtiene una aptitud que indica lo bueno o malo que es. Después se hace una selección de unos cuantos cromosomas que son los que pasarán a la pila de apareamiento para reproducirse. Los cromosomas hijos resultantes se inyectan a la población. Y todo ello se repite. Cada repetición constituye una generación. La Fig. 2 muestra el diagrama de flujo de los datos empleado por el algoritmo evolucionario o evolutivo.

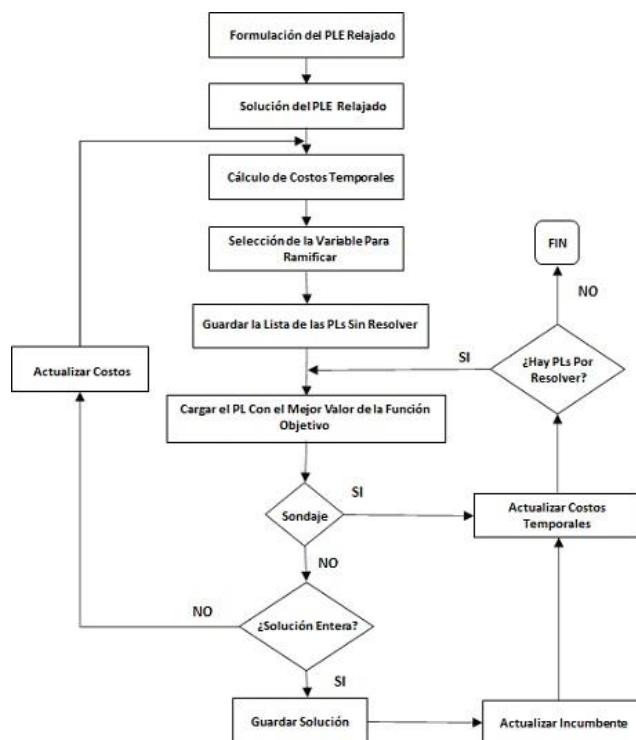


Fig. 1 Metodología de Ramificación y Acotamiento Para Resolver el Problema de Ruteo de Vehículos VRP.

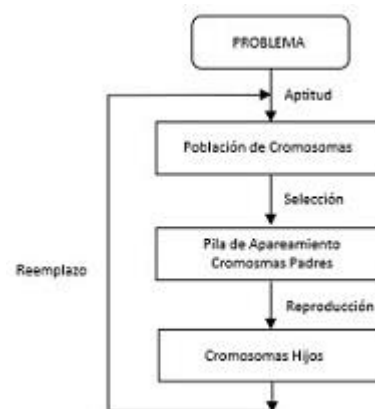


Fig. 2 Metodología de Solución Para el Algoritmo Evolutivo Para Resolver el Problema de Ruteo de Vehículos VRP.

### D. Metodología Para el Cálculo de la Matriz de Distancias

Para la solución del problema de ruteo de vehículos por los métodos de ramificación y acotamiento y evolucionario se requirió generar una matriz de distancias para cada uno de los tamaños de problema estudiados. Se utilizó el algoritmo de Haversine [22] para calcular la matriz de distancias. Este algoritmo, además de las coordenadas geográficas entre dos puntos requiere el radio ecuatorial R de la tierra el cual es de 6378 kilómetros, pues toma en cuenta la curvatura de la tierra. Se utilizó el generador de la base de datos del modelo de ruteo, denominado GenRestClust desarrollado por el autor, el cual construye la función objetivo en términos de la distancia recorrida y las restricciones funcionales del modelo de programación lineal propuesto por Fisher, Tucker y Zemlin [18]. El generador de modelos se alimenta de las coordenadas geográficas de latitud y longitud de cada uno de los clientes a visitar para el cálculo de la matriz de distancias. Está codificado en el lenguaje de programación Visual Basic incorporado en la plataforma de Microsoft Office Excel. La Fig. 3 muestra el procedimiento de Haversine para el cálculo de las distancias entre clientes o nodos del clúster estudiado

Paso #1. Calcular el delta de latitud  $\Delta Lat$ .

$$\Delta Lat = Lat2 - Lat1$$

Lat1 y Lat2 son las coordenadas geográficas decimales de latitud entre una localización y otra.

Paso #2. Calcular el delta de longitud  $\Delta Long$ .

$$\Delta Long = Long2 - Long1$$

Long1 y Long2 son las coordenadas geográficas decimales de longitud entre una localización y otra.

Paso #3. Calcular la distancia entre las dos posiciones geográficas estudiadas.

$$d = R \cdot C$$

donde:

$$R = \text{radio de la Tierra}$$

$$A = \sin^2(\Delta Lat/2) + \cos(Lat1) \cdot \cos(Lat2) \cdot \sin^2(\Delta Long/2)$$

$$C = 2 \cdot \text{atan2}(\sqrt{A}, \sqrt{1-A})$$

Fig. 3 Procedimiento de Haversine Para el Cálculo de la Matriz de Distancias.

La fórmula haversine es una forma muy precisa de calcular distancias entre dos puntos en la superficie de una esfera utilizando la latitud y la longitud de los dos puntos. Además, la formulación en términos de haversines es más útil para distancias y ángulos pequeños [24].

### II. DEFINICION DEL CASO DE ESTUDIO

Se seleccionaron cinco escenarios de ruteo como caso de estudio para comparar la solución obtenida mediante la aplicación del modelo de programación lineal propuesto por Fisher, Tucker y Zemlin [3], con las solución encontrada utilizando el algoritmo evolucionario implementado en la aplicación Solver de Microsoft Office Excel. Se comparó la distancia total recorrida en la ruta y el tiempo transcurrido en segundos para resolver el problema. En cada escenario estudiado se determinó el orden de visita de los clientes para minimizar la distancia total recorrida por un solo vehículo. Los datos recolectados incluyeron el orden de la ruta, la distancia total recorrida por el vehículo y el tiempo necesario para encontrar la solución del problema con tamaños de clústeres de 10, 20, 30, 40 y 50 clientes o nodos por visitar. El trabajo de campo para determinar las coordenadas geográficas de latitud y longitud de cada uno de los clientes por visitar, así como del depósito de donde parte y llega el vehículo de transporte fue realizado por la organización GS1 Costa Rica [23]. La tabla II presenta las coordenadas geográficas utilizadas para el cálculo de la matriz de distancias entre clientes para un clúster de tamaño 20.

TABLA II  
COORDENADAS GEOGRÁFICAS UTILIZADAS PARA EL CÁLCULO DE LA MATRIZ DE DISTANCIAS EN UN CLÚSTER DE TAMAÑO 20

Punto de Venta	Latitud	Longitud
<b>Depósito</b>	<b>9.97287</b>	<b>-84.085458</b>
Punto de Venta 1	10.016752	-84.215312
Punto de Venta 2	9.980268	-84.178228
Punto de Venta 3	9.870415	-83.93045
Punto de Venta 4	9.973569	-84.009765
Punto de Venta 5	9.899762	-84.068011
Punto de Venta 6	9.996509	-84.118927
Punto de Venta 7	9.944601	-84.034377
Punto de Venta 8	9.951305	-84.049239
Punto de Venta 9	9.93752	-84.187322
Punto de Venta 10	9.958598	-84.076929
Punto de Venta 11	9.851545	-83.917854
Punto de Venta 12	10.012802	-84.221878
Punto de Venta 13	9.885642	-84.065961
Punto de Venta 14	10.078026	-84.316342
Punto de Venta 15	10.005181	-84.134611
Punto de Venta 16	9.910267	-84.056156
Punto de Venta 17	9.979999	-84.199559
Punto de Venta 18	10.095322	-84.469049
Punto de Venta 19	9.963311	-84.086889

El clúster de tamaño 20 incluye visitar a 19 clientes más el depósito, de donde partió y arribó el vehículo de transporte. Se omiten los nombres de los clientes por razones de confidencialidad. De manera similar, se incluyeron las coordenadas geográficas para los tamaños de clústeres con 10, 30, 40 y 50 nodos de visita.

La tabla III muestra un resumen de la matriz de distancias que se obtuvo para un clúster de tamaño 20 utilizando el algoritmo de Haversine. De manera análoga se obtuvieron las matrices de distancia para los clústeres de tamaños 10, 30, 40 y 50 nodos.

TABLA III  
MATRIZ DE DISTANCIAS PARA EL CLÚSTER DE TAMAÑO 20 MEDIANTE EL ALGORITMO DE HAVERSINE

	Empresa	PV1	...	PV18	PV18
Empresa	0	23	...	67.55	1.64
PV1	23	0	...	44.56	23.36
PV2	15.6	8.78	...	52.5	15.57
PV3	31.28	53.84	...	98.02	30.61
...	...	...	...	...	...
PV17	19.16	6.79	...	49.22	19.09
PV18	67.55	44.56	...	0	67.85
PV19	1.64	23.36	...	67.85	0

### III. METODOLOGÍA PARA OBTENER LA SOLUCIÓN DEL PROBLEMA MEDIANTE LOS ALGORITMOS DE RAMIFICACIÓN Y ACOTAMIENTO L Y EVOLUCIONARIO

La Fig. 4 muestra el procedimiento seguido para resolver el problema de rutas mediante el modelo de programación lineal propuesto por Fisher y otros. El primer paso del procedimiento consistió en generar la base de datos del modelo mediante el generador de modelos GenRestClust, el cual se alimenta de la matriz de distancias. El segundo paso fue determinar la solución del modelo. Se utilizó el programa de optimización OR Brainware Decision Tools para encontrar las rutas de mínima distancia recorrida. Las rutas debieron cumplir la restricción de formar un ciclo hamiltoniano, que indica que el vehículo parte del depósito, visita a los clientes una única vez, y regresa al depósito. Las variables legítimas fueron declaradas enteras binarias. El método de solución seleccionado fue el de ramificación y acotamiento. En el tercer paso del procedimiento se generó el informe de respuestas con la solución del problema. En este paso se colecta la ruta a seguir, la distancia total recorrida y el tiempo transcurrido para obtener la solución. Este procedimiento se repitió para los cinco tamaños de clúster estudiados.

Para resolver el problema de ruteo por el método evolucionario se utilizó la plataforma de optimización

incorporada en el Solver de Microsoft Office Excel. La Fig.5 resume el procedimiento seguido.

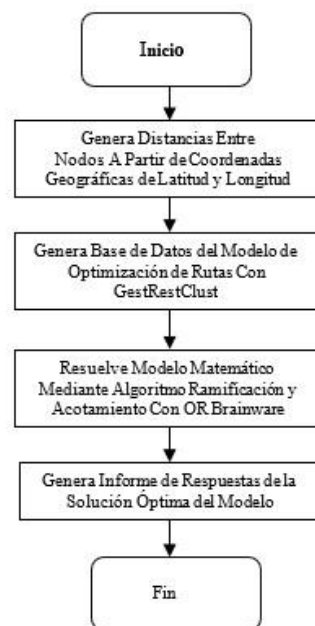


Fig. 4 Procedimiento de Obtención de Rutas Mediante el Modelo de Programación Lineal Entera Binaria.

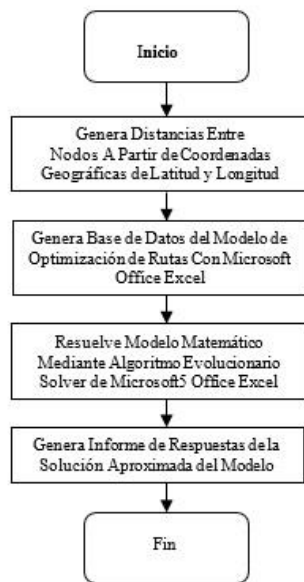


Fig. 5 Procedimiento de Obtención de Rutas Mediante el Modelo Evolucionario .

#### IV. RESULTADOS OBTENIDOS MEDIANTE LOS MODELOS DE PROGRAMACIÓN LINEAL ENTERA BINARIA Y EVOLUCIONARIO

Los procedimientos establecidos en la Fig. 4 y Fig. 5 se replicaron seis veces para cada uno de los cinco tamaños de clúster establecidos, por lo que se corrió un diseño experimental de 30 corridas. Con base en las seis réplicas por tamaño de clúster se determinó el tamaño de muestra requerido.

La tabla IV indica que al 95% de confianza hay evidencia suficiente para creer que 6 réplicas es un tamaño de muestra suficiente para estimar el tiempo promedio que tarda el modelo de programación lineal en encontrar la solución con un error máximo de 10%.

TABLA IV  
TAMAÑO DE MUESTRA REQUERIDO POR TAMAÑO DEL CLÚSTER AL 95% DE CONFIANZA

Número de Nodos	Tamaño de Muestra
10	3
20	3
30	6
40	4
50	6

##### 1. Resultados Obtenidos de la Solución del Ruteo de Vehículos Por Medio del Modelo de Programación Lineal

La tabla V muestra la secuencia óptima de visita de a los clientes encontrada en las seis réplicas ejecutadas cuando el tamaño del clúster fue de 20 nodos o clientes por visitar. Cabe destacar que para las seis replicas se obtuvo la misma solución óptima.

TABLA V  
SECUENCIA ÓPTIMA OBTENIDA POR EL ALGORITMO DE RAMIFICACIÓN Y ACOTAMIENTO PARA UN CLÚSTER DE 20 NODOS

Solución Obtenida
1-2-13-15-19-18-10-3-16-7-20-11-17-6-14-12-4-5-8-9-1

La tabla VI muestra que la distancia total recorrida en kilómetros no cambia en ninguna de las réplicas. Además, el tiempo total requerido en segundos para encontrar la solución óptima presenta poca variabilidad.

La tabla VII muestra los tiempos promedio requeridos por el modelo de programación lineal para encontrar la solución óptima, según el tamaño del clúster. La Fig. 6 muestra el crecimiento exponencial del tiempo para encontrar la solución óptima a medida que crece el número de nodos en el clúster de ruteo. También proporciona una ecuación de predicción para

estimar el tiempo promedio para encontrar una solución óptima en función del número de nodos en el clúster. Obsérvese que el coeficiente de determinación de la ecuación de predicción se estima en 0.9953.

TABLA VI  
DISTANCIA TOTAL RECORRIDA Y TIEMPO PARA ENCONTRAR LA SOLUCIÓN ÓPTIMA OBTENIDA POR EL MODELO DE PROGRAMACIÓN LINEAL PARA UN CLÚSTER DE 20 NODOS

Réplica	Distancia Recorrida	Tiempo Para Encontrar la Solución
1	236.27	25.91
2	236.27	25.73
3	236.27	26.67
4	236.27	26.14
5	236.27	26.34
6	237.27	25.95

TABLA VII  
TIEMPO PROMEDIO EN SEGUNDOS REQUERIDO POR TAMAÑO DE CLÚSTER

Número de Nodos	Tiempo Promedio
10 Nodos	2.403
20 Nodos	26.125
30 Nodos	288.911
40 Nodos	1788.188
50 Nodos	3777.840

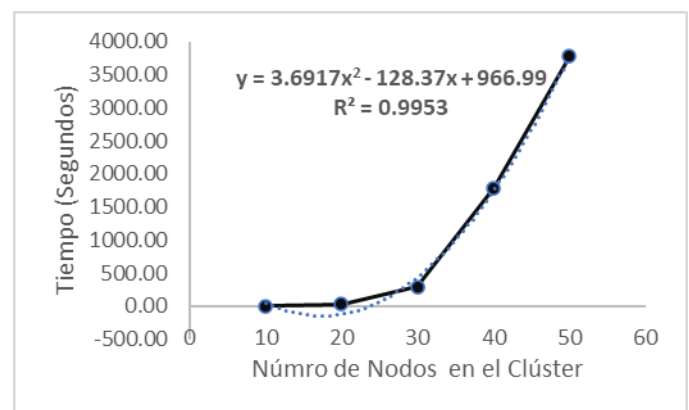


Fig. 6 Tiempo Requerido Por el Modelo de Programación Lineal Para Encontrar la Solución Óptima.



La tabla VIII muestra las distancias recorridas, en kilómetros, para cada uno de los tamaños del clúster. Los datos de esta tabla se utilizan en este trabajo para determinar la cercanía a la solución óptima que producen las soluciones obtenidas por el método evolucionario.

TABLA VIII  
DISTANCIA PROMEDIO RECORRIDA EN FUNCIÓN DEL TAMAÑO DEL CLÚSTER

Número de Nodos	Distancia Promedio
10	<b>134.85</b>
20	<b>236.27</b>
30	<b>244.66</b>
40	<b>266.10</b>
50	<b>277.60</b>

El número de variables y el número de restricciones se calculan respectivamente con las ecuaciones (1) y (2).

$$\text{Número de Nodos}^2 + (\text{Número de Nodos} - 1) \quad (1)$$

$$(\text{Número de Nodos}^2 + 1) \quad (2)$$

## 2. Resultados Obtenidos de la Solución del Ruteo de Vehículos Por Medio del Modelo Evolucionario

La tabla IX muestra el tiempo promedio requerido para encontrar una solución factible utilizando el algoritmo evolucionario incorporado en el Microsoft Office Excel 365. El algoritmo se parametrizó para que corriera en un tiempo máximo en segundos igual al máximo tiempo que tardó el modelo de programación lineal para encontrar la solución, y un tiempo máximo sin mejora de dos minutos ejecutando el modelo en un procesador Intel(R) Core(TM) i7 con un sistema operativo de 64 bits. La Fig. 7 muestra que el algoritmo evolutivo tarda significativamente menos tiempo en encontrar una solución factible a medida que crece el número de nodos en la ruta.

TABLA IX  
TIEMPO PROMEDIO PARA ENCONTRAR UNA SOLUCIÓN FACTIBLE MEDIANTE EL MODELO EVOLUCIONARIO

Número de Nodos	Tiempo Promedio
<b>10 Nodos</b>	6.75
<b>20 Nodos</b>	34.40
<b>30 Nodos</b>	391.74
<b>40 Nodos</b>	543.58
<b>50 Nodos</b>	773.68

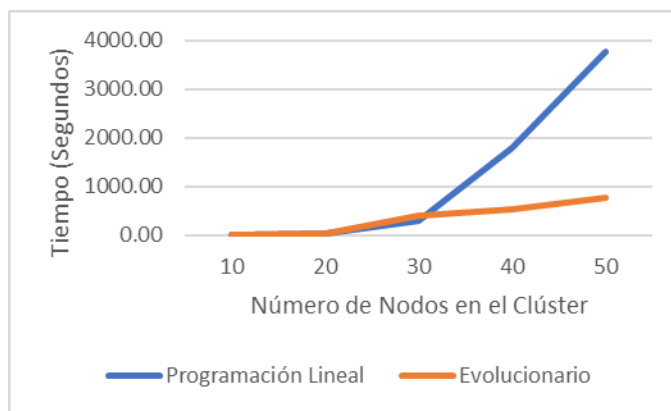


Fig. 7 Comparación entre el tiempo requerido por el modelo de programación lineal para encontrar la solución óptima y el requerido por el modelo evolucionario.

Con respecto a la distancia total recorrida en la ruta, la tabla X indica que el modelo de programación lineal encontró una solución 96,26% menor que la encontrada con el modelo de ruteo evolucionario para una ruta con diez clientes, y 48,48% menor en una ruta con 50 clientes.

TABLA X  
RAZÓN ENTRE LA DISTANCIA PROMEDIO OBTENIDA POR EL MODELO DE PROGRAMACIÓN LINEAL Y EL MODELO EVOLUCIONARIO

Número de Nodos	Distancia Promedio PL	Distancia Promedio E	Razón PL/E
<b>10 Nodos</b>	134.85	140.0900	0.9626
<b>20 Nodos</b>	236.27	253.0861	0.9336
<b>30 Nodos</b>	244.66	364.3346	0.6715
<b>40 Nodos</b>	266.10	439.3716	0.6056
<b>50 Nodos</b>	277.60	572.6080	0.4848

TABLA XI  
RAZÓN ENTRE EL TIEMPO PROMEDIO OBTENIDO POR EL ALGORITMO DE RAMIFICACIÓN Y ACOTAMIENTO Y EL ALGORITMO EVOLUCIONARIO

Número de Nodos	Tiempo Promedio	Tiempo Promedio	Razón
10 Nodos	<b>2.40</b>	<b>6.7461</b>	<b>2.8078</b>
20 Nodos	<b>26.13</b>	<b>34.3996</b>	<b>1.3167</b>
30 Nodos	<b>288.91</b>	<b>391.7441</b>	<b>1.3559</b>
40 Nodos	<b>1788.19</b>	<b>543.5777</b>	<b>0.3040</b>
50 Nodos	<b>3777.84</b>	<b>773.6844</b>	<b>0.2048</b>

La tabla XI indica que el modelo evolucionario requirió un tiempo 79.52% menor que el tiempo requerido por el modelo de programación entera binaria para encontrar una solución óptima para un tamaño de problema de 50 nodos en la ruta de distribución.

### 3. Conclusiones

1. Para un número de clientes en el clúster de ruteo de 50 o menos parece más conveniente usar el modelo de programación lineal para encontrar una ruta de distancia mínima
2. Para un número de clientes en el clúster de ruteo de 50 o más es conveniente usar el modelo evolucionario que tarda menos tiempo en encontrar una buena solución factible que cumple los requisitos de un ciclo hamiltoniano.
3. El tiempo requerido por el computador para encontrar una solución óptima crece de manera exponencial cuando se utiliza el modelo de programación lineal con un clúster de ruteo de 50 o más clientes.
4. El número de restricciones y variables aumenta exponencialmente a medida que se aumenta el número de clientes a visitar en la ruta.

### AGRADECIMIENTO

Un agradecimiento a la organización sin fines de lucro GS1 Costa Rica por el apoyo para proporcionar los datos necesarios para realizar este estudio, en particular al Ing. José Ugalde.

### REFERENCES

[1] Rocha, L.; González, C. y Orjuela, J. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. Ingeniería, Vol. 16, No. 2, pp. 35 - 55 .  
<https://www.redalyc.org/articulo.oa?id=498850173004>

[2] Moya, M. (2017) "Optimización del Ruteo de Vehículos en una Cadena de Suministros Con Programación Matemática En Una Base Diaria", LACCEI Proceedings, 2017.  
[https://laccei.org/LACCEI2017-BocaRaton/full\\_papers/FP169.pdf](https://laccei.org/LACCEI2017-BocaRaton/full_papers/FP169.pdf)

[3] Fisher, M., & Jaikumar, R. (1981). A generalized assignment heuristic for the vehicle routing problem. Networks 11, 109 -124  
<https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230110205>

[4] Zhang,Z, Qin,H,Zhu,W,Lim,A, "The single vehicle routing problem with toll-by-weight scheme: A branch and bound approach", European Journal of Operational Research. Jul 2012, Vol. 220 Issue 2, p295-304. 10p.  
<https://web.s.ebscohost.com/ehost/detail/detail?vid=5&sid=ac1b69e5-63e5-4657-91f7-aa767cf75f8f%40redis&bdata=JkF1dGhUeXBIPXNzbyZsYW5nPWVzJnNpdGU9ZWVhc3QtbGl2ZQ%3d%3d#AN=73804374&db=iib>

[5] Singh, V, Sharma, K, Chakraborty, D, "A Branch-and-Bound-based solution method for solving vehicle routing problem with fuzzy

stochastic demands", Indian Academy of Sciences. 2021, Sādhanā (2021) 46:195.  
<https://link.springer.com/article/10.1007/s12046-021-01722-0>

[6] Benítez, A, Acosta, J, (2017), "Optimización Basada en Metaheurísticas: Una Aproximación a la Solución del Problema de Ruteo de Vehículos Con Ventanas Horarias", Repositorio Universidad de Rosario, Colombia.  
<https://repository.urosario.edu.co/server/api/core/bitstreams/c6a0a6fa-ee25-470e-9205-8919ead7c1f8/content>

[7] García, M, (2014)"Problema del Viajante de Comercio: "Métodos Exactos de Resolución", Repositorio de la Universidad de la Laguna, España.  
[https://riull.ull.es/xmlui/bitstream/handle/915/222/Problema+del+Viajante+de+Comercio+\(TSP\).+Metodos+exactos+de+resolucion..pdf;jsessionid=897BB40F94915DB2E611249E4CD04923?sequence=1](https://riull.ull.es/xmlui/bitstream/handle/915/222/Problema+del+Viajante+de+Comercio+(TSP).+Metodos+exactos+de+resolucion..pdf;jsessionid=897BB40F94915DB2E611249E4CD04923?sequence=1)

[8] Yuliza, E; Puspita, F M; Supadi, S S; Octarina, S. "Journal of Physics: Conference Series; Bristol", Tomo 1663, N.º 1, (Oct 2020).  
<https://www.proquest.com/pq1academic/docview/2571149196/507009A3EF07460EPQ/2?accountid=32236>

[9] Medvedev, S. (2020). "Comparative Analysis of the Exact and Heuristic Algorithms for Solving the Vehicle Routing Problem for Several Agents Among the Objects of Two Types", The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings; Piscataway, pp.817-821.  
<https://www.proquest.com/pq1academic/docview/2471919460/A7034EC18C184D57PQ/4?accountid=32236>

[10] Kececi, B, Tusan, D, Dinler, E, Yusuf, T, (2017), "A comparative study of the capability of alternative mixed integer programming formulations", Technological and Economic Development of Economy; Vilnius, pp. 561-584.

[11] Syed, I, (2011), "Vehicle Routing Problem in logistics: A Genetic Algorithm-based comparative study", MAI 49/05M, Masters Abstracts International", United States.  
<https://www.proquest.com/dissertations-theses/vehicle-routing-problem-logistics-genetic/docview/868328373/se-2?accountid=32236>

[12] Ochelska, J, Ponszewska, A, Marańda, W, (2021), "Selected Genetic Algorithms for Vehicle Routing Problem Solving", Electronics, Tomo 10, N.º 24, Switzerland, Basel.  
<https://www.proquest.com/scholarly-journals/selected-genetic-algorithms-vehicle-routing/docview/2612764969/se-2>

[13] Bai, Y, Zhou, X, Zhang, Y, Yang, M, (2015), "Study of multi-vehicle routing problem with time window", IET Conference Proceedings, United Kingdom.  
<https://www.proquest.com/conference-papers-proceedings/study-multi-vehicle-routing-problem-with-time/docview/1776480416/se-2?accountid=32236>

[14] Rezgui, D ; Bouziri, Hend., A, Wassila, S, Jouhaina C, (2020), "A Comparative Study of Local Search Techniques Addressing an Electric Vehicle Routing Problem with Time Windows", The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings, United States, pp. 1-5.

[15] Santana, R, Sirbiladze, G, Ghvaberidze, B, Matsaberidze, B, (2017), "A comparison of probabilistic-based optimization approaches for vehicle routing problems", The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings, United States, pp. 2606-2613.  
<https://www.proquest.com/conference-papers-proceedings/comparison-probabilistic-based-optimization/docview/1917531574/se-2?accountid=32236>

[16] Adhi, A, Santosa, B , Siswanto, N, (2019). "A new metaheuristics for solving vehicle routing problem: Partial Comparison Optimization ", IO P Conference Series. Materials Science and Engineering; Bristol Tomo 598, N.º 1, United Kingdom.  
<https://www.proquest.com/scholarly-journals/new-metaheuristics-solving-vehicle-routing/docview/2561211648/se-2>

[17] Saremi, A, ElMekkawy, T, Aboonabi, A, Wang, G, "(2008), "A Comparison of Four Different Local Search Methods for Solving Vehicle Routing Problem with Backhauls", IIE Annual Conference. Proceedings, United States, PP. 354-359.

- <https://www.proquest.com/scholarly-journals/comparison-four-different-local-search-methods/docview/192463204/se-2?accountid=32236>
- [18] Miller. C, Tucker. A, Zemplin. R, (1960) “Integer programming formulation of traveling salesman problems”, *Journal of the ACM*, Vol. 7, United States, pp. 326–329.  
<https://dl.acm.org/doi/10.1145/321043.321046#d6885162e1>
- [19] Salazar. B. “Como calcular una matriz de distancias para modelar un VRP”,(2021).  
<https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/como-calcular-una-matriz-de-distancias-para-modelar-un-vrp/>
- [20] Tapias. C, Galeano. A, Hincapié. R, (2011), “Planeación de sistemas secundarios de distribución usando el algoritmo de branch and bound”, *Ingeniería y Ciencia*. ene-jun2011, Vol. 7 Issue 13, pp. 47-64.  
<https://web.s.ebscohost.com/ehost/detail/detail?vid=4&sid=26849715-5ecf-4df0-8f04-0e538dacef70%40redis&bdata=JkF1dGhUeXBIPXNzbyZsYW5nPWVzJnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=74484983&db=asn>
- [21] García. A, “Algoritmos Evolutivos”. Universidad del Valle. Colombia,  
<https://libros.univalle.edu.co/index.php/programaeditorial/catalog/download/150/72/1359?inline=1>
- [22] Genbeta, (2011), “¿Cómo Calcular la Distancia Entre Dos Puntos Geográficos en C#? (Fórmula de Haversine).  
<https://www.genbeta.com/desarrollo/como-calcular-la-distancia-entre-dos-puntos-geograficos-en-c-formula-de-haversine>
- [23] GS1 Costa Rica. (2023), “Portafolio de Servicios”  
<https://www.gs1.org/services>
- [24] Kettle. Simon, (2017), “Distance on a sphere: The Haversine Formula”  
<https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128>