

Resolución del Problema de Enrutamiento de Vehículos con Capacidad Utilizando Algoritmos Genéticos.

Thalia Blanco Figueras

LIA.BLANCO98@GMAIL.COM

C-512

Ariel Plasencia Díaz

ARIELPLASENCIA00@GMAIL.COM

C-512

Tutor(es):

MsC. Fernando Raúl Rodríguez Flores, *Universidad de La Habana*

Resumen

Este artículo presenta un procedimiento alternativo para resolver el Problema de Enrutamiento de Vehículos con Limitaciones de Capacidad. Al ser un problema del tipo no polinomial (NP-Difícil) se propone el Método de Algoritmo Genético como metaheurística para solucionarlo.

A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde a los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin en 1859. Como constituye la supervivencia la mayor problemática a la que se enfrentan los seres vivos, cuentan con habilidades innatas provistas en su material genético; a nivel de genes, se buscan aquellas adaptaciones beneficiosas en un medio hostil y cambiante. La selección natural influye en que cada especie obtenga una cierta cantidad de “conocimiento”, el cual es incorporado a la información de sus cromosomas; así pues, la evolución tiene lugar en los cromosomas, en donde está codificada la información del ser vivo. Durante el proceso de formación de nuevos individuos, se combina la información cromosómica de sus progenitores (no se conoce aún la forma en que esta combinación es llevada a cabo), por lo que, en el transcurso de las generaciones, esta información almacenada sufre de un constante cambio. Basándose en este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real, siendo sumamente útiles, en los problemas de búsqueda y optimización.

Abstract

This article presents an alternative procedure to solve the Vehicle Routing Problem with Capacity Limitations. Being a non-polynomial problem (NP-Hard), a metaheuristic algorithm is proposed using the Genetic Algorithm Method.

Throughout the generations, populations evolve in nature according to the principles of natural selection and the survival of the fittest, postulated by Darwin in 1859. As survival is the greatest problem faced by living beings alive, they have innate abilities provided in their genetic material; At the gene level, beneficial adaptations are sought in a hostile and changing environment. Natural selection influences each species to obtain a certain amount of “knowledge”, the which is incorporated into the information of your chromosomes; thus, evolution takes place in the chromosomes, where the information of the living being is encoded. During the process of formation of new individuals, the chromosomal information of their parents is combined (the way in which this combination is carried out is not yet known), so that, over the course of generations, this stored information suffers from a constant change. Based on this process, Genetic Algorithms are capable of creating solutions for real-world problems, being extremely useful in search and optimization problems.

Palabras Clave: Problema de Enrutamiento de Vehículos con Limitaciones de Capacidad, Método de Algoritmo Genético, algoritmo metaheurístico, problemas de búsqueda y optimización.

Tema: Resolución del Problema de Enrutamiento de Vehículos con Capacidad Utilizando Algoritmos Genéticos.

1. Introducción

En la sociedad existen muchos problemas que requieren utilizar un conjunto de vehículos para satisfacer, de manera óptima, las necesidades de varios clientes. Ejemplos de estas situaciones son la dis-

tribución logística de una empresa para entregar mercancías a un grupo de clientes, la recogida eficiente de un grupo de alumnos para llevarlos a su centro de estudio o el recorrido de un grupo de turistas por una ciudad de forma que se pueda visitar la mayor can-

tividad de lugares posibles. Este tipo de problemas se estudian bajo el nombre de *Problema de Enrutamiento de Vehículos*.

El Problema de Enrutamiento de Vehículos (VRP, por sus siglas en inglés) es un problema de optimización combinatoria que consiste en planificar recorridos que permitan satisfacer las demandas de un conjunto de clientes geográficamente dispersos. Para ello se cuenta con una flota de vehículos que parten desde uno o varios depósitos centrales. El objetivo del VRP es diseñar rutas, para cada vehículo, que minimicen o maximicen algún objetivo.

El VRP y sus variantes han sido estudiados durante más de 50 años. La primera referencia al VRP la hace Dantzig y John Ramser [1], donde define el Problema de Enrutamiento de Vehículos con Restricción de Capacidad (CVRP), en el que cada vehículo tiene una capacidad que no debe exceder, y se desea minimizar el costo total de los recorridos.

El VRP es un problema NP-duro [2], por lo que los métodos exactos solo son factibles para resolver problemas de pequeña dimensión. Por ese motivo se han desarrollado diferentes heurísticas y metaheurísticas para solucionarlos, entre las que se encuentran las de búsqueda local [3, 4] y los algoritmos genéticos.

El objetivo de este trabajo es encontrar una heurística o metaheurística, computacionalmente eficiente y que genere resultados los más próximos posibles al óptimo real, para el *Problema de Enrutamiento de Vehículos con Capacidad*.

Este documento está estructurado en seis capítulos. En el primer capítulo se hace una breve introducción acerca de los problemas de enrutamiento de vehículos. En el capítulo 2 se define específicamente el problema de enrutamiento de vehículos a resolver y todas sus especificaciones. En el capítulo 3 se presenta el modelo matemático de la solución junto con la explicación del código realizado, mientras que en el capítulo 4 se describen los detalles del algoritmo genético programado y utilizado. En el capítulo 5 se muestran los resultados obtenidos al aplicar el algoritmo en comparación con la respuesta óptima. Finalmente, en el último capítulo, se muestran las conclusiones.

2. Problema a Resolver

Nos proponemos resolver un Problema de Enrutamiento de Vehículos con Capacidad, en donde los vehículos tienen una capacidad limitada, se encuentran en diferentes posiciones, el número de medios de transportes es finito y deben entregar a los clientes siempre en el mismo destino. Se debe considerar como restricción final que los vehículos satisfagan la mayor cantidad de demanda posible.

3. Modelos de la Solución

Para una instancia del problema se generará una ruta, la cual indicará la ruta a seguir por cada vehículo. Nos referiremos a la ruta de cada vehículo como *subru-*

tas y a la ruta total como *ruta*, la cual está conformada por subrutas dependiendo de la cantidad de vehículos que participen en esta instancia.

Cada subruta será representada como una lista y estas estarán conformadas por los clientes a quienes deben servir. El total de subrutas para una instancia del problema formará la ruta total. Por ejemplo la ruta: $[[1, 2], [5, 9], [12, 3]]$ indica que los clientes 1, 2, 5, 9, 12 y 3 serán servidos por 3 vehículos, los cuales serán divididos en 1 y 2, 5 y 9, y, 12 y 3, respectivamente.

Con el término *gen* nos referiremos a genes, haciendo alusión a cada cliente que se encuentre dentro de una ruta. Cada cliente será representado con un número entero, y este constituye la llave de un diccionario que almacena por cada cliente (a través de su número), su demanda.

Los individuos son la decodificación de las rutas en una sola lista, esto facilita la manipulación de datos para las futuras funciones que se explicarán más adelante. Por ejemplo la ruta mostrada con anterioridad se representará como un individuo de la forma: $[1, 2, 5, 9, 12, 3]$.

Cada ruta tiene su propio valor *fitness* que la distingue entre las demás y que permite diferenciar las mejores rutas de las que entregan una solución inferior. La función fitness está dada por:

$$fitness = \sum sub_route_distance \quad (1)$$

En el caso de que la solución para ese individuo esté utilizando más vehículos de los disponibles, se multiplica la solución por un factor muy grande para eliminar esta solución, es decir, la ecuación quedaría:

$$fitness = \sum sub_route_distance \cdot 999999 \quad (2)$$

La reproducción es la encargada de asegurar la diversidad genética tomando las mejores soluciones encontradas hasta ahora. Funciona de la siguiente manera:

1. Se realiza una competencia de torneo entre dos individuos al azar, de los cuales existe un ganador.
2. El individuo ganador es escogido como el padre número 1, el cual será el primer encargado de crear la descendencia.
3. Se repite el proceso para obtener al segundo padre.

Luego de tener ambos padres listos para la procreación se realiza el proceso de reproducción que consiste en lo siguiente:

1. Para el padre 1 se escogen dos genes aleatoriamente.
2. Se extrae la sección ubicada entre ambos genes y se une al individuo dos.
3. Una vez el individuo resultante es originado, se deben eliminar todos los genes repetidos partiendo de izquierda a derecha.

4. Luego repetir el procedimiento pero esta vez con el padre número 2.

La mutación nos asegura la diversidad genética, logrando solucionar posibles problemas cuando nos estamos acercando a un óptimo local que puede estar muy lejos de la solución real. La mutación funciona de la siguiente manera:

- Se escogen dos genes aleatoriamente desde el individuo.
- Una vez escogido los genes se invierte el orden de esa sección del individuo.
- Para dejar más claro tomemos como ejemplo el individuo: [123456789] del cual aleatoriamente se escogen los genes 2 y 5, lo que da paso a una mutación que da como resultado el individuo: [154326789].
- La mutación se aplica al momento de generar individuos nuevos de un proceso de reproducción, en el cual la mutación tiene una probabilidad muy chica (0.05) de que el individuo mute.

4. Algoritmo Genético

Nuestro algoritmo genético recibe como entrada los siguientes parámetros:

1. *metaheuristic*: Problema genético que contiene: un conjunto de clientes (identificación y posición) y un conjunto de vehículos (identificación, posición y capacidad de los vehículos).
2. *k*: Cantidad de participantes en el torneo de selección.
3. *ngen*: Número de generaciones.
4. *size*: Tamaño de la población.
5. *ratio_cross*: Cantidad de la población que descenderá por cruzamiento.
6. *prob_mutate*: Probabilidad de mutación.
7. *distances*: Matriz de distancias.

Primero se debe crear una población inicial, la cual es elegida aleatoriamente y que está dada por la variable *size*. Esta será la primera generación y es la que dará origen a generaciones posteriores las cuales serán cada vez más aptas. Posteriormente a la creación de nuestra primera población, debemos crear nuestras posteriores generaciones. En la nueva generación hay una cantidad de individuos que pasan directamente a la siguiente generación, estos descienden mediante selección por torneo en donde la cantidad de descendientes está dada por la cantidad total de la población menos la cantidad de individuos que resultaron mediante cruzamientos. Los individuos que pasan a la generación siguiente directamente lo hacen por elitismo, es decir, los mejores de su población.

El 85% de la población corresponde a hijos que nacieron mediante cruzamientos. Este por ciento se obtiene a partir de una selección por torneo de la siguiente forma:

- Se escogen dos individuos al azar, a los cuales se les calcula su función fitness. El individuo más apto (con mejor fitness) es quién será el padre número uno.
- Posteriormente se repite el paso anterior para obtener el padre número dos.

Por cada hijo existe una probabilidad de que este mute. Esta es una manera para no caer en óptimos locales, dándole paso a soluciones que actualmente parecen no ser las mejores para la siguiente generación. La nueva generación de soluciones más óptimas está dada por:

1. Individuos seleccionados por elitismo cuya cantidad está dada por cantidad total menos cantidad de individuos que se obtendrán por cruzamientos.
2. Individuos obtenidos por cruzamiento.

El algoritmo itera con este procedimiento hasta que el número de generaciones termina. Una vez se obtiene la generación final, el individuo con mayor fitness es el que se escoge como solución final obtenida.

Para evaluar el desempeño se emplea una métrica que determina qué tan buena es nuestra solución a partir de una solución óptima conocida para la misma entrada. La métrica de evaluación se define como:

$$\frac{\text{output} - \text{optimo}}{\text{optimo} \cdot 100} \quad (3)$$

Como podemos observar la medida de evaluación muestra el porcentaje de diferencia que el algoritmo varía de acuerdo a la solución exacta del problema.

5. Resultados

Se utilizará como prueba un problema de Augerat nombrado como An32k5 con los siguientes parámetros y especificaciones para el algoritmo:

- Cantidad de clientes: 31
- Cantidad de vehículos: 5
- Capacidad de los vehículos: 100
- Solución óptima: 787
- Cantidad de generaciones: 2500
- Tamaño de la población: 800
- Porcentaje de individuos que heredarán por cruzamiento: 85%
- Probabilidad de mutación: 0.05

Luego de 100 iteraciones de nuestra metaheurística basada en algoritmos genéticos para nuestro problema a resolver se obtuvieron los resultados mostrados en la tabla 1.

Características	Valores
Cantidad de iteraciones	100
Mejor solución	804.66
Peor solución	915.08
Promedio de las soluciones	850.86
Mejor tiempo de ejecución (segundos)	28.816
Peor tiempo de ejecución (segundos)	31.366
Promedio de los tiempos (segundos)	29.200
Total de los tiempos (segundos)	2919.961
Mejor métrica de evaluación	0.000224
Peor métrica de evaluación	0.001627
Promedio de las métricas	0.000811

Table 1: Resumen del algoritmo realizado

the heterogeneous probabilistic traveling salesman problem. *European Journal of Operational Research*, 176 2007. (Citado en la página 1).

6. Conclusiones

Si tomamos en cuenta la metaheurística analizada nos percatamos que se obtuvieron resultados un poco inferior al óptimo, aunque no muy alejado y con un tiempo aceptable pese a no tener equipos de cómputos con grandes capacidades. Se puede apreciar que la mejor solución está cerca del 2% del margen de error de acuerdo al óptimo y la peor solución tiene un 14% de diferencia con este. Además, el tiempo oscila alrededor de los 30 segundos aproximadamente, el cual consideramos como factible para problemas de gran magnitud.

Para obtener mejores resultados en cuanto al costo de la solución, es necesario sacrificar el tiempo de ejecución. Existen dos maneras, no excluyentes, para hacer esto:

1. Aumentando el número de generaciones.
2. Aumentando el tamaño de la población.

Hasta ahora la metaheurística de algoritmos genéticos parecen ser una muy buena herramienta para estimar enrutamiento de vehículos ya que nos provee soluciones muy buenas en tiempo.

References

- [1] J. H. Dantzig, G. B.; Ramser. The truck dispatch problem. *Management Science*, 6, 10, 1959. (Citado en las páginas 1 y 5).
- [2] Paolo Toth; Daniele Vigo. *The Vehicle Routing Problem (Monographs on Discrete Mathematics and Applications)*. Monographs on Discrete Mathematics and Applications. SIAM, 2001. (Citado en las páginas 1, 5 y 15).
- [3] Anand; Ochi Luiz Satoru Penna, Puca Huachi Vaz; Subramanian. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19, 04 2013. (Citado en la página 1).
- [4] Leonora Bianchi; Ann Melissa Campbell. Extension of the 2-p-opt and 1-shift algorithms to