

Heurística de dos fases para resolver el problema de ruteo de vehículos periódico

Abstract

El siguiente estudio presenta una heurística basada en un algoritmo de dos fases para resolver un *Periodic Vehicle Routing Problem* (PVRP) por medio de una programación en código en C++. Las variables de decisión son los itinerarios por día que se deben elegir para cada cliente. El objetivo es minimizar los costos totales de las rutas por semana para optimizar la entrega y de esta manera incrementar las ganancias obtenidas en la empresa. Los resultados numéricos muestran la ruta creada y el costo por semana. Las principales contribuciones de este trabajo son resolver el PVRP desde una nueva perspectiva para que se genere una solución factible en un tiempo considerable.

Palabras clave: PVRP, heurística, algoritmo de dos fases.

Introducción

Las empresas giran en torno a las necesidades de los usuarios adaptándose a los mercados globalizados que se encuentran en constante cambio, buscan reducir sus costos para maximizar sus ganancias. Uno de los principales problemas de operaciones logísticas a los que se enfrentan son los de ruteo porque los usuarios demandan la entrega de los servicios o productos a domicilio por diversos factores. Es por eso que se busca desarrollar diversos procesos, enfocándose en descubrir soluciones óptimas donde entra la disminución del costo, el número de rutas y el número de vehículos necesarios para realizarlas, incrementando la calidad en el servicio al cliente y hacer más eficiente el cumplimiento de la demanda.

En la actualidad, las heurísticas y las metaheurísticas han surgido para encontrar una solución de manera rápida y óptima que permita cumplir con el objetivo en un periodo de tiempo considerable.

Revisión de literatura

Traveling Salesman Problem (TSP) es la primera descripción de los problemas de ruteo, siendo la más sencilla. Se empezó a estudiar en 1759 por Euler. Bodin en 1993 dividió las heurísticas en tres categorías: el vecino más cercano, el método de ahorros de Clark and Wright y los métodos de inserción. [2]

Vehicle Routing Problem (VRP) fue propuesta por primera vez por Dantzing y Ramser en 1959. Bodin y Golden en 1981 agruparon las estrategias de solución en siete categorías: los procedimientos exactos, la optimización interactiva, agrupar primero y rutear después, rutear primero y agrupar después, ahorros o inserción, mejora o intercambios y programación matemática. [2]

Los problemas de ruteo se centran en diferentes aspectos, el problema más simple es el caso del TSP, donde lo único que se toma en cuenta es la distancia entre los vértices (clientes) buscando optimizar una ruta que visite a todos una vez con el menor costo posible. Después surge el VRP, que consiste en abastecer órdenes donde se tiene que determinar el número de rutas que minimicen el costo total de modo que empiecen y terminen en el depot. Cada vértice es visitado una vez por un vehículo, tomando en cuenta la demanda del cliente y la capacidad del vehículo. [1]

El VRP es uno de los problemas más fundamentales en la red de optimización debido a su aplicación y el valor económico que representa. Es por eso que ha sido un gran objeto de estudio.

El PVRP es una variante del VRP. Los investigadores se han centrado en el desarrollo de soluciones para este problema con diferentes enfoques. Beltraim y Bodin en 1974 proponen un enfoque heurístico para un problema periódico de recogida de residuos. Russell e Igo en 1979 definen el problema de asignación de enrutamiento y ofrecen un enfoque heurístico.

Christofides y Beasley en 1984 presentaron la primera formulación matemática del problema y un enfoque en dos fases de construcción y mejoramiento. Hadjiconstantinou y Baldacci en 1998 proponen una extensión de la PVRP, en la que ellos consideran un problema de planificación de los recursos de una empresa de servicios con múltiples depósitos que proporciona servicios de mantenimiento preventivo. Angelelli y Speranza en 2002 proponen un algoritmo de búsqueda tabú para la PVRP con instalaciones intermedias. Este problema es una extensión de la PVRP, en el que los vehículos pueden descargar o cargar artículos en las instalaciones intermedias y así renovar su capacidad durante su recorrido. [3]

Objetivos

Objetivo General: En la siguiente investigación se utiliza una metaheurística para desarrollar un programa en código C++, capaz de dar soluciones óptimas al problema de ruteo *Periodic Vehicle Routing Problem* (PVRP), enfocándose en la capacidad de los vehículos, la demanda de los clientes y el número de visitas que se requiere en un tiempo determinado, para generar las combinaciones de las rutas que tengan el menor costo.

Objetivos específicos:

- Estudiar el PVRP para encontrar una heurística capaz de resolver el problema.
- Codificar en C++ la heurística propuesta.
- Probar el programa con instancias de diferentes tamaños.

Metodología

El *Periodic Vehicle Routing Problem* (PVRP) es uno de los problemas más comunes en la optimización de problemas de ruteo y ha sido estudiado en los últimos años. Su objetivo es generar rutas, tomando en cuenta que existe un depósito donde se encuentran los productos y los vehículos que distribuyen a los usuarios, donde se satisface las necesidades del cliente, la

frecuencia de días que deben ser visitados, minimizando el costo de las rutas y el número de vehículos utilizados.

Las rutas se realizan en múltiples días generando horizontes de planeación. Cada cliente es distribuido en un itinerario con soluciones factibles para cada uno de ellos de acuerdo a sus especificaciones. Lo que se necesita es generar rutas al menor costo posible, tomando en cuenta que el vehículo parte y regresa al depósito en cada ruta, así como no se puede exceder la capacidad del vehículo. Se generara un algoritmo de dos fases, en primera instancia se genera la agrupación de los vértices y en segunda la creación de las rutas.

La complejidad del PVRP radica en tomar en cuenta que se debe reducir el costo de las rutas, así como crear los itinerarios de acuerdo a los días que se necesitan visitar, generando un amplio panorama de posible soluciones cuando se tienen instancias grandes.

Por otra parte, las decisiones que se toman son la secuencia en que se van asignando los clientes y la creación de los itinerarios, tomando en cuenta el número de visitas que se requieren por cliente.

La metodología que fue programada en lenguaje C++ se explica en el Pseudocódigo que se muestra en la figura 1, al programa se le proporcionan los datos de la instancia en un archivo .txt, se inicia con el número de clientes, la capacidad de los vehículos, las coordenadas donde se localizan los clientes, la demanda y la frecuencia de visitas.

Lo primero que realiza el programa es generar una matriz para conocer la distancia que existe entre cada uno de los vértices, después se aplica el algoritmo de dos fases. Primero los vértices se agrupan de acuerdo la frecuencia de visita y se generan rutas donde se utiliza el método de inserción para resolver un TSP para cada frecuencia de visita, después se van acomodando los clientes en los itinerarios posibles para su frecuencia de visita.

El vehículo tiene una capacidad establecida, por lo que cada vez que se excede se cierra la ruta para indicar que se necesita regresar al depot, para continuar después con otra ruta.

Finalmente, el ruteo por día es mostrado con su respectivo costo, así como el costo final por semana.

Pseudocódigo

Inicio

Entrada: Localización de vértices, demanda por clientes y frecuencia de visitas.

Salida: Secuencia de rutas y costo total de la semana.

1. Utilizar un método de inserción aleatoria para generar la secuencias hasta quedarse con la que minimice más el costo.
2. De acuerdo a la secuencia obtenida se van generando los itinerarios tomando en cuenta el número de visitas.
3. Cada vez que se excede la capacidad, la ruta se cierra para indicar que se debe regresar al depot.
4. Te imprime la secuencia de visitas por día y el costo total al finalizar la semana.

Figura 1

El cálculo de la distancia entre los vértices se realiza por medio de la distancia Euclidiana.

Su fórmula es $d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. El horizonte de planeación es de seis días, de lunes a sábado, eligiendo los itinerarios de acuerdo con el número de visitas que se requiere.

Si la frecuencia de visita es de:

- Seis días:

Lunes – Sábado

- Cinco días:

Lunes, Martes, Jueves, Viernes y Sábado

- Cuatro días:
Lunes, Miércoles, Viernes y Sábado
Lunes, Martes, Jueves y Sábado
- Tres días:
Lunes, Miércoles y Viernes
Martes, Jueves y Sábado
- Dos días:
Lunes y Jueves
Martes y Viernes
Miércoles y Sábado
- Un día
Lunes, Martes, Miércoles, Jueves o Viernes

Resultados

Se resolvieron tres diferentes instancias:

Primera instancia, seis clientes, capacidad del vehículo de 15, las coordenadas, la demanda y el número de visitas se presentan en la tabla 1.

Segunda instancia, 102 clientes, la capacidad de los vehículos es de 20, el resto de los datos son ingresados en el programa.

Tercera instancia, 153 clientes, la capacidad de los vehículos es de 20, el resto de los datos son ingresados en el programa.

El programa se utilizó para resolver las instancias que se encontraron en Networking and Emerging Optimization en el apartado de PVRP ingresando a “Description for files of Cordeau’s Instances” utilizando Data files P29, P32. [4]

Se codificó en lenguaje C++ y se resolvió en una computadora Intel® Core™ i5 CPU, con una velocidad de 2.27 GHz y una memoria RAM de 2 GB.

Primera instancia:

N° de cliente	Coordenada X	Coordenada Y	Demanda	N° de visitas
0	0	0	0	0
1	-5	10	9	3
2	2	8	2	4
3	6	6	6	5
4	1	2	3	3
5	9	9	7	2
6	0	3	1	5

Tabla 1

Se obtienen las rutas por día con su respectivo costo.

Lunes: $0 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 0 \rightarrow 3 \rightarrow 0$

Figura 3, mapeo de la ruta.

Costo: 34.6415

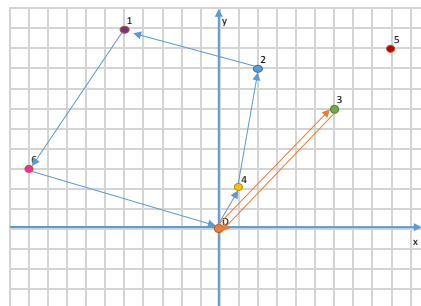


Figura 3

Martes: $0 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 0$

Figura 4, mapeo de la ruta.

Costo: 43.0930

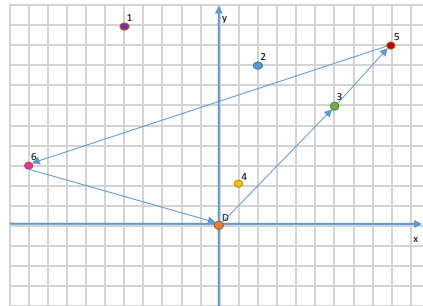


Figura 4

Miércoles: $0 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 0 \rightarrow 1 \rightarrow 0$

Figura 5, mapeo de la ruta.

Costo: 50.5441729

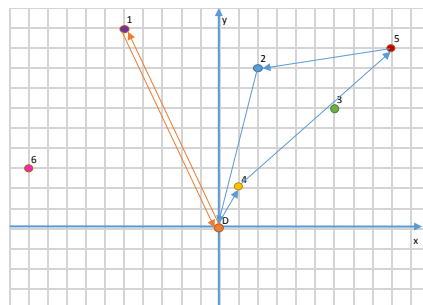


Figura 5

Jueves: $0 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 0$

Figura 6, mapeo de la ruta.

Costo: 43.0930

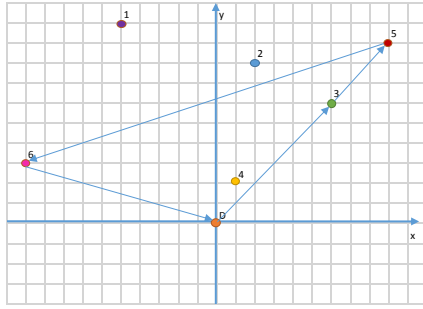


Figura 6

Viernes: $0 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 0 \rightarrow 3 \rightarrow 0$

Figura 7, mapeo de la ruta.

Costo: 34.6415

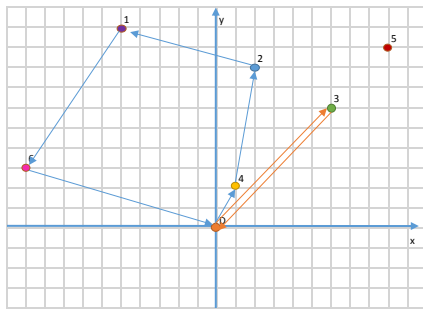


Figura 7

Sábado: $0 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 6 \rightarrow 0$

Figura 8, mapeo de la ruta.

Costo: 43.2392

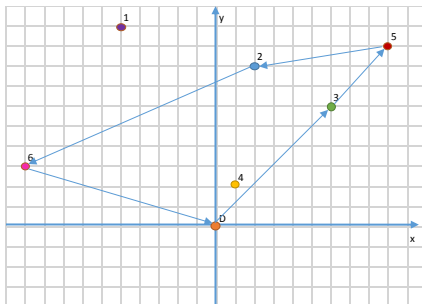


Figura 8

Costo final de la ruta: 249.252788

A continuación se presenta la solución para la instancia de 102 nodos.

Lunes

Costo: 5799.05874

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	29	20	13	14	15	21
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	63	45	62	44	46	61	42	60	59	57	0	94	74	75	92	93	73	72	71	70	68	0
---	----	----	----	----	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	---

Martes

Costo: 5536.21648

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	17	18	22	30	16	19
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	64	66	56	58	65	67	32	35	37	39	0	69	78	0
---	----	----	----	----	----	----	----	----	----	----	---	----	----	---

Miércoles

Costo: 5902.18405

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	28	23	24	25	26	27
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	41	31	48	50	51	49	47	43	40	38	0
---	----	----	----	----	----	----	----	----	----	----	---

Jueves

Costo: 4470.5933

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	29	20	13	14	15	21
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	36	34	33	54	55	52	53	88	102	89	0
---	----	----	----	----	----	----	----	----	-----	----	---

Viernes

Costo: 5631.15298

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	17	18	22	30	16	19
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	91	101	86	87	90	100	84	85	83	82	0
---	----	-----	----	----	----	-----	----	----	----	----	---

Sábado

Costo: 7694.3054

0	2	3	4	5	0	6	7	8	9	0	10	11	12	1	0	28	23	24	25	26	27
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	----	----

0	81	80	97	96	99	98	79	77	76	95	0
---	----	----	----	----	----	----	----	----	----	----	---

Costo final de la ruta: 35033.511

A continuación se presenta la solución para la instancia de 153 nodos.

Lunes

Costo: 20152.6516

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	---	----

15	0	37	28	19	20	26	27	0	69	70	67	68	82	81	79	65	66	46	0	84	126
----	---	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	---	----	-----

123	122	121	118	144	125	127	128	0
-----	-----	-----	-----	-----	-----	-----	-----	---

Martes

Costo: 15666.1097

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
---	---	---	---	---	---	---	---	---	---	---	----	----	----	---	---	----	----	----	----	---	----

15	0	21	23	22	35	36	24	0	77	64	63	78	80	47	50	49	71	72	0	129	130
----	---	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	---	-----	-----

131	132	133	134	150	149	98	97	0
-----	-----	-----	-----	-----	-----	----	----	---

Miércoles

Costo: 25413.2216

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
15	0	34	29	33	25	30	44	0	31	41	40	45	42	32	0	39	38	43	0	51	52
73	74	54	53	75	76	56	55	0	96	95	148	147	100	99	115	116	101	102	0		

Jueves

Costo: 17984.6856

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
15	0	37	28	19	20	26	27	0	57	60	61	58	62	117	48	113	112	111	0	103	104
106	105	153	152	151	135	136	137	0													

Viernes

Costo: 18063.1341

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
15	0	21	23	22	35	36	24	0	114	59	110	109	88	87	107	108	91	92	0	138	139
142	120	119	140	141	94	0															

Sábado

Costo: 26520.6298

0	6	2	3	4	0	5	1	7	8	0	12	11	10	9	0	14	13	18	17	0	16
15	0	34	29	33	25	30	44	0	31	41	40	45	42	32	0	39	38	43	0	146	89
85	86	93	90	145	124	143	83	0													

Costo final de la ruta: 123800.432

Discusión

El programa ofrece una solución factible para resolver los problemas de ruteo periódico buscando minimizar el costo, así como el tiempo en la creación de los itinerarios que debería seguir. El programa resuelve las instancias anteriores en menos de un minuto, lo que incrementa la eficiencia en una empresa y proporciona mayor flexibilidad para meter cualquier tipo de datos o cambiarlos cuando se necesite. El programa se puede correr en cualquier computadora dado que el software es gratuito. En la instancia de 102 clientes, si se genera una ruta solo con números aleatorios, obtenemos un costo por semana de 61354.3154, encontrándose el 75% arriba del costo que se obtendría si se utiliza la heurística propuesta. Por otro lado en la instancia más grande con 153 clientes, si seguimos el mismo procedimiento, obtenemos un costo por semana de 216992.46875, eso quiere decir que se encuentra el 75% arriba igual que con el ejemplo anterior, por lo que aproximadamente el costo se incrementa en la misma proporción sin importar la cantidad de clientes.

Conclusiones

Lo que se presenta en el trabajo es un algoritmo de dos fases, en el cual se crea una ruta en base al TSP utilizando un método de inserción que después se aprovecha para generar las secuencias por día considerando la capacidad de los vehículos.

Estos resultados se obtuvieron inclusive para las instancias más grandes en menos de un minuto, por lo que obtener soluciones factibles resulta práctico en un caso real donde las instancias pueden ser muy grandes.

Como un trabajo futuro se propone resolver el TSP y los itinerarios, metiendo un ciclo en todo el programa para que se corra determinado número de veces y proporcione el que genere un menor costo en el ruteo final. Explorar el uso de metaheurísticas con un formato más

general como iterated local search que privilegian la diversificación y ayudan a explorar regiones podrían dar una solución mejor.

Bibliografía

[1] Alba, E., Nakib, A. y Siarry, P. (2013). *Metaheuristics for Dynamic Optimization*. Polonia: Sprinder.

[2] Klatte, D., Luthi, H. J. y Schmedders, K. (2011). *Operations Research Proceedings*. Suiza: Sprinder.

[3] Kurz, A. y Zapfer, G., (2013). Modeling cost-delivery trade-offs for distribution logistics by generalized PRVP model. *En Sprinder Link*: 83(7), 705-726. doi 10.1007/s11573-013-0678-0

[4] S. A. (2013). Periodic VRP Instances. En *Networking and Emerging Optimization*. Recuperado de: <http://neo.lcc.uma.es/vrp/vrp-instances/description-for-files-of-cordeaus-instances/>