

Solución VRPTW haciendo uso de un método constructivo, GRASP y colonia de hormigas (ACO)

Juan Fernando Riascos.

Heurística - CM0439.
Ingeniería Física
2024-2



1 Planteamiento del Problema

2 Método constructivo

3 Método GRASP reactivo

4 Método de colonia de hormigas (ACO)

5 Cotas Inferiores

6 GAP

7 Comparación de parámetros

8 Comparación de resultados entre metodos

9 Conclusiones

10 Referencias

Definición del problema.

En el problema de enrutamiento de vehículos con ventanas de tiempo (VRPTW), un conjunto de vehículos homogéneos debe visitar un conjunto de nodos dispersos geográficamente. Todos los vehículos deben realizar una ruta visitando un subconjunto de nodos de manera que cada nodo sea visitado exactamente una vez por exactamente un vehículo. Cada ruta debe comenzar y terminar en un nodo depósito dado, y se debe minimizar la distancia total recorrida. Los vehículos se caracterizan por una capacidad, que limita su carga total, y los nodos deben ser visitados dentro de un intervalo de tiempo especificado. [1]

Formulación del problema.

En el VRPTW, una flota de K vehículos con capacidad Q debe visitar un conjunto de clientes V_c en un grafo $G = (V, E)$, partiendo y terminando en el depósito. Cada cliente i tiene una demanda q_i y una ventana de tiempo $[e_i, l_i]$ en la que debe ser atendido [1].

Las rutas deben cumplir con las siguientes restricciones:

- Cada cliente es atendido una vez.
- Las rutas comienzan y terminan en el depósito.
- La demanda total en una ruta no supera Q .
- El servicio en cada cliente debe iniciarse dentro de su ventana de tiempo, y los vehículos deben regresar al depósito antes de l_0 .

El objetivo es minimizar la distancia total recorrida.

Hacemos uso de la herramienta clase para representar nuestros nodos con las siguientes propiedades asociadas a este, como se definio previamente en el planteamiento.

- **Clase Nodo:** Cada nodo i se define por:

$i \in \{0, 1, \dots, n\}$ (con 0 como el depósito)

(x_i, y_i) Coordenadas del nodo

q_i Demanda del nodo

$[e_i, l_i]$ Ventana de tiempo [limintes inferior y superior respectivamente]

s_i Tiempo de servicio

- **Distancia Euclidiana:**

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

- Esta distancia se usa para construir la matriz de tiempos de viaje T_{ij} [1].

Función de Factibilidad: Se verifica si un nodo puede añadirse a la ruta sin violar las siguientes restricciones:

- **Restricción de Capacidad:**

$$\sum_{i \in r} q_i + q_j \leq Q$$

- **Restricción de Ventanas de Tiempo:**

$$t_{r_j} = \max(t_{r_{j-1}} + T_{r_{j-1}, r_j}, e_j) + s_i$$

Y posteriormente se revisa si se cumple que:

$$t_{r_j} < e_i$$

Selección de Rutas: El objetivo es construir una ruta factible que minimice la distancia total recorrida:

1. **Inicialización:** La ruta inicia en el depósito $r_0 = 0$.
2. **Selección del siguiente nodo:** De los nodos no visitados, se selecciona el que minimiza la distancia:

$$r_{i+1} = \arg \min_{j \in C \setminus r} T_{r_i, j}$$

3. **Verificación de factibilidad:** Se verifica que añadir el nodo no viole las restricciones.
4. **Actualización de la ruta:** Si el nodo es factible, se añade a la ruta:

$$r \leftarrow r \cup \{j\}$$

5. **Finalización:** La ruta se completa volviendo al depósito.

Planteamiento general

Con el objetivo de minimizar la distancia total recorrida, se emplea la siguiente expresión para calcular la distancia individual de cada nodo.

- **Distancia de una Ruta:**

$$D_r = \sum_{i=1}^{|r|-1} d_{r_i, r_{i+1}}$$

Luego, se suman todas las distancias individuales para obtener el recorrido total.

- **Distancia Total de las Rutas:**

$$D_{\text{total}} = \sum_{r \in R} D_r$$

Planteamiento general

Construcción de la Lista de Candidatos Restringidos (RCL):

A su vez, hay varias formas de construir la RCL:

- **RCL basada en valor:** Esta lista contiene los elementos con valor de la función voraz dentro de un rango dado:

$$\text{RCL} = \{e \in E \mid c(e) \leq \alpha \cdot (c_{\max} - c_{\min}) + c_{\min}\}$$

El valor α es un parámetro que controla el grado de aleatoriedad y $c(e)$ representa el costo del elemento e [2].

Asignación de α

Se define un conjunto de valores posibles para α

$$\delta = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$$

Donde inicialmente se calcula la probabilidad de obtener un valor de α mediante las siguientes expresiones:

$$p_i = \frac{1}{m}, \quad \forall i \in \{1, 2, 3, \dots, m\}$$

Para los valores siguiente se calcula la probabilidad mediante las siguientes expresiones :

$$p_i = \frac{q_i}{\sum_{j=1}^m q_j} \quad q_i = \frac{z^*}{A_i}$$

Donde z^* se define como la mejor solución hallada y A_i es el valor promedio de todas las soluciones halladas $\alpha = \alpha_{i^*}$ [2].

Planteamiento General del ACO

El Algoritmo de Colonia de Hormigas (ACO) es una técnica inspirada en el comportamiento de las hormigas para encontrar caminos óptimos. Las hormigas depositan feromonas en los caminos [3][4], que sirven como guía para las siguientes hormigas. El planteamiento matemático general es:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

Donde:

- $\tau_{ij}(t)$ es la cantidad de feromona en el camino entre i y j en el tiempo t .
- ρ es la tasa de evaporación de feromonas.
- $\Delta\tau_{ij}^k(t)$ es la cantidad de feromona depositada por la hormiga k entre i y j .

Colonias de hormigas

Las hormigas seleccionan su siguiente movimiento basado en una combinación de la heurística local y la cantidad de feromona en el camino [3]:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_k(i)} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \forall j \in P$$

Donde:

- $p_{ij}^k(t)$ es la probabilidad de que la hormiga k seleccione el nodo j desde i .
- $\eta_{ij} = \frac{1}{d_{ij}}$ es la información heurística (inversa de la distancia entre i y j).
- α y β son los parámetros que controlan el peso de la feromona y la heurística, respectivamente.
- $N_k(i)$ es el conjunto de nodos disponibles para la hormiga k en el nodo i .

Cotas Inferiores para el VRPTW

Se plantean las siguientes cotas inferiores para evaluar la calidad de las soluciones del problema VRPTW:

1. Cota Inferior Número de Rutas: Se basa en la demanda total de los clientes y la capacidad de los vehículos, establecida mediante:

$$\text{Cota Inferior de Rutas} = \left\lceil \frac{\text{Demanda Total}}{\text{Capacidad del Vehículo}} \right\rceil$$

2. Cota Inferior Distancia (MST): Esta cota se calcula aplicando el Árbol de Recubrimiento Mínimo a un grafo compuesto por el depósito y los clientes, donde los arcos representan las distancias:

$$\text{MST} = \sum (\text{arcos del MST})$$

Ambas cotas permiten evaluar qué tan eficiente es una solución en términos del número mínimo de rutas y la distancia más corta posible.

Cálculo del GAP

El GAP es una medida que representa cuán alejada está una solución obtenida de su cota inferior, tanto en el número de rutas como en la distancia total, se expresa como:

1. GAP en el Número de Rutas: Se calcula comparando las rutas obtenidas con la cota inferior de rutas, y está dado por:

$$\text{GAP Rutas} = \frac{\text{Rutas Obtenidas} - \text{Cota Inferior de Rutas}}{\text{Cota Inferior de Rutas}} \times 100$$

2. GAP en la Distancia Total: Se obtiene comparando la distancia total recorrida con la cota inferior de distancia:

$$\text{GAP Distancia} = \frac{\text{Distancia Total} - \text{Cota Inferior de Distancia}}{\text{Cota Inferior de Distancia}} \times 100$$

Un GAP más pequeño indica una solución más cercana a la óptima, mientras que un valor mayor sugiere una mayor desviación respecto a la cota inferior.

- **Diversidad:** ACO se beneficia del uso de múltiples agentes ($\# \text{Hormigas} = 50$), lo que incrementa la exploración del espacio de soluciones, permitiendo un balance más robusto entre explotación y exploración. En GRASP reactivo, la búsqueda es secuencial y la diversificación depende exclusivamente de los valores de α , limitando la diversidad en cada iteración.
- **Peso de Feromonas (α):** En ACO, el parámetro $\alpha = 1,5$ regula la influencia de las soluciones previas en la selección de rutas, logrando un equilibrio entre memoria histórica y visibilidad (peso del heurístico). En GRASP Reactivo, α varía entre valores predefinidos ($\{0.03, 0.05, 0.10, 0.11, 0.12\}$), lo que proporciona una diversificación limitada y estática.
- **Visibilidad (β):** En ACO, $\beta = 2$ controla el impacto de la visibilidad (heurístico de distancia). Un valor alto de β favorece soluciones cercanas en términos de distancia, lo que acelera la convergencia hacia el óptimo. En GRASP Reactivo, este concepto no está explícito, limitando el enfoque heurístico a la aleatoriedad controlada por α .

- **Evaporación de Feromonas (ρ):** ACO utiliza una tasa de evaporación $\rho = 0,7$, lo que permite evitar la sobresaturación de feromonas en rutas subóptimas, promoviendo la continua exploración del espacio de búsqueda. GRASP Reactivo no tiene un mecanismo análogo, lo que puede llevar a convergencia temprana a soluciones subóptimas.
- **Refuerzo de Feromonas (Q):** El parámetro $Q = 10,0$ en ACO asegura un refuerzo proporcional de las mejores soluciones, mejorando iterativamente la calidad. Este mecanismo es clave para que ACO supere a GRASP Reactivo, que carece de una estructura de memoria y aprendizaje acumulativo.

Es importante destacar que la influencia de estos parámetros se refleja en la disminución tanto de la distancia total recorrida como en la reducción del número de rutas utilizadas, tal como se observa en la tabla 1. Asimismo, estos parámetros impactan de manera significativa en la reducción de los GAPs tanto de vehículos como de distancia.

Criterio de comparación

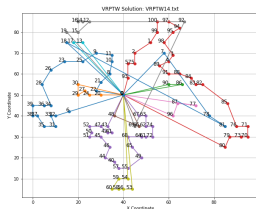
- El objetivo del problema es minimizar la distancia total recorrida, cumpliendo con las restricciones de ventanas de tiempo y capacidad de carga por vehículo. A continuación, se presenta una tabla comparativa que muestra la menor distancia alcanzada en cada sistema de nodos planteado.
- Se añade igualmente como criterio comparativo el tiempo de computo para cada método.
- Se tiene igualmente presente las comparaciones de las cotas inferiores y el GAP de la distancia y de las rutas (vehículos usados).

Tabla comparativa

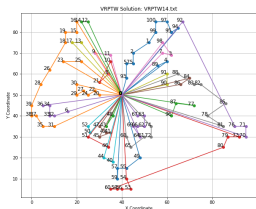
Datos Generales					Constructivo					GRASP Reactivo					ACO				
Ejemplo	n	Q	LB K	LB D	K (vehículos)	Distancia Total	GAP (vehículos)	GAP (distancia)	Tiempo (ms)	K (vehículos)	Distancia Total	GAP (vehículos)	GAP (distancia)	Tiempo (ms)	K (vehículos)	Distancia Total	GAP (vehículos)	GAP (distancia)	Tiempo (ms)
VRPTW1	25	200	3	98,99	6	334,751	1,000	2,382	1	4	264,8	0,3333	1,675	83	4	221,138	0,333	1,234	10018
VRPTW2	25	700	1	141,82	6	462,680	5,000	2,262	1	4	411,926	3	1,9045	99	3	285,436	2,000	1,013	11385
VRPTW3	25	200	2	276,78	10	681,656	4,000	1,463	1	9	640,183	3,5	1,313	80	9	617,071	3,500	1,229	8703
VRPTW4	25	1000	1	276,78	8	643,539	7,000	1,325	1	8	643,539	7	1,3251	85	6	532,224	5,000	0,923	8880
VRPTW5	25	200	3	178,41	9	743,391	2,000	3,167	1	7	620,476	1,3333	2,4779	83	7	609,480	1,333	2,416	8891
VRPTW6	25	1000	1	178,41	8	673,284	7,000	2,774	2	6	592,623	5	2,3218	90	6	573,316	5,000	2,214	8859
VRPTW7	50	200	5	195,66	12	639,903	1,400	2,271	3	11	601,69	1,2	2,0752	350	7	464,011	0,400	1,372	34482
VRPTW8	50	700	2	264,11	15	1060,556	6,500	3,016	4	13	947,36	5,5	2,5871	366	6	545,656	2,000	1,066	35451
VRPTW9	50	200	4	417,38	20	1322,489	4,000	2,169	3	17	1248,073	3,25	1,9902	314	19	1232,806	3,750	1,954	26603
VRPTW10	50	1000	1	417,38	18	1282,077	17,000	2,072	3	18	1282,08	17	2,0717	319	15	1073,181	14,000	1,571	27845
VRPTW11	50	200	5	315,74	18	1617,999	2,600	4,125	3	14	1359,559	1,8	3,306	327	16	1448,748	2,200	3,588	27954
VRPTW12	50	700	1	315,74	17	1656,929	16,000	4,248	3	15	1432,456	14	3,5369	331	14	1348,908	13,000	3,272	28675
VRPTW13	100	200	10	417,30	25	1786,623	1,500	3,281	14	30	2272,756	2	4,4663	1264	16	1173,830	0,600	1,813	126739
VRPTW14	100	1000	3	492,47	25	1894,997	7,333	2,848	17	30	2342,895	9	3,7574	1342	11	1215,006	2,667	1,467	148565
VRPTW15	100	200	8	562,26	39	2379,696	3,875	3,232	12	37	2418,648	3,625	3,3017	1292	35	2192,137	3,375	2,899	93342
VRPTW16	100	1000	2	562,26	33	2058,949	15,500	2,662	14	29	1913,64	13,5	2,4035	1329	27	1854,152	12,500	2,298	99014
VRPTW17	100	700	9	564,00	35	2831,146	2,889	4,020	13	30	2774,621	2,3333	3,9196	1286	31	2582,331	2,444	3,579	96059
VRPTW18	100	1000	2	564,00	32	2628,889	15,000	3,661	13	30	2547,367	14	3,5166	1317	29	2387,624	13,500	3,233	97098

Cuadro: Comparación de métodos Constructivo, GRASP Reactivo y ACO para las instancias VRPTW

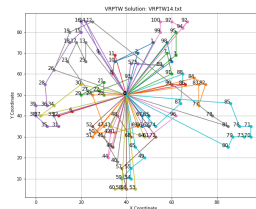
Comparación de Métodos



ACO



Constructivo



GRASP Reactivo

- **ACO como método superior de optimización:** El algoritmo de ACO equilibra eficazmente la exploración y explotación de rutas a través de un proceso iterativo robusto que converge hacia soluciones cercanas al óptimo global. El mecanismo de actualización de feromonas refuerza de manera adaptativa las soluciones más eficientes, lo que permite minimizar tanto la distancia total recorrida como el número de vehículos requeridos. Esta retroalimentación basada en el refuerzo positivo, combinada con la capacidad de ajustar iterativamente las soluciones, otorga al ACO una clara ventaja sobre otros métodos heurísticos, como GRASP reactivo y el método constructivo, en términos de reducción del *GAP*. Si bien estos últimos logran mejoras en ciertos casos, su capacidad de ajuste iterativo no es tan eficiente ni dinámica como la de ACO, que demuestra una mayor capacidad para optimizar de manera simultánea múltiples parámetros del problema VRPTW.
- **ACO y su costo computacional:** A pesar de la alta capacidad del ACO para encontrar soluciones cercanas al óptimo global, el algoritmo presenta un costo computacional considerablemente más alto en comparación con otros métodos heurísticos. Aunque el ACO logra una mejora sustancial en términos de la calidad de las soluciones y la minimización del *GAP*, su alto tiempo de cómputo puede ser un factor limitante en aplicaciones donde el tiempo de respuesta es crucial.

- [1] J. C. R. Agudelo, *Métodos constructivos y aleatorizados*, Curso: Heurística – CM0439, Universidad EAFIT, Medellín, Colombia, 15 de jul. de 2024.
- [2] J. C. R. Agudelo, *Heurística: Construcciones GRASP*, Diapositivas, 2023.
- [3] Wikipedia contributors, *Ant colony optimization algorithms* — Wikipedia, The Free Encyclopedia,
https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms,
Accessed: 2024-09-06, 2023.
- [4] J. C. R. Agudelo, *COLONIAS DE HORMIGAS HEURÍSTICA*, 08 - ACO, 2023.