

UN PROBLEMA LOGÍSTICO DE PROGRAMACIÓN DE VEHÍCULOS CON CAPACIDAD FINITA

A logistic case, the capacited vehicle routing problem

RESUMEN

Éste documento presenta la solución a un caso logístico, un problema de programación de vehículos con restricciones de capacidad CVRP (The capacitated vehicle routing problem). Éste caso es modelado en su primera parte como un problema SALBP-1 (Simple assembly line balancing 1) para determinar el mínimo número de vehículos. Nosotros usamos el algoritmo COMSOAL, y para determinar el orden en que cada vehículo debe visitar a sus clientes usamos la heurística R. Ambos algoritmos fueron implementado en Visual Basic para Excel.

PALABRAS CLAVES: Programación de vehículos con restricciones de capacidad, Problema de balanceo de línea de ensamble simple, algoritmo COMSOAL, heurística R

ABSTRACT

This document presents the solution to a logistic case, the capacited vehicle routing problem. This case is modeled first as a Simple assembly line balancing (SALB-1) for determining minimum vehicles number necessary. We used COMSOAL algorithm, and we determined every sequence how every vehicle should visit its customers with minimum cost. We used the R heuristic. Both COMSOAL algorithm and R heuristic were solved with the Visual Basic Excel.

KEYWORDS: CVRP (The capacitated vehicle routing problem), SALBP-1 (Simple assembly line balancing 1), Comsoal algorithm, R heuristic.

1. INTRODUCCIÓN

Todos los días se entregan productos en diferentes puntos de una ciudad, teniendo como referencia un origen que es el punto de distribución. Uno de los objetivos de los distribuidores es determinar el número de vehículos que se requieren para la entrega de bienes a los clientes con el objetivo de minimizar el costo de transporte. Este documento presenta un caso de estudio de una empresa distribuidora de comestibles en la ciudad de Santa Rosa de Cabal Risaralda que desea determinar el número de vehículos minimizando sus costos de transporte. El problema tiene un almacén y 20 clientes para atender con requerimientos definidos. Los vehículos tienen una capacidad de 100 unidades del producto de entrega. El problema se modela como un problema CVRP (The capacitated vehicle routing problem) donde se tienen vehículos de capacidad (carga) finita. Para determinar la solución, el problema este se modela como en su primera parte como un problema de balanceo de línea simple (SALB-1) para determinar el número de vehículos implementando el algoritmo COMSOAL. En su segunda parte para determinar las secuencias de entrega de cada vehículo se implementa la heurística R. Ambos métodos se programan en Visual Basic para Excel.

Fecha de Recepción: 25 de Enero de 2008
Fecha de Aceptación: 22 de Abril de 2008

JORGE HERNAN RESTREPO

Ingeniero Industrial, M. Sc.
Profesor Asistente
Universidad Tecnológica de Pereira
jhrestrepoco@utp.edu.co

PEDRO DANIEL MEDINA V

Ingeniero Mecánico, M. Sc.
Profesor Especial
Universidad Tecnológica de Pereira
pmedin@utp.edu.co

2. TEORÍA

El VRP (vehicle routing problem) tiene diferentes variaciones⁽¹⁾. Estas variaciones son:

2.1 CVRP (capacited VRP)

Es el VRP más general y consiste en uno o varios vehículos con capacidad limitada y constante encargados de distribuir los productos según la demanda de los clientes.

2.2 MDVRP (Multi-Depot VRP)

Es el VRP con múltiples depósitos. Cada depósito (con su flota de vehículos) debe servir a todos los clientes.

2.3 PVRP (Period VRP)

Contempla en su planteamiento un horizonte de operación de M días, periodo durante el cual cada cliente debe ser visitado una vez.

2.4 SDVRP (Split Delivery VRP)

Contempla entrega dividida, donde se permite que un cliente pueda ser atendido por varios vehículos si el costo total se reduce, lo cual es importante si el tamaño de los pedidos excede la capacidad de un vehículo.

2.5 SVRP (Stochastic VRP)

Contempla uno o varios componentes aleatorios; clientes, demanda y tiempos pueden ser estocásticos.

2.6 VRPPD (VRP pickup and Delivery)

Contempla entrega y recogida. Es aquel donde cabe la posibilidad de que los clientes pueden devolver determinados bienes, por tanto se debe tener presente que el vehículo tenga capacidad para las dos acciones. Esta restricción hace el problema más complejo y puede causar una mala utilización de los vehículos. Si el problema tiene una restricción de culminar todas las entregas antes de iniciar las recogidas da lugar a un VRPB (VRP Backhauls).

2.7 MFVRP (Mix Fleet VRP)

Contempla vehículos con distintas capacidades, por lo tanto hay que considerar estas capacidades en la ruta que seguirá cada recurso, ya que un vehículo con mayor capacidad podrá cubrir una ruta más amplia (mayor demanda).

2.8 VRPTW (VRP with Time Windows)

En el se incluye una restricción adicional donde se asocia una ventana de tiempo a cada cliente, es decir, cada cliente sólo está dispuesto a recibir la visita del vehículo durante un intervalo de tiempo dado.

El problema de programación de vehículos con capacidad finita CVRP⁽²⁾, en el únicamente se transportan bienes de un almacén o almacenes a los clientes. Las demandas de los clientes son determinísticas, se conocen de antemano y no pueden ser divididas (por ejemplo entre varios viajes o varios almacenes). Los vehículos son idénticos y tienen como base un único almacén central y únicamente tienen una restricción de capacidad.

El número de vehículos se puede obtener dividiendo los requerimientos totales sobre la capacidad de un vehículo.

$$\text{Número de vehículos} = \left\lceil \frac{\text{requerimientos_totales}}{\text{capacidad_de_un_vehículo}} \right\rceil$$

Este problema se puede formular así:

3. FORMULACIÓN MATEMÁTICA

3.1 INDICES

Los índices del modelo son:

i = nodo de partida i (1,2,...,n)

j = nodo de llegada j (1,2,..., n)

n = nodos totales

k = Vehículo k (1,2,..., K)

3.2 Variables

Las variables que se definen son:

$x_{i,j}^k = 1$ si se asigna el vehículo k para recorrer el arco

del nodo i al nodo j . cero(0) de lo contrario.

$y_{i,j} = 1$ si se realiza el recorrido desde i hasta j o cero

(0) de lo contrario.

K = Número de vehículos a utilizar.

3.3 PARÁMETROS

Los parámetros del problema son:

$C_{i,j}$ = Costo de transporte del nodo i al nodo j

d_i = Demanda en el nodo j

u = capacidad del recurso k

n = número de clientes

3.4 MODELO ⁽³⁾

$$\text{Minimizar } \sum_{(i,j) \in A} C_{i,j} y_{i,j} \quad (1)$$

Sujeto a:

$$\sum_{1 \leq k \leq K} x_{i,j}^k = y_{i,j} ; \quad \forall i, j \quad (2)$$

$$\sum_{1 \leq j \leq n} y_{i,j} = 1 ; \quad \forall i$$

$$\sum_{1 \leq i \leq n} y_{i,j} = 1 ; \quad \forall j$$

$$\sum_{1 \leq j \leq n} y_{0,j} = k$$

$$\sum_{1 \leq i \leq n} y_{i,0} = k$$

$$\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} d_i x_{i,j}^k \leq u ; \quad \forall k$$

$$\sum_{i \in Q} \sum_{j \in Q} y_{i,j} \leq |Q|$$

\forall subconjunto Q de $(1,2,\dots,n)$

$$k \leq K$$

$$y_{i,j} \in \{0,1\} ; \quad \forall (i,j) \in A$$

$$x_{i,j}^k \in \{0,1\} ; \quad \forall (i,j) \in A, \quad \forall k$$

El conjunto A se define como $A = \{ (i,j) : y_{i,j} = 1 \}$

La restricción (2) se encarga de hacer obligatoria la asignación de un vehículo a la ruta (i,j) , 1 si esta es recorrida, y 0 si no.

Las restricciones (5) y (6) indica que k es la cantidad de vehículos utilizados en la solución y que todos los que parten del almacén deben regresar al mismo.

La restricción (7) garantiza que los vehículos no sobrepasen su capacidad.

La restricción (8) vigila que la solución no contenga ciclos usando los nodos $1,2,\dots,n$.

(3) La restricción (9) limita el número de vehículos a usar.

Las restricciones (10) y (11) indican que las variables son binarias.

(4) 4. DETERMINACIÓN DEL NÚMERO DE VEHÍCULOS.

Para resolver este problema nosotros nos apoyamos en la teoría del balanceo de líneas de ensamble haciendo una analogía con entre el problema CVRP y el problema SALBP-1⁽⁴⁾(Simple assembly line balancing 1). En el SALBP-1 se conoce el tiempo de ciclo (tiempo máximo que debe invertir una estación de trabajo en procesar una o más tareas que le sean asignadas) y se desea minimizar el número de estaciones de trabajo.

(5) La analogía consiste en:

- La capacidad d_i de un vehículo será igual al tiempo de ciclo.
- El número de vehículos a determinar será igual al número de estaciones a minimizar.
- Los requerimientos de cada cliente serán semejantes a los tiempos de las tareas.

(6)

(7)

(8)

(9)

(10)

(11)

La única diferencia en la analogía es que los clientes no tienen una prioridad para ser atendidos.

En este trabajo nosotros utilizamos el algoritmo Comsoal (Computer Method for Sequencing Operations for Assembly Lines) para darle solución a este problema de mínimo número de vehículos (para el Comsoal estaciones)

PASOS :Algoritmo COMSOAL⁵

Paso 1: Para cada tarea, identifique las tareas que la preceden inmediatamente. Cree una lista A.

Paso2: para cada tarea identifique el número de predecesores inmediatos. Cree un espacio en la lista A para esta información.

Paso3: Cree una lista con las tareas que no tienen predecesores inmediatos. Cree una lista B.

Paso 4: De la lista B, cree una lista C con las tareas cuyo tiempo sea menor que el tiempo disponible de la estación. Si la lista C queda vacía, abra una nueva estación y repita el paso 4.

Paso 5: Aleatoriamente seleccione una tarea de la lista C y asígnela a la estación.

Paso 6: Actualice el tiempo disponible de la estación y la lista B. Si la lista B esta vacía retorne al paso 3, de lo contrario retorne al paso 4.

Datos del problema:

- Numero de clientes: 20
- La tabla 1 muestra los requerimientos de los clientes

1	2	3	4	5	6	7	8	9	10
15	9	24	35	23	32	45	17	49	2
11	12	13	14	15	16	17	18	19	20
13	36	15	16	37	21	49	46	10	29

Tabla 1. Requerimientos de los clientes

- Cada vehículo tiene una capacidad d_i de 100 unidades

La mejor solución obtenida se presenta en la tabla 2.

Vehículo	Cliente	Carga
1	1,2,3,4,8	100
2	5,6,7	100
3	9,10,11,12	100
4	13,14,15,16,19	99
5	17,18	95
6	20	29

Tabla 2. Programación de vehículos

5. DETERMINACIÓN DE LAS SECUENCIAS DE ENTREGA

Las secuencias de entrega de cada vehículo se determinan utilizando la heurística $R^{(6)}$.

Ésta heurística toma algunos eventos y elementos que intervienen en un partido de fútbol (campo de juego, balón, jugadores). El campo de juego es el espacio de soluciones posibles, las posiciones del balón y de un grupo de jugadores son soluciones buenas y malas del problema.

Cuando se inicia el partido de fútbol la primera posición del balón se toma como primera solución del problema y se guarda como la mejor solución hasta el momento. Ahora las posiciones que tienen los jugadores que están cerca o lejos del balón, son también posibles soluciones candidatas del problema, las cuales se deben comparar con la mejor solución hasta el momento. Si alguna de

ellas la supera, pasa temporalmente a ser la mejor hasta ahora. Seguidamente los jugadores tratan de apoderarse del balón, y para lograr este objetivo se tienen que desplazar (cambiar de posición). Estas nuevas posiciones o pasos son comparados con la mejor solución calculada hasta el momento. En caso de ser superada, será reemplazada. Es posible que algunos de los jugadores tengan que realizar más movimientos para buscar el balón.

El escenario anterior, se repite continuamente durante el partido, con la particularidad que el balón va ocupando nuevas posiciones en el campo de juego e inclusive se puede dar el evento que se repitan algunas. Y para cada una de estas nuevas posiciones del balón aparecen los mismos(en cantidad y posiblemente en posición) y/o nuevos jugadores.

Después de correr todos los escenarios, se presenta como solución del problema la mejor explorada hasta ese momento. El número de iteraciones depende del número de posiciones del balón que el investigador define previamente. El tiempo entre iteraciones depende del número de jugadores y de las posiciones que éstos toman con respecto a la posición del balón.

En conclusión, el número de posiciones del balón y el número de jugadores depende del tiempo que tenga el investigador de esperar la respuesta.

4.1 PRESENTACIÓN DE LA HEURÍSTICA R .

1. Definir el número de posiciones del balón y número de jugadores
2. Generar una solución inicial como primera posición del balón
3. Guardar esta secuencia como la mejor hasta el momento
4. Si el número de posiciones del balón no se han agotado hacer lo siguiente:
 - Generar los jugadores a partir de la posición del balón
 - Hacer que cada jugador se mueva hacia el balón y evaluar cada movimiento:
 - Sí el movimiento supera al mejor, hacer este como el mejor
 - Generar una nueva posición del balón a partir de la mejor secuencia y evaluar:
 - Sí la posición supera la mejor, hacer ésta como la mejor
 - Retornar al paso 4

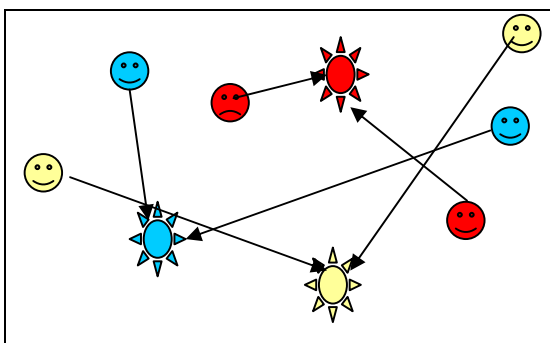
4.2 MECANISMOS DE PERTURBACIÓN

1. Generación de la nueva posición del balón: Para generar la nueva posición del balón se perturba la mejor solución que se lleva hasta el momento, seleccionando de manera aleatoria dos

- elementos de esta secuencia y se intercambia de posición.
2. Generación de las nuevas posiciones de los jugadores: Se genera la nueva posición para cada jugador perturbando la posición del balón, seleccionando de manera aleatoria dos elementos de esta secuencia y se intercambia de posición.
 3. Movimiento de los jugadores: Cada movimiento de los jugadores se hace perturbando de manera sistemática un elemento a la vez de la secuencia que representa la posición actual, buscando calcar la secuencia de la posición del balón.

Se observa que se utilizan dos mecanismos de perturbación aleatorios y uno sistemático.

En el cuadro 1, se presenta gráficamente la heurística y el cuadro 2, muestra las convenciones respectivas.



Cuadro1. Representación gráfica de la heurística R.

Posición del balón	
Posición del jugador	

Cuadro 2 Convenciones del gráfico.

Para la aplicación de la heurística se desarrolló un programa en **Visual Basic para Excel** donde el usuario puede definir el tamaño de su matriz de distancias y los parámetros de entrada que el método requiere. Se asumió los siguientes valores para los parámetros:

Parámetros:

Posiciones del balón: 100

Número de jugadores: 3

Para ejecutar la heurística se utiliza la matriz de costos (distancias) determinada por Restrepo y Sánchez⁽⁷⁾

Al aplicar la heurística para cada vehículo se obtienen los resultados de mínimo costo, que se presentan en la tabla 3

Vehículo	Orden entrega a clientes	Costo (distancia en metros)
1	4,8,3,2,1	3070
2	5,6,7	2311
3	12,9,10,11	2321
4	16,15,14,13,19	3170
5	17,18	1890
6	20	1910

Tabla 3. Secuencias de entrega y costo

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Realizar analogías entre problemas facilita la solución de estos, pues en ocasiones los métodos propuestos para resolver un tipo de problema son más fáciles de implementar y adicionalmente encuentran las soluciones rápidamente. Es el caso de usar el método COMSOAL diseñado para balanceo de línea como alternativa para determinar el mínimo de vehículos en el caso del CVRP.

5.2 Aunque al dividir los requerimientos de los clientes entre la capacidad de un vehículo nos damos cuenta aproximadamente del número de vehículos requeridos, se hace necesario determinar las secuencias de entrega por vehículo que minimicen su costo. Por esta razón se recomienda para otra instancia del problema aquí tratado usar otras metodologías y generar comparaciones.

6. BIBLIOGRAFÍA

¹ Gonzáles Vargas, Guillermo y Gonzáles Aristizábal, Felipe. Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio. Parte 1: formulación del problema. Revista Ingeniería e Investigación vol.26 N° 3 Diciembre de 2006 (149-156)

2 Valenzuela, Manuel Logística mediante inteligencia Artificial, <http://homepages.mty.itesm.mx/valenzuela/Clases/Logistica-IA/index.html>

3 Gonzáles Vargas, Guillermo y Gonzáles Aristizábal, Felipe. Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio. Parte 1: formulación del problema. Revista Ingeniería e Investigación vol.26 N° 3 Diciembre de 2006 (149-156)

4 Scholl, Armin, Becker Christian. State of art exact and heuristic solution procedures for simple assembly line balancing. pag 666-693, European Journal of Research 168 (2006)

5 McClintock Lindsay, COMSOAL Computer Method for Sequencing Operations for Assembly Lines, www.freequality.org/.../documents/Training/Classes%20Spring%202003/PowerPoint%20Slides/COMSOAL.ppt

6 Restrepo Jorge Hernán, Medina Pedro Daniel. Un problema logístico de ruteo de vehículos y una solución con la heurística R: Un caso de estudio. Scientia Technica. Año XIII - Número 37 - Diciembre de 2007 Pag 407

7 Restrepo Correa Jorge Hernán, Sánchez Castro John Jairo. Aplicación de la teoría de grafos y el algoritmo de dijkstra para determinar las distancias y las rutas más cortas en una ciudad. Scientia et Technica Año X, No 26, Diciembre 2004, Pag 121