

Capítulo I

Roteamento de Veículos

I.1 - Introdução:

Roteamento de veículos é uma designação genérica utilizada para uma grande quantidade de problemas de otimização combinatória, envolvendo basicamente, a distribuição e coleta de mercadorias e serviços.

Trabalharemos especificamente com o problema de roteamento de veículos com restrições de capacidade (PRVRC). Deveremos atender um conjunto (fixo) de clientes, cada um com uma demanda específica, utilizando para isso, uma frota de veículos com capacidades idênticas. Todos os veículos devem partir e retornar a uma mesma origem (depósito) e cada cliente deve ser visitado uma única vez. Nosso objetivo será minimizar a distância total percorrida pelos veículos atendendo a demanda dos clientes sem violar a capacidade de cada veículo.

Muitas aplicações são extensões diretas deste problema clássico de roteamento (c/ restrições de capacidade). Entre as restrições adicionais associadas ao problema podemos citar:

- 1) a utilização de vários depósitos;
- 2) a utilização de percursos que possam se iniciar em um depósito e terminar em um depósito distinto;
- 3) atendimento aos clientes em janelas de tempo (*time-windows*) previamente estabelecidas;
- 4) a utilização de um tempo total T de percurso para todos os veículos;
- 5) a utilização de uma distância máxima D para o percurso de cada veículo;
- 6) veículos que percorrem duas ou mais rotas distintas em um mesmo dia;
- 7) restrições de precedência na visita aos clientes;
- 8) a utilização de outros objetivos, como por exemplo, a minimização do número de veículos utilizados;
- 9) veículos com capacidades distintas, etc.

O problema clássico de roteamento de veículos pode ser definido em um grafo simétrico $G = (N_0, E)$ sendo que $N_0 = \{0, 1, 2, \dots, n\}$ representa um conjunto de $n+1$ vértices (o vértice 0 representa o depósito e os demais vértices os clientes). O conjunto de arestas $E = \{(i, j) / i, j \in N_0 \text{ e } i \neq j\}$ define as conexões entre cada par de vértices. A cada uma das arestas (i, j) de E , representamos por c_{ij} , o custo de transporte associado ao se viajar do vértice i para o vértice j .

Podemos encontrar uma grande quantidade de trabalhos na literatura abordando as inúmeras variações do problema clássico de roteamento de veículos (Laporte e Osman[94]). Entre os “estados da arte” sobre o assunto podemos citar os trabalhos de Bodin et al. [83], Christofides[85], Laporte e Nobert[87], Laporte[92.b] entre outros.

O problema de roteamento de veículos c/ restrições de capacidade (PRVRC) pode ser visto como uma extensão do problema do m-caixeiro viajante. No m-caixeiro viajante desejamos atender a um conjunto de clientes utilizando uma frota de m veículos com capacidade ilimitada, iniciando e finalizando cada um dos percursos em um vértice pré-determinado (depósito). Analogamente ao problema de roteamento, cada cliente deve ser percorrido uma única vez e o

objetivo será minimizar o “custo total” de transporte. Os problemas de roteamento descritos acima podem ser classificados como NP-árduos por serem uma extensão do problema do m-caixeiro viajante, reconhecidamente NP-árduo (vide Garey & Johnson[79] e Lenstra & Rinnooy [81]).

Fazemos a seguir, uma breve exposição de algumas das principais técnicas utilizadas na solução do problema de roteamento de veículos com restrições de capacidade (PRVRC):

I.2 - Métodos Heurísticos:

Em função da elevada complexidade dos problemas de roteamento em geral (problemas NP-Árduos), muitos dos algoritmos propostos para solução destes problemas, não buscam diretamente um mínimo global. Nestes casos, uma solução de baixo custo computacional (em tempo polinomial) e idealmente “próxima” da solução ótima procurada é obtida. Os métodos heurísticos podem ser divididos basicamente em duas fases: determinação de uma “boa” solução viável e busca local para melhoria desta solução.

A determinação de soluções viáveis para o problema de roteamento é frequentemente derivada de procedimentos para o caixeiro viajante (vide Laporte [92.a], [92.b]). O método do vizinho mais próximo, algoritmo de inserção e melhoramento de rotas (*tour improvement*) podem ser aplicados ao problema de roteamento quase sem modificações. A única preocupação nestes casos, é satisfazer as restrições adicionais presentes no problema de roteamento.

Uma das heurísticas mais conhecidas para o problema de roteamento (c/ restrições de capacidade) é o algoritmo de Clarke & Wright [64]. Neste caso, o número de veículos utilizados não é fixado *a priori*. O método se inicia com a construção de rotas simples (depósito-cliente-depósito) e, a cada passo, busca-se a formação de novas rotas viáveis a partir das rotas já obtidas. A justificativa para o procedimento é que o custo associado a duas rotas distintas é sempre maior que o custo associado a uma única rota que atenda a esses clientes. Desta forma, tem-se uma diminuição no “custo total” de transporte sempre que duas novas rotas puderem ser combinadas para formação de uma nova rota, sem violar obviamente, a capacidade do veículo que atende a esses clientes.

Esta situação é representada esquematicamente na Figura I.1 a seguir:

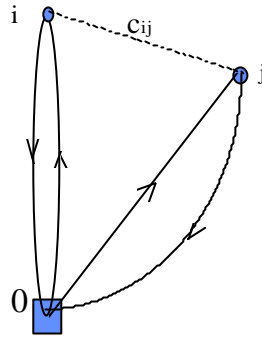


Figura I.1: Construção da rota 0-i-j-0

Note que $s_{ij} = 2c_{0i} + 2c_{0j} - (c_{0i} + c_{ij} + c_{0j}) = c_{0i} + c_{0j} - c_{ij}$ representa a economia de custo obtida após a substituição das rotas 0-i-0-j-0 pela rota 0-i-j-0.

Descrevemos a seguir, as principais etapas presentes no algoritmo de Clarke&Wright[64]. Seja n , o número total de clientes e c_{ij} o custo de transporte associado ao percurso direto do nó i para o nó j (ou vice-versa).

ALGORITMO: (Clarke & Wright [64])

Passo 0) Constrói n rotas simples (depósito-cliente-depósito);

Passo 1) calcula as economias $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, onde $i \neq j$ (o vértice 0 representa o depósito e os demais vértices representam os clientes);

Passo 2) ordena (em ordem decrescente) as economias s_{ij} ;

Passo 3) seleciona a maior economia s_{ij} ;

Passo4) se não for violada a restrição de capacidade, i e j não pertencerem a mesma rota e i (ou j) não forem interiores a nenhuma das rotas já obtidas (isto é, se i (ou j) não estiverem “ligados” ao depósito) então conectamos os vértices i e j e atualizamos a capacidade da nova rota;

Passo 5) atualiza a ordenação das economias;

Passo 6) retorna ao passo 3.

Figura I.2: Algoritmo Clarke&Wright

As economias $s_{ij} > 0$ indicam a redução obtida no custo de transporte ao percorrermos a aresta (i,j) . O algoritmo acima termina quando a lista de economias s_{ij} se tornar vazia. Este procedimento pode ser executado em $O(n^2 \log n)$ passos mas sua complexidade pode ser reduzida utilizando-se estruturas de dados apropriadas (vide p. ex. Golden et al.[77], Nelson et al. [85]). Paessens [88], propõe ainda algumas variantes deste método.

Em Laporte [92.b] pode-se encontrar outras heurísticas bastantes utilizadas para o problema clássico de roteamento (PRVRC).

Uma das técnicas mais eficientes na busca local desenvolvida para o problema do caixeiro viajante é o algoritmo de Lin & Kernighan[73] (k-optimal). Esta técnica pode ser utilizada também para o problema de roteamento após determinarmos uma solução viável para o problema. Neste caso, cada uma das rotas definidas pela solução heurística no roteamento, determina uma rota viável para o problema do caixeiro viajante. Como discutido em Savage et al.[76], os métodos de busca local aplicados a vizinhanças de tamanho polinomial nunca garantem otimalidade, ou seja, sempre podemos construir instâncias aonde a busca local se mostra bastante ineficiente (Papadimitriou & Steiglitz[78]). Logo, não temos garantida a obtenção do ótimo global do problema utilizando-se apenas métodos heurísticos.

Especialmente na última década, outras técnicas conhecidas como metaheurísticas tiveram uma grande aceitação na resolução de problemas combinatórios em geral (vide Reeves[93]). Entre elas podemos citar o *simulated annealing*, busca tabu, método GRASP, algoritmos genéticos entre outras. Estes métodos geram soluções viáveis e fazem busca local como nos métodos heurísticos clássicos, mas permitem também, um aumento no valor da função objetivo na intenção de escapar dos mínimos locais (considerando obviamente que o problema original seja de minimização).

Em Gendreau, Laporte e Potvin[94], podemos destacar a aplicação de várias metaheurísticas distintas na resolução do problema de roteamento de veículos.

I.3 - Métodos Exatos:

Os métodos exatos trabalham, em sua maioria, com esquemas de enumeração conhecidos como “busca em árvore” (*branch-and-bound*) e tem por objetivo a determinação de um ótimo global para o problema. Nestes métodos, visamos basicamente a determinação de limites inferiores (e superiores) para o custo de um ótimo global. Estes limites podem ser obtidos por procedimentos distintos, podendo-se destacar por exemplo, a relaxação lagrangeana (Geoffrion[74]) ou relaxação linear (Dantzig[51]).

Métodos de busca em árvore geram esquemas de enumeração a partir do domínio do problema original. Este domínio é dividido (idealmente particionado) sempre que o limite inferior obtido não atinja a uma solução ótima do respectivo domínio. Pesquisamos somente aquele nós (sub-problemas) da árvore de busca onde uma solução ótima pode ser encontrada. Vários nós poderão ser descartados implicitamente utilizando-se os melhores resultados parciais obtidos ao longo do processo de busca (limites inferiores e superiores). Quanto melhor forem os limites inferiores associados a cada nó, menor será o tamanho (número de nós) de nossa árvore de busca (Murty[76]).

Para ilustrar a geração de limites inferiores nos métodos de busca em árvore (*branch-and-bound*), vejamos por exemplo, a estratégia utilizada por Miller[95] na resolução do problema de roteamento de veículos (PRVRC). Miller trabalha basicamente na determinação do b-emparelhamento (*b-matching*) de custo mínimo (Miller & Pekny[94]).

Seja $G(V,E)$ um grafo completo com um conjunto V , de $n+1$ vértices, e um conjunto E de arestas. A determinação de um b-emparelhamento mínimo em $G(V,E)$ é realizada buscando-se um subgrafo $M(V, \bar{E})$ tal que, cada vértice i de V seja adjacente em exatamente b_i arestas de E . Cabe ressaltar ainda que uma aresta e pertencente a E , pode ser computada d_e vezes. Como veremos a seguir, o problema do b-emparelhamento escolhendo-se convenientemente os valores de b_i para cada $i \in V$, e d_e para cada $e \in E$, pode ser visto como uma relaxação para o problema de roteamento de veículos c/ restrições de capacidade.

Os custos e a capacidade de cada aresta e pertencente a E são representados por c_e e d_e respectivamente. Para cada subconjunto $S \subseteq V$, representamos por $\mathcal{S}(S)$ o conjunto das

arestas de E com ambas as extremidades em S , e $\mathbf{d}(S)$ o conjunto das arestas pertencentes a E com apenas uma das extremidades em S . Seja x_e , a variável que representa o número de vezes que computamos a aresta $e \in E$. Temos então a seguinte relaxação para o PRVRC:

$$\min \sum_{e \in E} c_e x_e \quad (1)$$

$$\text{s.a} \quad \sum_{e \in \mathbf{d}(i)} x_e = b_i, \quad \forall i \in V \quad (2)$$

$$0 \leq x_e \leq d_e \quad (3)$$

$$x_e, d_e \in \mathbb{Z}^+, \quad \forall e \in E \quad (4)$$

onde:

$$b_0 = 2K \text{ é grau do depósito;}$$

$$b_i = 2 \quad \forall i \in \{1, \dots, n\};$$

$$d_e = 2, \quad \forall e \in \mathbf{d}(\{0\});$$

$$d_e = 1, \quad \forall e \notin \mathbf{d}(\{0\}).$$

Note que o conjunto de restrições (2), indica a existência de K rotas (já que temos $2K$ arestas incidentes ao depósito). Além disso, cada cliente deve ser visitado uma única vez. O conjunto de restrições (3) e (4) indicam que apenas as arestas incidentes ao depósito podem ser duplicadas, possibilitando a existência de rotas simples do tipo (depósito-cliente-depósito).

Note, na formulação (1)-(4) acima, que temos uma relaxação (limite inferior) para o problema de roteamento já que as restrições de capacidade de cada veículo e conexidade do grafo são desprezadas.

Representamos na figura I.3 uma solução viável para o b-emparelhamento com 3 veículos e 11 vértices:

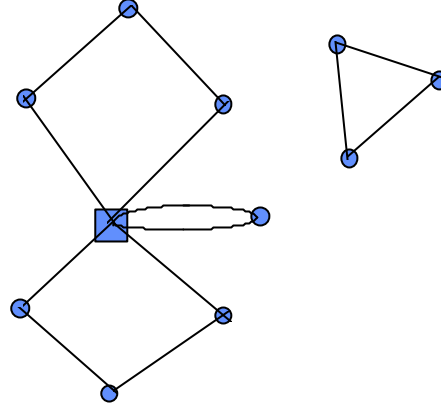


Figura I.3: Solução viável p/ o problema do b-emparelhamento

Seja q_i , a demanda associada a cada cliente e C , a capacidade de cada veículo. Uma formulação para o PRVRC pode ser apresentada se introduzimos o seguinte conjunto de restrições (de eliminação de sub-rotas) às restrições que definem o b-emparelhamento:

$$\sum_{e \in \mathbf{S}(S)} x_e \leq |S| - \left\lceil \frac{a(S)}{C} \right\rceil \quad \text{onde } a(S) = \sum_{i \in S} q_i, \quad S \subset V \setminus 0 \text{ e } |S| \geq 3 \quad (5)$$

Seja Ψ , o conjunto de todos os subconjuntos S que satisfazem (5). Se dualizamos o conjunto (exponencial) de restrições (5) e associamos multiplicadores $\mathbf{I}_S \geq 0$, a cada uma destas restrições, teremos o seguinte problema lagrangeano:

$$L(x, \mathbf{I}) = \min \left\{ \sum_e \bar{c}_e x_e - \sum_{S \in \Psi} \mathbf{I}_S \left(|S| - \left\lceil \frac{a(S)}{C} \right\rceil \right) \right\}$$

$s.a \quad (2) - (4)$

$$\text{onde } \bar{c}_e = c_e + \sum_{\{S | e \in \mathbf{S}(S)\}} \mathbf{I}_S.$$

Como temos um número exponencial de restrições de eliminação de sub-rotas são dualizadas apenas aquelas que venham a violar a solução de um problema lagrangeano (Miller[95]).

Desejamos ajustar os multiplicadores de Lagrange de maneira a obter o maior limite inferior possível. Para isto, podemos resolver o seguinte problema dual lagrangeano pelo método subgradiente (Geoffrion[74], Fisher[81]):

$$\begin{aligned} \max \quad & L(x, \mathbf{I}) \\ \text{s. a} \quad & \mathbf{I} \geq 0 \end{aligned} \quad (\text{PD})$$

No método subgradiente, basicamente, determinamos um conjunto de multiplicadores λ inicial e resolvemos o problema lagrangeano associado. Em seguida, esses multiplicadores são atualizados e o processo é repetido até que algum critério de parada tenha sido atendido. Maiores detalhes podem ser encontrados em Geoffrion[74] ou Fisher[81].

Assim, para cada conjunto de multiplicadores λ dado, temos o problema do b-emparelhamento, que pode ser resolvido em tempo polinomial (vide Miller e Pekny[94]). Em seu trabalho, Miller[95] implementa ainda uma pequena variação do método subgradiente apresentado em Geoffrion[74].

Entre os métodos de resolução para o problema de roteamento de veículos utilizando programação linear inteira podemos citar o trabalho de Balinski e Quandt[64]. Nesta abordagem, eles utilizam uma formulação que trabalha com a partição de conjuntos (*set partitioning*) e geração de colunas.

Seja J , o conjunto de todas as possíveis rotas \mathbf{j} viáveis e a_{ij} um conjunto de coeficientes binários 0-1, representando 1 se e somente se o vértice $i > 0$ aparece na rota \mathbf{j} e 0 em caso contrário. Representaremos por c_j^* , o custo ótimo associado a rota \mathbf{j} e por x_j , a variável binária igual a 1 se e somente se a rota \mathbf{j} aparece na solução ótima. Se não fizermos restrições quanto ao número de veículos utilizados o problema de roteamento poderá ser formulado da seguinte maneira:

$$\min \sum_{j \in J} c_j^* x_j \quad (6)$$

$$\text{s.a.} \quad \sum_{j \in J} a_{ij} x_j = 1, \quad \forall i \in V \setminus \{0\} \quad (7)$$

$$x_j \in \{0,1\}, \quad \forall j \in J \quad (8)$$

Observe que o conjunto de restrições (7) indica que cada cliente deve pertencer a uma única rota. Podemos destacar 2 dificuldades principais associadas a esta formulação:

- (i) o número extremamente grande de variáveis binárias x_j ;
- (ii) a grande dificuldade no cálculo dos custos c_j^* .

Note (item (ii)) que no problema de roteamento (PRVRC), cada rota j corresponde a um conjunto de vértices S_j satisfazendo a capacidade de cada veículo. O valor c_j^* é então obtido resolvendo-se o problema do caixeiro viajante em S_j . Se considerarmos que o número de variáveis envolvidas seja razoavelmente pequeno (não mais que alguns milhares de variáveis), qualquer algoritmo desenvolvido especificamente para o problema de partição pode ser utilizado (vide Balas & Padberg[79], Marstens[74]).

Uma abordagem natural para se contornar tais dificuldades é utilizarmos o método de geração de colunas na obtenção da relaxação linear de (6)-(8) (isto é, (8) é substituído por $0 \leq x_j \leq 1, \forall j \in J$). Esta técnica foi aplicada inicialmente ao problema de roteamento de veículos por Rao e Zionts[68], Foster e Ryan[76], Orloff[76] entre outros.

Na geração de colunas são construídos, a cada iteração, problemas reduzidos contendo um subconjunto de todas as possíveis colunas (variáveis). A relaxação linear do problema reduzido gera variáveis duais ótimas λ . Para checarmos a otimalidade da solução, calculamos uma coluna s (associada a uma variável x_s) e satisfazendo a:

$$c_s^* - \mathbf{I}y_s = \min_{j \in J} \{c_j^* - \mathbf{I}y_j\} \quad (9)$$

onde y_j é o vetor coluna associado à variável x_j . Se o custo reduzido associado à variável x_s for positivo, a solução corrente será ótima. Caso contrário, x_s entra na base e o problema é reotimizado. Como no problema de roteamento de veículos as soluções viáveis devem ser inteiras, este procedimento deve ser utilizado conjuntamente com o algoritmo *branch-and-bound* como forma de garantir soluções ótimas. Desrosiers, Soumis e Desrochers[84] resolvem (9) utilizando um algoritmo para o problema do caminho mínimo com as mesmas restrições que o problema original de roteamento.

A técnica de programação dinâmica foi originalmente proposta para o problema de roteamento de veículos por Eilon, Watson-Gandy e Christofides[71]. Outras técnicas geram limites inferiores utilizando programação dinâmica. Na programação dinâmica, a principal etapa é a determinação de uma fórmula de recorrência. A solução ótima do problema original ou de uma relaxação para o problema original é então obtida iterativamente a partir da base do processo recursivo. Entre os principais trabalhos que utilizam programação dinâmica podemos citar Christofides et al.[81.a], a relaxação de espaços de estado de Christofides, Mingozzi e Toth[81.b] e mais recentemente, Hadjiconstantinou, Christofides e Mingozzi[95].

Em Lucena[86], é proposto um algoritmo exato para o problema de roteamento (PRVRC). O procedimento elimina inicialmente várias rotas sub-ótimas através do uso de limitantes e condições de dominância derivadas da relaxação de espaços de estados. Um conjunto de rotas contendo uma solução ótima é identificado e um problema de partição (*Set partitioning*) é resolvido. Uma abordagem envolvendo uma formulação de dois fluxos é também apresentada.

Em Araque, Kudva, Morin e Pekny[94]; Hill[95] e Augerat[95] são propostos algoritmos do tipo *branch-and-cut*, na solução do problema clássico de roteamento. Em todas as abordagens apresentadas são resolvidas instâncias que variam de 50 a 150 clientes. Novas classes de restrições válidas para o problema são também apresentadas.