

3

Algoritmos Exatos

Nesse capítulo, apresenta-se alguns algoritmos exatos para o CVRP que são baseados em diferentes técnicas e formulações para a obtenção de limites inferiores para a solução ótima do CVRP. Todos eles são importantes para o estudo do CVRP e para o entendimento do algoritmo de *branch-and-cut-and-price* que é proposto nessa dissertação. Primeiramente, são mostrados algoritmos que utilizam a abordagem de relaxação Lagrangeana. Depois, são mostrados algoritmos baseados em geração de colunas. E, por fim, são mostrados algoritmos que utilizam planos de corte.

3.1

Relaxação Lagrangeana

Essa seção descreve um algoritmo proposto em CMT [21], que apresenta diferentes propostas de relaxação Lagrangeana, onde os melhores resultados são obtidos utilizando o conceito de q -rotas. Os limites inferiores utilizados pelo algoritmo são obtidos seja diretamente através da resolução por programação dinâmica de uma formulação relaxada do CVRP, seja através da utilização de um algoritmo de subgradiente para resolver a relaxação Lagrangeana que utiliza restrições para reforçar a formulação correspondente ao primeiro limite.

Antes de apresentar-se estes dois limites inferiores apresentados em [21], define-se o que é uma q -rota, um conceito importante não só nesse capítulo, mas também ao longo de toda a dissertação. Uma q -rota pode ser definida como sendo um caminho de custo mínimo que vai do depósito a um determinado cliente i e depois retorna ao depósito. Nesse caminho a demanda acumulada não ultrapassa um valor dado (capacidade do veículo). Além disso, um cliente pode ser visitado mais de uma vez e a cada vez

toda a sua demanda é carregada no veículo. Uma q -rota é uma relaxação da solução do CPCTSP. Se o tamanho dos ciclos que podem ocorrer nas q -rotas forem restritos a ter no mínimo um valor fixo s , esta relaxação é fortalecida, o que leva a melhores limites inferiores. Para um valor de s igual ao número de clientes total, ou seja proíbe-se que a q -rota retorne a qualquer cliente, tem-se uma q -rota que corresponde à solução do CPCTSP.

Seja J o conjunto de todas rotas viáveis para o CVRP. Considerando o politopo R_3 descrito na seção 2.3, deriva-se o limite inferior descrito em [21]. Considere particionar o conjunto J em n blocos, onde o bloco i ($i = 1, \dots, n$) está associado ao índice i do último cliente nas rotas do bloco. Dentro dos blocos as rotas serão indexadas por s .

$$\text{Seja } \xi_{is} = \begin{cases} 1 & \text{se a rota } s \text{ está na solução} \\ 0 & \text{caso contrário.} \end{cases}$$

Assim, pode-se definir o politopo R_6 ,

$$\sum_{(i,s) \in \bar{N}_j} \xi_{is} = 1, \quad j = 1, \dots, n, \quad (3-1)$$

$$\sum_{i=1}^n \sum_{s=1}^{m_i} \xi_{is} = K, \quad (3-2)$$

$$\xi_{is} \in \{0, 1\}, \quad (3-3)$$

onde o par ordenado (i, s) é usado para rotular a rota s do bloco i , $\bar{N}_j$ é o conjunto de pares ordenados (i, s) identificando as rotas contendo o cliente j , e m_i é o número de rotas no bloco i (m é o número total de rotas).

Tem-se, então, a formulação a partir do politopo R_6 ,

$$\min \left\{ \sum_{i=1}^n \sum_{s=1}^{m_i} c_{is} \xi_{is} : \xi_{is} \in R_6 \right\}, \quad (3-4)$$

onde c_{is} é o custo da rota (i, s) .

Pode-se obter uma relaxação da formulação (3-4), substituindo-se as restrições (3-1) pelas seguintes restrições mais fracas:

$$\sum_{s=1}^{m_i} \xi_{is} \leq 1, \quad i = 1, \dots, n, \quad (3-5)$$

$$\sum_{i=1}^n \sum_{s=1}^{m_i} d(is) \xi_{is} = d(V_+), \quad (3-6)$$

onde $d(is)$ é a demanda acumulada na rota (i, s) e $d(V_+) = \sum_{i=1}^n d_i$. O politopo definido pela substituição das inequações (3-1) pelas inequações (3-5) e (3-6) será referenciado como R_7 . Temos, então, a seguinte formulação

$$\min \left\{ \sum_{i=1}^n \sum_{s=1}^{m_i} c_{is} \xi_{is} : \xi_{is} \in R_7 \right\}. \quad (3-7)$$

A formulação (3-7) pode ser reduzida, aplicando-se o conceito de dominância que determina que a rota (i, s^*) domina todas rotas (i, s) se $c_{is^*} \leq c_{is}$ e $q_{is} = q_{is^*}$. Assim, para cada i e cada quantidade q acumulada de demanda, apenas uma rota deve ser considerada. Seja essa rota (i, l) . Seja W o conjunto de todas as quantidades possíveis de demanda acumulada e $|W| = w$. Logo existem w rotas a serem consideradas para cada i . O problema relaxado é descrito abaixo através do politopo R_8 ,

$$\sum_{l=1}^w \xi_{il} \leq 1, \quad i = 1, \dots, n, \quad (3-8)$$

$$\sum_{i=1}^n \sum_{l=1}^w d(l) \xi_{il} = d(V_+), \quad (3-9)$$

$$\sum_{i=1}^n \sum_{l=1}^w \xi_{il} = K, \quad (3-10)$$

$$\xi_{il} \in \{0, 1\}. \quad (3-11)$$

A formulação para o problema pode ser dada então como

$$\min \left\{ \sum_{i=1}^n \sum_{l=1}^w c_{il} \xi_{il} : \xi_{il} \in R_8 \right\}. \quad (3-12)$$

Um limite inferior para o valor c_{il} (o custo da rota mínima com demanda total $d(l)$ e tendo i como último cliente visitado) é o custo m_{il} da q -rota mais barata com demanda acumulada l . Então o limite LB1 pode

ser definido como sendo:

$$\min \left\{ \sum_{i=1}^n \sum_{l=1}^w m_{il} \xi_{il} : \xi_{il} \in R_8 \right\}. \quad (3-13)$$

O limite inferior LB1 é calculado utilizando programação dinâmica. Seja $h_i(b, a)$ a solução ótima para (3-13) com o lado direito de (3-9) substituído por b e o lado direito de (3-10) substituído por a , com $\xi_{jl} = 0$ para $j > i$, $l = 1, \dots, w$. A função $h_i(b, a)$ pode ser computada através da recursão

$$h_i(b, a) = \min \left[h_{i-1}(b, a), \min_{q \in W} \{ h_{i-1}(b - q, a - 1) + m_{i, \pi(q)} \} \right] \quad (3-14)$$

para $a = 2, \dots, K$; $i = a, \dots, n - K + a$; $d(V_+) - (K - a) \cdot C \leq b \leq \min[a \cdot C, d(V_+)]$. Onde $\pi(q)$ representa a melhor rota com demanda acumulada igual a q .

A função h_i é inicializada com:

$$h_i(b, 1) = \begin{cases} m_{i, \pi(b)} & \text{se } b \in W, \\ \infty & \text{se } b \notin W. \end{cases}$$

O limite inferior LB1 é, então,

$$LB1 = h_n(d(V_+), K).$$

Em [21], este limite inferior é reforçado pelas restrições

$$\sum_{i=1}^n \sum_{l=1}^w (d_i/d(l)) \xi_{il} = K, \quad (3-15)$$

$$\sum_{i=1}^n \sum_{l=1}^w d_i \xi_{il} = d(V_+). \quad (3-16)$$

Isso dificulta a resolução do problema. Para resolver isso, as restrições (3-15) e (3-16) são relaxadas e resolvidas utilizando relaxação Lagrangeana. Um algoritmo de subgradiente é utilizado para obter o novo limite (chamado de LB2), onde as iterações usam a programação dinâmica utilizada para o

cálculo de LB1.

3.2

Geração de Colunas

Embora a seção anterior descreva uma abordagem onde um número exponencial de rotas é considerado implicitamente, esta abordagem não utiliza algoritmos de programação linear para resolver as relaxações propostas. Nesta seção, discute-se duas propostas de algoritmos para resolução do CVRP sobre formulações também com número exponencial de variáveis, onde, entretanto, utiliza-se programação linear na obtenção dos limites inferiores.

Na abordagem proposta em Agarwal et al.[7], a dificuldade maior da geração de colunas que, como citado anteriormente, corresponde à resolução de um CPCTSP, é contornada de uma forma diferente da utilizada por CMT [21] e descrita acima. Neste caso, a função que determina o custo de uma rota é linearizada, fazendo com que o problema de geração de colunas passe a ser um problema da mochila (KP , do inglês *Knapsack Problem*).

Seja o politopo R_3 descrito na seção 2.3. No esquema de geração de colunas, apenas as colunas na base corrente são conhecidas e outras colunas são geradas quando necessário. Assim, em cada iteração deve ser identificada a coluna λ_j com menor custo reduzido $\bar{c}_j$. Se $\bar{c}_j \geq 0$, a solução corrente é ótima; caso contrário, a coluna λ_j entra na base. Note que todas colunas binárias $y_i = \{0, 1\}$ satisfazem às restrições de capacidade do veículo ($\sum_{i=1}^n d_i y_i \leq C$).

Seja o politopo R_9 definido como

$$\sum_{i=1}^n d_i y_i \leq C, \quad (3-17)$$

$$y \in \{0, 1\}. \quad (3-18)$$

Então uma coluna λ_j é definida por y^* , a solução ótima para o problema

$$\min \left\{ f(y) - \sum_{i=1}^n u_i y_i : y \in R_9 \right\}, \quad (3-19)$$

onde $f(y)$ é o custo da solução ótima do TSP considerando somente os clientes visitados na rota representada pela coluna e u_i é a variável dual associada a i -ésima linha do LP.

Note que a função $f(y)$ é bastante complexa e seu valor só pode ser obtido resolvendo um TSP. Isso torna o subproblema difícil de resolver. No entanto, se $f(y)$ for substituída por uma função linear em y (como $f(y) = \sum_i p_i y_i$), então o subproblema se transforma em um KP, que é bem mais fácil de ser resolvido. Apesar de $f(y)$ não poder ser escrita como uma função linear, é possível encontrar uma função linear que dá um limite inferior para $f(y)$. Assim, o algoritmo utiliza esse limite para cada nó da árvore de *branch-and-bound*.

Outra abordagem interessante utilizando geração de colunas é dada em Desrosiers et al.(1984) [26]. Nesse caso o problema resolvido não é o CVRP, mas o problema de roteamento de veículos com janelas de tempo (*VRPTW*, do inglês *vehicle routing problem with time windows*). No algoritmo de geração de colunas utilizado, o subproblema de geração de colunas não é mais um CPCTSP e sim um caminho mais curto que é resolvido utilizando uma adaptação do algoritmo de Ford-Bellman que é chamado pelos autores de caminho mais curto com janelas de tempo (*SPTW*, do inglês *Shortest path with time windows*).

3.3

Planos de corte

Conforme mencionado na introdução desta dissertação, a partir dos anos 90, a pesquisa no CVRP se concentrou no estudo poliedral do problema e no desenvolvimento de algoritmos de *branch-and-cut* derivados desse estudo. Diversas desigualdades válidas para o CVRP foram identificadas, algumas muito complexas, e algoritmos de separação foram desenvolvidos para as mesmas. Alguns algoritmos importantes de *branch-and-cut* desenvolvidos são descritos em Ralphs(1993)[65], Augerat et al.(1995)[1] e Lysgaard et al.(2004)[46].

A idéia de um algoritmo de planos de corte (ou *branch-and-cut*) é, dado um LP inicial, resolvê-lo e adicionar desigualdades violadas (cortes) a medida que elas forem sendo encontradas. Para encontrar as desigualdades violadas a serem adicionadas ao LP, são utilizados os algoritmos de separa-

ção mencionados acima. Os algoritmos de separação encontram dentre todas desigualdades de uma família de desigualdades válidas para o problema, pelo menos uma violada, se existir, pela solução ótima corrente do LP.

O algoritmo de BC descrito em [46] pode ser considerado o mais elaborado desenvolvido e o com melhores resultados. Nele são utilizadas diversas desigualdades para o CVRP (utilizaremos os nomes em inglês): *capacity*, *framed capacity*, *strengthened comb*, *multistar*, *partial multistar*, *extended hypotour*. Além disso, são utilizados também os clássicos cortes de Gomory. Para cada uma dessas desigualdades, algoritmos de separação elaborados são utilizados para separar de maneira eficiente (na maioria dos casos são utilizadas heurísticas, já que a separação exata normalmente é cara computacionalmente) os cortes violados. Os algoritmos de separação foram utilizados também no algoritmo de BCP dessa dissertação. As desigualdades e os algoritmos de separação utilizados são apresentados de maneira sucinta no capítulo 5.

Observe que as restrições de capacidade (*capacity cuts*) têm que ser sempre completamente separadas, isto é, a solução final não pode ter nenhuma restrição de capacidade violada pois, nesse caso, não seria uma solução viável. Entretanto, as demais desigualdades válidas podem ser separadas ou não. Esta decisão depende da relação entre a melhoria do limite inferior e o esforço computacional necessário para a separação.

Os limites inferiores obtidos em [46] eram os melhores disponíveis na literatura até este trabalho. Com o algoritmo de BC descrito pelos autores, foram resolvidas três instâncias que estavam em aberto (B-n50-k8, B-n66-k9 e B-n78-k10).

A abordagem por planos de cortes descrita nessa seção pode ser considerada complementar às abordagens descritas nas duas seções anteriores. O algoritmo de BCP dessa dissertação utiliza esse fato para, unindo as abordagens da seção anterior e atual, obter resultados excelentes para o CVRP.