1 Programação Inteira e *Local branching*

Ao gerar uma rota, algoritmo de verificação de viabilidade e inserção de tempo de parada nos clientes restringe algumas ações do veículo, por exemplo, ao não considerar a possibilidade de o veículo fazer uma parada ao longo do percurso entre clientes. Na tentativa de incorporar todas as ações permitidas pela definição do problema no processo de solução, optou-se por utilizar técnicas de programação inteira para otimizar as rotas. Assim, para uma determinada solução gerada heuristicamente, cada rota será otimizada através de um algoritmo de busca local definido sobre um problema de programação matemática, conhecido como *Local Branching*.

O modelo matemático é descrito a seguir e, posteriormente, o algoritmo de *local branching* será detalhado.

1.1 Modelo matemático

Inspirado no modelo proposto em [Xiao and Konak, 2016] para o HGVRSP, o modelo a seguir foi adaptado para otimizar uma única rota r, previamente gerada de forma heurística, permitindo paradas em qualquer ponto do trajeto. Somente uma rota é considerada, porque, devido à complexidade do modelo, só é possível resolver instâncias pequenas (e mesmo assim com tempo de execução elevado).

O modelo considera um subconjunto do grafo original $G(V, A)$ de uma instância, restrito aos pontos que compõem uma rota específica r. O modelo mantém o mesmo veículo da rota indicada (isso é, o veículo é fixo), mas permite que a ordem de visitação dos clientes e, consequentemente, os instantes de tempo relacionados sejam modificados. A Tabela 1.1 apresenta os dados de entrada do modelo matemático. As variáveis do modelo são descritas na Tabela 1.2, sendo as variáveis x_{ij} e x_{ij}^k binárias e as demais contínuas.

Tabela 1.1: Parâmetros do modelo

Parâmetros	Descrição
V_r	Conjunto de pontos (depósito e clientes) da rota r
V_r^+	Conjunto de clientes da rota r (excluindo o depósito)
A_r	Conjunto de arestas do subgrafo induzido por V_r
A_r^*	Subconjunto de arestas de A_r composto pelas arestas originais da rota r
M	Conjunto dos períodos
i, j	Índices dos clientes
k, m	Índices dos períodos, sendo que m representa o último período
$D_{i,j}$	Distância entre aresta (i, j)
Dem_i	Demanda do cliente i
TS_i	Tempo de serviço do cliente i
$[T_i^{Ini}, T_i^{Fim}]$	Janela de tempo do cliente i
$[T_k^{Ini}, T_k^{Fim}]$	Início e fim do período k
$[T_v^{Ini}, T_v^{Fim}]$	Janela de tempo do veículo
Vel_{ij}^k	Velocidade do veículo do arco (i, j) no período k
Cb^{Max}	Capacidade máxima do tanque de combustível do veículo
v	Tipo de veículo v associado a rota r.
$Cb_{ij}^{k,v}$	Taxa de consumo de combustível ao percorrer o arco (i, j) no período k
Cb_v^+	Consumo adicional de combustível para uma unidade a mais de carga
$Tx_v^{CO_2}$	Taxa de conversão combustível / CO_2

Tabela 1.2: Variáveis do modelo matemático que otimiza uma rota r

Variáveis	Descrição
x_{ij}	Indica se o arco (i, j) é percorrido ou não
x_{ij}^k	Indica se um arco (i, j) é percorrido em um período k
d_{ij}^k	Indica a distância percorrida do arco (i, j) no período k
τ_{ij}^k	Indica o tempo gasto no arco (i, j) no período k
t_i^{sai}	Indica o instante de tempo em que o veículo sai do cliente i
t_i^{chg}	Indica o instante de tempo em que o veículo chega no cliente i
f_{ij}	Indica a carga do veículo ao percorrer o arco (i, j)
q_{comb}	Indica a quantidade total de combustível utilizada na rota.

O modelo é apresentado nas equações 1.1 à 1.21.

$$\text{Minimizar} \quad CO_2 = Tx_v^{CO_2}.q_{comb} \tag{1.1}$$

$$\text{Sujeito a} \quad \sum_{j \in V_r} x_{ij} = \sum_{j \in V_r} x_{ji} = 1 \qquad \forall i \in V_r \tag{1.2}$$

$$x_{ij}^k \le x_{ij} \qquad \forall (i,j) \in A_r, k \in M \tag{1.3}$$

$$\sum_{l \in V_r, l \neq i,j} x_{jl}^{k'} \le 1 - \sum_{i \in V_r, i \neq j} x_{ij}^k \qquad \forall k', k \in M, k' < k, \quad \forall j \in V_r^+ \tag{1.4}$$

$$d_{ij}^k \le D_{ij}.x_{ij}^k \qquad \forall (i,j) \in A_r, k \in M \tag{1.5}$$

$$\sum_{k \in M} d_{ij}^k = D_{ij}.x_{ij} \qquad \forall (i,j) \in A_r \tag{1.6}$$

$$\sum_{i \in V_r, i \neq j} f_{ij} - \sum_{i \in V_r, i \neq j} f_{ji} = Dem_j \qquad \forall j \in V_r \tag{1.7}$$

$$f_{ij} \le (\sum_{l \in V_r^+} Dem_l).x_{ij} \qquad \forall (i,j) \in A_r \tag{1.8}$$

$$\tau_{ij}^k = d_{ij}^k / Vel_{ij}^k \qquad \forall (i,j) \in A_r, k \in M \tag{1.9}$$

$$\sum_{(i,j) \in A_r} \tau_{ij}^k \le T_k^{Fim} - T_k^{Ini} \qquad \forall k \in M \tag{1.10}$$

$$t_i^{sai} \le T_k^{Fim} - \tau_{ij}^k + T_m^{Fim}(1 - x_{ij}^k) \qquad \forall (i,j) \in A_r, \quad \forall k \in M \tag{1.11}$$

$$t_j^{chg} \ge T_k^{Ini} + \tau_{ij}^k - T_m^{Ini}(1 - x_{ij}^k) \qquad \forall (i,j) \in A_r, \quad \forall k \in M \tag{1.12}$$

$$t_j^{chg} \ge t_i^{sai} + \sum_{k \in M} \tau_{ij}^k - T_m^{Fim}(1 - x_{ij}) \qquad \forall (i,j) \in A_r \tag{1.13}$$

$$t_i^{chg} + TS_i \le t_i^{sai}, \; T_i^{Ini} \le t_i^{chg} \le T_i^{Fim} \qquad \forall i \in V_r \tag{1.14}$$

$$\sum_{(i,j) \in A_r} \sum_{k \in M} Cb_{ij}^{k,v}.d_{ij}^k + \sum_{(i,j) \in A_r} Cb_v^+.D_{ij}.f_{ij} = q_{comb} \le Cb^{Max} \tag{1.15}$$

$$T_k^{Fim} - \sum_{j \in V_r^+} \tau_{0j}^k \ge T_v^{Ini} - T_m^{Fim} + \sum_{j \in V_r^+} T_m^{Fim} x_{0j}^k \qquad \forall k \in M \tag{1.16}$$

$$t_j^{chg} - \tau_{0j}^k \ge T_v^{Ini} - T_m^{Fim} + T_m^{Fim} x_{0j}^k \qquad \forall k \in M, \forall j \in V_r^+ \tag{1.17}$$

$$T_k^{Ini} + \sum_{i \in V_r^+} \tau_{i0}^k \le T_v^{Fim} + T_m^{Fim} - \sum_{i \in V_r^+} T_m^{Fim} x_{i0}^k \qquad \forall k \in M \tag{1.18}$$

$$t_i^{sai} + \tau_{i0}^k \le T_v^{Fim} + T_m^{Fim} - T_m^{Fim} x_{i0}^k \qquad \forall k \in M, \forall i \in V_r^+ \tag{1.19}$$

$$x_{ij}^k \in \{0,1\}, \tau_{ij}^k \ge 0, d_{ij}^k \ge 0 \qquad \forall i,j \in V_r, \forall k \in M \tag{1.20}$$

$$x_{ij} \in \{0,1\}, t_i^{chg} \ge 0, t_i^{sai} \ge 0, f_{ij} \ge 0, \qquad \forall i,j \in V_r \tag{1.21}$$

As restrições 1.2 asseguram que o veículo visita todos os vértices da rota, garantindo que são selecionadas exatamente uma aresta saindo e uma aresta chegando em cada vértice.

As restrições 1.3 são inseridas no modelo para que as variáveis x_{ij}^k sejam positivas somente se as variáveis x_{ij} forem positivas.

As restrições 1.4 impedem a formação de sub-ciclos ao garantir que, se o veículo chega em um cliente j no período k, não pode sair deste mesmo cliente em um período anterior a k. Cada variável de distância é limitada superiormente ao valor da distância da respectiva aresta em 1.5. As restrições 1.6 controlam as variáveis d_{ij}^k para que a distância total percorrida na mesma aresta (i, j), em diferentes períodos, seja exatamente a distância da aresta (i, j).

Para cada cliente j, as restrições 1.7 garantem que o veículo, ao passar por j, tenha uma redução em sua carga (representada pelas variáveis f_{ij}) equivalente à demanda de j. Em 1.8, cada variável que representa a carga em um arco é limitada à demanda total da rota. Este mesmo conjunto de restrições, reduz a carga transportada a zero no arco (i, j), se o veículo não passar pelo respectivo arco, isso é, caso a variável que indica se o veículo percorre este arco seja nula.

As restrições 1.9 obtém o tempo de percurso de cada arco a partir da distância percorrida, considerando a velocidade estimada para o trecho no período indicado. O tempo de percurso de qualquer arco em um período k é limitado ao tempo total deste mesmo período, em 1.10.

As restrições 1.11 a 1.12 determinam o tempo de saída e de chegada do veículo nos clientes, t_i^{sai} e t_i^{chg}. Nas restrições 1.11, o tempo de saída do cliente i é limitado ao instante final do período k menos o tempo gasto para percorrer o arco, τ_{ij}^k. De forma similar, as restrições 1.12 limitam o tempo de chegada no cliente j ao valor do início do período k somado ao tempo gasto para percorrer o arco. Nos dois casos, a restrição só é imposta se o veículo sai (ou chega) em i no período k considerado (quando x_{ij}^k é nula, uma constante é somada ao lado direito para que a limitação não seja imposta).

Para todo arco (i, j) percorrido pelo veículo, as restrições 1.13 asseguram que a chegada a j ocorre a partir do momento correspondente ao tempo de saída de i mais o tempo gasto para percorrer o arco (i, j). Nas restrições 1.14, os tempos de saída e de chegada do veículo nos clientes, t_i^{sai} e t_i^{chg}, são forçados a respeitar o tempo de serviço no cliente i, TS_i e a janela de atendimento do cliente, de T_i^{Ini} a T_i^{Fim}.

A quantidade de combustível consumida pelo veículo q é calculada pela restrição 1.15. A primeira parcela é referente à distância percorrida, com base na estimativa de consumo de combustível $Cb_{ij}^{k,v}$ de cada arco (i, j) e período k. A segunda é referente ao peso transportado pelo veículo, considerando a distância D_{ij} do arco (i, j), o peso no veículo f_{ij} e o coeficiente Cb_v^+, que indica o consumo de uma unidade de combustível para uma unidade de carga. Na mesma restrição 1.15, a quantidade de combustível é limitada à capacidade do tanque.

As restrições 1.16 garantem que o horário de saída do veículo do depósito é maior ou igual do que o início da janela de tempo do veículo quando mais de um período de tempo são utilizados. O lado esquerdo, com somente um termo τ_{0j}^k diferente de zero, representa o instante de saída do depósito.

Quando o veículo utiliza somente um período para sair do depósito é necessário utilizar o tempo de chegada do primeiro cliente da rota. As restrições 1.17 asseguram que o horário de saída é maior que início da janela de tempo do veículo.

As restrições 1.18 garantem que o horário de chegada do veículo ao depósito é menor ou igual do que o final da janela de tempo do veículo quando mais de um período de tempo são utilizados. O lado esquerdo, com somente um termo τ_{i0}^k diferente de zero, representa o instante de chegada ao depósito.

Quando o veículo utiliza somente um período para chegar ao depósito é necessário utilizar o tempo de saída do cliente da rota. As restrições 1.19 asseguram que o horário de chegada é menor ou igual que final da janela de tempo do veículo.

A função objetivo minimiza o CO_2 produzido pelo veículo ao percorrer a rota, que é calculado pela taxa de conversão de combustível por CO_2 vezes o combustível.

O tempo de saída e chegada ao depósito são facilmente calculados após o resultado ser recuperado do resolvedor, pois, acrescentar isso ao modelo tornaria o mesmo mais complexo e suas respectivas variáveis não são usadas no modelo.

1.2 O algoritmo de *local branching*

Local branching (LB) é um algoritmo de busca local proposto por [Fischetti and Lodi, 2003], que utiliza um modelo de programação linear inteira (MIP) para encontrar novas soluções viáveis para determinado problema. Uma restrição é inserida no MIP para restringir o espaço de busca do problema a uma vizinhança em torno de uma solução de referência, usualmente, limitando o número de modificações que podem ser realizadas nesta solução.

$$\sum_{(i,j)\in A_r^*} x_{ij} \geq |A_r^*| - 3 \tag{1.22}$$

A restrição 1.22 de *local branching* permite que até três arestas (i, j) da rota original sejam substituídas durante a otimização. Para isso, a soma das variáveis x_{ij} correspondentes às arestas (i, j) da rota r, que inicialmente equivale ao número de arestas da rota, pode ter redução de até 3 unidades. Com a restrição, a rota original continua sendo viável, mas pode também ser substituída por uma rota de menor custo à medida que alguma aresta (i, j) deixe de ser utilizada (e a respectiva variável passe a valer 0).

Como todo algoritmo de busca local, LB redefine a vizinhança reiteradamente. A cada otimização em que uma nova solução é retornada, a solução de referência é substituída e a restrição de LB é atualizada. Eventualmente, quando a otimização não é capaz de encontrar uma nova solução, considera-se que o algoritmo atingiu um ótimo local.

Algoritmo 1.1: *Local Branching*

```
1   input: instancia, rota, numMaxIteracoes
2   output: rota
3   modelo ← criaModelo(instancia);
4   criaRestricaoLB(modelo, rota);
5   novaRota ← mip(modelo, rota);
6   while(novaRota != rota)
7       rota ← novaRota;
8       atualizaRestricaoLB(modelo, rota);
9       novaRota ← mip(modelo, rota);
10  end while
11  return rota;
```

O Algoritmo 1.1 ilustra o *local branching.* A função *criaModelo*, utilizada na linha 3, cria o modelo apresentado na Subseção 1.1, considerando apenas os pontos visitados pela respectiva rota. A função *criaRestricaoLB* insere a restrição de LB 1.22 tendo como referência a rota inicial. A cada nova rota encontrada pelo algoritmo, a restrição de LB é reescrita pela função *atualizaRestricaoLB*.

Na função *mip*, o modelo é otimizado até atingir um valor menor ou igual à 15% do *gap* de integralidade, porque, para atingir o ótimo, é necessário muito tempo de computação e pode não existir uma solução inteira com esse valor. Quando a solução é recuperada, verifica-se se a mesma é diferente da rota atual. Se sim, a restrição de LB é atualizada com a nova rota e o algoritmo continua. Caso contrário, a rota atual é considerada a melhor rota que o modelo foi capaz de encontrar na vizinhança definida. Neste caso, o algoritmo é encerrado, retornando a rota atual.

Por exemplo, considerando a rota $0 - a - b - c - 0$, o somatório do lado esquerdo incluiria as variáveis x_{0a}, x_{ab}, x_{bc} e x_{c0}, permitindo que até 3 fossem substituídas durante a otimização. Com isso, o modelo irá selecionar a opção de menor custo entre as rotas (0-b-a-c-0), (0-a-c-b-0), (0-b-c-a-0), (0-c-a-b-0) e a rota original (0-a-b-c-0).

O modelo, ao selecionar a melhor sequência vizinha a uma rota de referência, considera todas as possíveis combinações de paradas nas arestas, determinando instantes de partida e chegada do veículo. Toda rota gerada respeita as restrições de janela de tempo dos clientes, capacidade de combustível e janela de atendimento do veículo.

1.3 Extensão do modelo para duas rotas

No intuito de aumentar a vizinhança definida pela busca local, o modelo apresentado anteriormente foi adaptado para otimizar duas rotas simultaneamente. Neste caso, além de permitir uma permutação da sequência de clientes visitados em uma rota e determinação de novos instantes de partidas, chegadas e paradas, o modelo também explora a troca de clientes entre as rotas.

Pequenas adaptações são necessárias no modelo. Em relação às variáveis, todas recebem um índice adicional para identificar a respectiva rota. São criadas ainda variáveis adicionais para cada rota, incluindo arestas e vértices relativos aos clientes visitados na outra rota (ou seja, para cada rota, os índices i e j passam a considerar o conjunto de clientes de ambas as rotas).

Há também adaptações nas restrições do modelo. Um somatório relativo às rotas é incorporado às expressões da função objetivo e das restrições 1.2 de cada cliente. Todas as demais restrições, de 1.3 a 1.21, são duplicadas, isto é, todo este conjunto de restrições é criado separadamente para cada rota.

O algoritmo de *local branching* passa a receber um par de rotas da solução e, a cada passo, acrescenta uma restrição 1.22 para cada rota. Ao final, as rotas obtidas substituem as originais na solução heurística.

A Constantes e variáveis do modelo matemático

Tabela A.1: Parâmetros do modelo

Parâmetros	Descrição
V_r	Conjunto de pontos (depósito e clientes) da rota r
V_r^+	Conjunto de clientes da rota r (excluindo o depósito)
A_r	Conjunto de arestas do subgrafo induzido por V_r
A_r^*	Subconjunto de arestas de A_r composto pelas arestas originais da rota r
M	Conjunto dos períodos
i, j	Índices dos clientes
k, m	Índices dos períodos, sendo que m representa o último período
$D_{i,j}$	Distância entre aresta (i, j)
Dem_i	Demanda do cliente i
TS_i	Tempo de serviço do cliente i
$[T_i^{Ini}, T_i^{Fim}]$	Janela de tempo do cliente i
$[T_k^{Ini}, T_k^{Fim}]$	Início e fim do período k
$[T_v^{Ini}, T_v^{Fim}]$	Janela de tempo do veículo
Vel_{ij}^k	Velocidade do veículo do arco (i, j) no período k
Cb^{Max}	Capacidade máxima do tanque de combustível do veículo
v	Tipo de veículo v associado a rota r.
$Cb_{ij}^{k,v}$	Taxa de consumo de combustível ao percorrer o arco (i, j) no período k
Cb_v^+	Consumo adicional de combustível para uma unidade a mais de carga
$Tx_v^{CO_2}$	Taxa de conversão combustível / CO_2

Tabela A.2: Variáveis do modelo matemático que otimiza uma rota r

Variáveis	Descrição
x_{ij}	Indica se o arco (i, j) é percorrido ou não
x_{ij}^k	Indica se um arco (i, j) é percorrido em um período k
d_{ij}^k	Indica a distância percorrida do arco (i, j) no período k
τ_{ij}^k	Indica o tempo gasto no arco (i, j) no período k
t_i^{sai}	Indica o instante de tempo em que o veículo sai do cliente i
t_i^{chg}	Indica o instante de tempo em que o veículo chega no cliente i
f_{ij}	Indica a carga do veículo ao percorrer o arco (i, j)
q_{comb}	Indica a quantidade total de combustível utilizada na rota.

Bibliografia

M. Fischetti and A. Lodi. Local branching. *Mathematical programming*, 98(1):23–47, 2003.

Y. Xiao and A. Konak. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review*, 88:146–166, 2016.