

Formulação p -step para o problema de caminho mínimo com restrições de recursos

Júnior César Bonafim¹

Universidade Federal de São Carlos, São Carlos, SP

Pedro Munari²

Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, SP

Resumo. Neste artigo, aborda-se o problema de caminho elementar com restrições de recursos (RCESPP, do inglês *resource-constrained elementary shortest path problem*) que, além de formular diversas situações reais importantes, ocorre comumente como subproblema na resolução de outros problemas clássicos de otimização combinatória. Por exemplo, em formulações do roteamento de veículos cujas variáveis de decisão se relacionam a rotas factíveis, comumente resolvidos pela técnica de geração de colunas, tais rotas são obtidas pela resolução do RCESPP. Recentemente, foram introduzidas novas formulações para problemas de roteamento de veículos com base em variáveis que estão relacionadas a caminhos parciais de comprimento p , chamados p -steps. Tais formulações têm mostrado limitantes mais fortes em relação às formulações tradicionais com variáveis baseadas em um único arco e , portanto, têm o potencial de beneficiar métodos de solução que dependem dessa característica. O objetivo deste trabalho é propor formulações p -step para a modelagem do RCESPP, dado que ainda não foram usadas nesse contexto, explorando propriedades e características especiais do problema. Experimentos computacionais com instâncias da literatura foram realizados de modo a comparar o desempenho das formulações propostas em relação à formulação tradicional, considerando diferentes valores de p .

Palavras-chave. Caminho mínimo com restrições de recursos, p -step, otimização combinatória.

1 Introdução

O problema de caminho mínimo (SPP, do inglês *shortest path problem*) tem por objetivo determinar um caminho de custo mínimo entre dois nós s e t em um grafo dirigido. Denota-se por c_{ij} o custo de se percorrer um arco (i, j) do grafo. Quando esses custos não induzem ciclos com custos negativos, algoritmos de complexidade polinomial, e portanto eficientes, são comumente usados na resolução do SPP. No entanto, se há presença de ciclos com custos negativos, estratégias adicionais devem ser consideradas para garantir que cada nó seja visitado no máximo uma vez, conduzindo ao chamado problema de caminho mínimo elementar (ESPP, do inglês *elementary shortest path problem*). Uma variante do ESPP, o problema de caminho mínimo elementar com restrições de recursos (RCESPP, do inglês *resource-constrained elementary shortest path problem*), no qual um número L de recursos é consumido a cada arco ou nó percorrido, modela diversas situações práticas além de aparecer como subproblema na resolução de outros problemas de otimização combinatória.

¹junior.bonafim@fatec.sp.gov.br

²munari@dep.ufscar.br

Por exemplo, em formulações do problema de roteamento de veículos (VRP, do inglês *vehicle routing problem*) baseadas em variáveis de decisão que se relacionam a rotas factíveis, o RCESPP é usado para determinar as rotas factíveis junto a um método de geração de colunas.

O RCESPP é um problema NP-difícil [4]. Os principais métodos de solução recorrem a algoritmos baseados em programação dinâmica [1, 5, 9, 10]. Porém, o desempenho destes algoritmos é afetado de forma negativa pela existência de rotas longas, em geral ocasionadas por restrições de recursos fracas, como mostrado em [6] para o RCESPP com restrições de capacidade, em que um algoritmo *branch-and-cut* supera o desempenho de algoritmos baseados em programação dinâmica. Métodos de solução baseados em algoritmos *branch-and-cut* são fortemente afetados pela formulação utilizada, mais precisamente, por sua relaxação linear. Assim, a escolha da formulação tem impacto direto no desempenho deste tipo de método de solução.

Recentemente, foram introduzidas formulações generalizadas para o VRP, considerando suas variantes com restrições de capacidade e janelas de tempo [3, 7]. Tal formulação se apoia na construção de rotas usando caminhos parciais de comprimento p , isto é, que percorrem p arcos. Esses caminhos, denominados *p-steps*, são associados a variáveis binárias que determinam as rotas que compõem a solução. Para valores fixos e relativamente pequenos de p , a formulação *p-step* resulta em um modelo compacto, cuja relaxação linear pode ser mais forte que aquela de modelos compactos conhecidos na literatura. Para valores maiores de p , o método de geração de colunas deve ser utilizado já que a enumeração de todos os *p-steps* torna-se impraticável. Até o momento, não se tem conhecimento de nenhuma formulação *p-step* desenvolvida especificamente para o RCESPP. Assim, o objetivo deste trabalho é propor uma formulação *p-step* para o RCESPP, explorando características e propriedades específicas do problema, e analisar seu desempenho na forma compacta para pequenos valores de p .

O restante deste trabalho é organizado da seguinte maneira. A Seção 2 apresenta a formulação tradicional para o RCESPP, com variáveis baseadas em único arco. A Seção 3 introduz uma nova formulação *p-step* para o RCESPP. Os experimentos computacionais e considerações finais são apresentados nas Seções 4 e 5, respectivamente.

2 Formulação clássica para o RCESPP

Considere um grafo dirigido $G = (N, A)$ em que $N = \{0, \dots, n+1\}$ é o conjunto dos nós, sendo que 0 e $n+1$ representam o mesmo nó inicial ou depósito; $N' = \{1, \dots, n\}$ é o conjunto de clientes; e $A = \{(i, j); i, j \in N, i \neq j, i \neq n+1, j \neq 0\}$ é o conjunto de arcos. Denota-se por $c_{ij} \geq 0$ o custo para se percorrer o arco $(i, j) \in A$. Adicionalmente, seja $\delta^-(i)$ o conjunto de arcos que chegam ao nó $i \in N$ e $\delta^+(i)$ o conjunto de arcos que partem do nó $i \in N$. Considerando 0 como o nó de origem e $n+1$ como o nó de destino, seja x_{ij} a variável binária que assume valor 1 se, e somente se o arco $(i, j) \in A$ é percorrido. Comumente, o RCESPP é formulado de modo geral por meio do seguinte problema de problema linear inteira [8]:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.a.} \quad \sum_{(i,j) \in \delta^+(i)} x_{ij} - \sum_{(j,i) \in \delta^-(i)} x_{ji} = \begin{cases} 1 & , \text{ se } i = 0 \\ -1 & , \text{ se } i = n+1 \\ 0 & , \text{ caso contrário} \end{cases}, \quad i \in N \tag{2}$$

$$\sum_{(i,j) \in A} w_{ij}^h x_{ij} \leq W^h, \quad h \in L \tag{3}$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A \tag{4}$$

A função objetivo (1) busca minimizar o custo total do caminho. As restrições (2) asseguram o fluxo adequado entre os nós, garantindo que o caminho inicie em 0, termine em $n + 1$ e que o fluxo de entrada seja igual ao fluxo de saída em cada cada nó. Observe que não é exigido que todo nó seja visitado (exceto os nós 0 e $n + 1$). As restrições (3) controlam o consumo do recurso $h \in L$ a cada arco percorrido, de modo que seja limitado à disponibilidade W^h deste recurso. O consumo de recursos por arco percorrido pode ser facilmente formulado como consumo por nó visitado, caso necessário. O domínio das variáveis é definido em (4). É necessário incluir estratégias para eliminação de sub-rotas e garantir elementaridade da solução. Tais estratégias em geral estão associadas à estrutura dos recursos envolvidos, podendo estar inclusas em (3).

Dois recursos comuns para o RCESPP, advindos de seu uso como subproblema na resolução do VRP, são capacidade e janelas de tempo. Estes recursos serão abordados neste trabalho. Para este fim, considere $q_i > 0$ a demanda para cada nó $i \in N$ e Q a capacidade do veículo que percorrerá o grafo G . Assume-se que $q_i \leq Q$ para todo $i \in N$. Adicionalmente, considere $[\omega_i^a, \omega_i^b]$ a janela de tempo para cada nó $i \in N$, t_{ij} o tempo de percurso para o arco $(i, j) \in A$ e s_i o tempo de serviço em $i \in N$. Definindo a variável contínua $\varphi_j \geq 0$ que representa o momento de início de serviço em $j \in N$, o modelo a seguir é uma formulação para o RCESPP com restrições de capacidade e janela de tempo:

$$\min \sum_{(i,j) \in A} c_{ij}x_{ij} \tag{5}$$

$$\text{s.a} \quad \sum_{(i,j) \in \delta^+(i)} x_{ij} - \sum_{(j,i) \in \delta^-(i)} x_{ji} = \begin{cases} 1 & , \text{ se } i = 0 \\ -1 & , \text{ se } i = n + 1 \\ 0 & , \text{ caso contrário} \end{cases} \quad i \in N \tag{6}$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} \leq 1, \quad i \in N \tag{7}$$

$$\varphi_j \geq \varphi_i + s_i + t_{ij} - M_{ij}(1 - x_{ij}), \quad (i, j) \in A \tag{8}$$

$$\omega_i^a \leq \varphi_i \leq \omega_i^b, \quad i \in N \tag{9}$$

$$\sum_{i \in N} q_i \sum_{(i,j) \in \delta^+(i)} x_{ij} \leq Q \tag{10}$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A \tag{11}$$

Como anteriormente, a função objetivo (11) minimiza o custo do caminho, enquanto (6) assegura o fluxo nos nós. As expressões (7) garantem a elementaridade do caminho. As restrições (8) controlam o tempo transcorrido durante a execução da rota e, além disso, previnem sub-rotas, sendo que M_{ij} é um parâmetro suficientemente grande que pode ser definido como $M_{ij} = \max\{\omega_i^b + s_i + t_{ij} - \omega_j^a, 0\}$. As expressões (9) e (10) asseguram a factibilidade do caminho quanto a janelas de tempo e capacidade, respectivamente.

Outros recursos podem ser facilmente modelados e incluídos, vantagem importante da abordagem via modelos de programação inteira mista para o RCESPP em relação às demais, como ressalta [2].

3 Formulação *p-step* para o RCESPP

Nesta seção, são propostas formulações *p-step* para o RCESPP, considerando os recursos de capacidade e janelas de tempo como anteriormente. Mantendo a notação fixada na Seção 2, um *p-step* r é um caminho elementar que: (a) percorre exatamente p arcos; ou (b) se inicia em 0 e percorre no máximo p arcos. Um *p-step* r é dito factível se o consumo dos recursos disponíveis não

viola suas respectivas capacidades (por exemplo, a soma das demandas dos nós visitados por r não viola a capacidade Q). Dois p -steps podem ser concatenados quando o último nó de um é igual ao primeiro nó do outro. Além disso, 0 deve ocorrer apenas como nó inicial e $n + 1$ como nó final dos p -steps. A Figura 1 exemplifica a construção de um caminho iniciando em 0 e retornando ao mesmo nó ($n + 1$) para $p = 3$ cujos p -steps utilizados são $0 - 2 - 3$, $3 - 5 - 6 - 7$ e $7 - 8 - 9 - 11$.

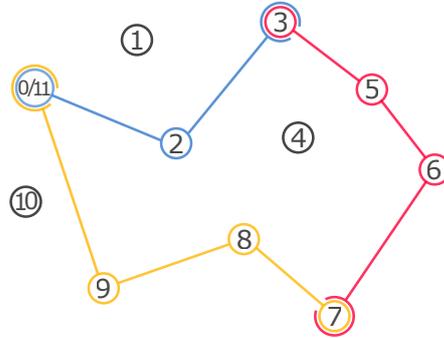


Figura 1: Exemplo de caminho elementar via p -steps com $p = 3$.

Seja R o conjunto dos p -steps factíveis; x_r a variável binária que assume valor 1 se, e somente se, o p -step $r \in R$ é percorrido no caminho ótimo; e $\varphi_j \geq 0$ a variável contínua que representa o acúmulo de tempo pelo caminho até o nó $j \in N$ (inclusive). O modelo a seguir é formulação p -step proposta para o RCESPP considerando capacidade e janelas de tempo como recursos:

$$\min \sum_{r \in R} c_r x_r \tag{12}$$

$$\text{s.a.} \sum_{r \in R} a_r^i x_r = \begin{cases} 1 & , \text{ se } i = 0 \\ -1 & , \text{ se } i = n + 1 \\ 0 & , \text{ caso contrário} \end{cases}, \quad i \in N \tag{13}$$

$$\sum_{r \in R} e_r^i x_r \leq 1, \quad i \in N' \tag{14}$$

$$\varphi_j \geq \varphi_i + s_i + t_{ij} - M_{ij} (1 - \sum_{r \in R_{ij}} x_r), \quad (i, j) \in A \tag{15}$$

$$\omega_i^a \leq \varphi_i \leq \omega_i^b, \quad i \in N \tag{16}$$

$$\sum_{i \in N \setminus \{n+1\}} q_i \sum_{r \in R} e_r^i x_r \leq Q \tag{17}$$

$$x_r \in \{0, 1\}, \quad r \in R \tag{18}$$

em que c_r representa o custo total de se percorrer os arcos do p -step r , e os parâmetros a_r e e_r são definidos como:

$$a_r^i = \begin{cases} 1, & \text{ se } i \text{ é o primeiro nó visitado pelo } p\text{-step } r \\ -1, & \text{ se } i \text{ é o último nó visitado pelo } p\text{-step } r \\ 0, & \text{ caso contrário} \end{cases} \tag{19}$$

$$e_r^i = \begin{cases} 1, & \text{ se } i \text{ é visitado pelo } p\text{-step } r \text{ mas não é o último nó de } r \\ 0, & \text{ caso contrário} \end{cases} \tag{20}$$

A função objetivo (12) minimiza o custo do caminho. As restrições (13) impõem que os p -steps sejam concatenados corretamente de acordo com nós iniciais e finais de cada um deles enquanto (14) asseguram que cada nó seja visitado no máximo uma vez. As restrições (15) garantem que o instante de início de serviço em um nó j seja maior ou igual ao instante de chegada no nó i imediatamente anterior somado do tempo de serviço em i e do tempo de trajeto no arco $(i, j) \in A$ em que M_{ij} é uma constante suficientemente grande definida como anteriormente. Adicionalmente, (16) asseguram que as janelas de tempo sejam respeitadas. A restrição (17) garante que a capacidade não seja excedida e (18) define o domínio da variável x_r .

Proposição 3.1. *A formulação p -step (12)-(18) é equivalente à formulação (5)-(11) para $p = 1$.*

Demonstração. Para $p = 1$, o conjunto dos p -steps factíveis R é constituído pelos arcos $(i, j) \in A$. Dessa forma, as variáveis λ_r associadas aos 1-steps de R podem ser reescritas como $\lambda_r = x_{i_r, j_r}$, em que (i_r, j_r) é o arco cujo 1-step r atravessa. Reescrevendo o modelo (12)-(18) com as variáveis x_{i_r, j_r} obtém-se a formulação (5)-(11). \square

Assim, pela proposição anterior, resultados computacionais para a formulação p -step (12)-(18) não serão apresentados para $p = 1$.

4 Experimentos computacionais

Como forma de avaliar a formulação p -step (12)-(18) proposta para o RCESPP e comparar seus resultados com a formulação tradicional (5)-(11), foram conduzidos experimentos computacionais com instâncias que possibilitam a utilização da formulação p -step compacta, para pequenos valores de p . As instâncias usadas são adaptações das instâncias de Solomon [11], assim como em outros trabalhos abordando o RCESPP [2, 5, 10], e são classificadas em três classes: instâncias do tipo R, em que a localização de cada cliente é definida aleatoriamente; instâncias do tipo C, com alocação dos clientes em *clusters*; e instâncias do tipo RC, com localização mista dos clientes, sendo algumas aleatórias e outras em *clusters*. O custo do arco $(i, j) \in A$ é definido como $c_{ij} = d_{ij} - \alpha(i)$ em que d_{ij} é a distância Euclidiana entre os nós i e j definida originalmente na instância, enquanto $\alpha(i)$ é número inteiro gerado usando uma distribuição uniforme em $\{1, \dots, 20\}$, de forma a garantir uma quantidade razoável de arcos com custos negativos. Essa é a única adaptação realizada nas instâncias de Solomon. O número de clientes de cada instância foi limitado a 25 como forma de garantir a construção do modelo compacto de (12)-(18) para valores de p de 2 a 4.

Os modelos foram implementados em linguagem C++, usando-se a biblioteca Concert do *solver* IBM CPLEX Optimization Studio v. 12.10, e resolvidos por meio do método *branch-and-cut* de propósito geral deste mesmo *solver*, usando-se suas configurações padrão e tempo limite de 1 hora. Todos os experimentos foram executados em um computador com processador Intel(R) Core(TM) i7-8700K 3.70GHz, 62GB de memória RAM e sistema operacional Linux Mint 20.1 Cinnamon.

A Tabela 1 apresenta os valores médios para a relaxação linear dos dois modelos e a diferença relativa entre a formulação p -step (12)-(18) e o modelo (5)-(11), por classe de instância. Observa-se vantagem da formulação p -step que é maior nas instâncias da classe C. Os tempos médios para a resolução da relaxação linear são apresentados na Tabela 2, em que verifica-se o crescimento do tempo com os valores de p . Tal comportamento se mantém nos tempos médios para a resolução dos modelos, mostrados na Tabela 3. A Tabela 4 traz as instâncias em que a solução ótima não pode ser alcançada para alguma das formulações no tempo limite. Os valores $p = 2$ e $p = 4$ foram escolhidos para compor a tabela pois são extremos dos valores testados. Para cada modelo, as colunas mostram o limitante inferior (LB), o limitante superior (UB), o gap relativo (GAP) calculado como $|UB - LB|/|LB|$, e o tempo computacional em segundos. É possível observar a semelhança entre os limitantes da formulação (5)-(11) e o modelo (12)-(18) para $p = 2$. A diferença

entre os limitantes cresce com p evidenciando a dificuldade na resolução do modelo quando o número de p -steps se eleva.

Tabela 1: Valor ótimo médio da relaxação linear dos modelos.

Classe	Modelo (5)-(11)	Modelo p -step (12)-(18)		
		$p = 2$	$p = 3$	$p = 4$
R	-53,517	-45,212 (15,52%)	-42,802 (20,02%)	-41,219 (22,98%)
C	-136,515	-134,052 (1,80%)	-130,214 (4,62%)	-127,608 (6,52%)
RC	-151,212	-146,243 (3,29%)	-142,520 (5,75%)	-140,461 (7,11%)

Tabela 2: Tempos médios para a resolução da relaxação linear dos modelos, em segundos.

Classe	Modelo (5)-(11)	Modelo p -step (12)-(18)		
		$p = 2$	$p = 3$	$p = 4$
Instâncias R	0,004	0,095	4,040	139,955
Instâncias C	0,004	0,114	5,789	140,458
Instâncias RC	0,004	0,099	4,699	130,146

Tabela 3: Tempos médios para a resolução dos modelos, em segundos.

Classe	Modelo (5)-(11)	Modelo p -step (12)-(18)		
		$p = 2$	$p = 3$	$p = 4$
Instâncias R	0,511	0,752	46,916	1133,797
Instâncias C	148,132	260,543	799,253	1147,705
Instâncias RC	931,110	980,975	1537,823	2297,064

Tabela 4: Instâncias não resolvidas.

Instância	Modelo (5)-(11)				Modelo p -step (12)-(18)							
					$p = 2$				$p = 4$			
	LB	UB	GAP	Tempo	LB	UB	GAP	Tempo	LB	UB	GAP	Tempo
C104	-55,80	-55,80	-	506,90	-55,80	-55,80	-	1676,88	-91,81	-54,60	40,53%	3600,00
C204	-84,60	-84,60	-	1561,51	-84,60	-84,60	-	2180,84	-104,81	-84,60	19,28%	3600,00
RC202	-26,10	-26,10	-	25,69	-26,10	-26,10	-	187,26	-74,45	-26,10	64,94%	3600,00
RC203	-43,51	-15,70	63,91%	3600,00	-53,72	-15,70	70,77%	3600,00	-129,63	-8,00	93,83%	3600,00
RC204	-151,70	-50,20	66,91%	3600,00	-156,17	-38,80	75,15%	3600,00	-170,82	-33,60	80,33%	3600,00
RC205	-3,80	-3,80	-	207,58	-3,80	-3,80	-	785,63	-100,89	-1,00	99,01%	3600,00
RC207	-120,72	-68,80	43,01%	3600,00	-123,94	-68,80	44,49%	3600,00	-189,58	-46,80	75,31%	3600,00
RC208	-116,04	-11,00	90,52%	3600,00	-120,54	-11,00	90,87%	3600,00	-134,65	-11,00	91,83%	3600,00

5 Conclusões

O RCESPP é um problema importante no contexto de roteamento de veículos cujos métodos de solução exatos presentes na literatura são em geral baseados em algoritmos de programação dinâmica, sendo então pouco explorado do ponto de vista de modelos de programação inteira mista. Este trabalho apresentou uma formulação p -step para o RCESPP e realizou experimentos computacionais com o modelo compacto para pequenos valores de p que demonstraram que melhores limitantes são obtidos a partir desta formulação em relação à formulação padrão baseada em arcos a medida que os valores de p crescem, o que mostra o potencial desta formulação no desenvolvimento de métodos de solução exatos. Soluções ótimas também foram obtidas para a maioria das instâncias testadas. Os tempos de processamento, tanto para, relaxação linear e solução inteira crescem com o valor de p , evidenciando um possível *trade-off*. Pesquisas futuras prevêem a incorporação de desigualdades válidas ao modelo, comparação com outras formulações

e desenvolvimento de um método de geração de colunas para a forma não compacta do modelo, que permita aplicação para instâncias e valores de p maiores.

Referências

- [1] Boland, N., Dethridge, J., Dumitrescu, I. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34:58-68, 2006. DOI: 10.1016/j.orl.2004.11.011.
- [2] Da, J., Zheng, L., Tang, X. A polyhedral study of the elementary shortest path problem with resource constraints. *Lecture Notes in Computer Science*, 10572 LNCS:79-93, 2017. DOI: 10.1007/978-3-319-68496-3.6.
- [3] Dollevoet, T., Munari, P., Spliet, R. A p-step Formulation for the Capacitated Vehicle Routing Problem. n. 2014, 2020.
- [4] Dror, M. Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW. *Operations Research*, 42:977-978, 1994. DOI: 10.1287/opre.42.5.977.
- [5] Feillet, D. et al. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216-229, 2004. DOI: 10.1002/net.20033.
- [6] Jepsen, M., Petersen, B., Spoorendonk, S. A branch-and-cut algorithm for the elementary shortest path problem with a capacity constraint. Department of Computer Science, University of Copenhagen, n. 08/01, 2008.
- [7] Munari, P., Dollevoet, T., Spliet, R. A generalized formulation for vehicle routing problems. n. 1, p. 1-19, 2017.
- [8] Pugliese, L. D. P., Guerriero, F. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62:183-200, 2013. DOI: 10.1002/net.21511.
- [9] Righini, G., Salani, M. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3:255-273, 2006. DOI: 10.1016/j.disopt.2006.05.007.
- [10] Righini, G., Salani, M. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51:155-170, 2008. DOI: 10.1002/net.20212.
- [11] Solomon, M. M. Vehicle routing and scheduling with time window constraints: Models and Algorithms. Tese de Doutorado, University of Pennsylvania, 1983.
- [12] Toth, P., Vigo, D. *Vehicle routing: problems, methods, and applications*. 3a edição. Society for Industrial and Applied Mathematics, Philadelphia, 2014.