O Problema do Caminho Elementar Mínimo com Restrições de Recursos e Subcaminhos Proibidos

Paulo Cezar Costa*
Humberto Longo†
Instituto de Informática
Universidade Federal de Goiás
Goiânia - GO

Resumo

O trabalho descreve uma variante do Problema do Caminho Mínimo que considera restrições de recursos e de subcaminhos proibidos. Este problema tem aplicações em situações práticas, em especial a programação de veículos para realizar coletas ou entregas de mercadorias em diversos pontos de uma cidade. O problema foi modelado como um problema de programação linear inteira, e o modelo resultante foi testado com o uso das ferramentas *Ilog CPLEX Optimizer* e *Gurobi Optimizer*. Também são apresentadas outras possíveis abordagens para a solução do problema.

Palavras-Chave: caminho mínimo, restrições topológicas, restrições de recursos, subcaminhos proibidos.

1 Introdução

Um grafo é basicamente um conjunto de elementos, que podem representar as mais diversas entidades, e conexões entre esses elementos, representando as relações existentes entre as entidades. Esses elementos são chamados de vértices e as conexões de arestas. Caso as relações entre as entidades no grafo não sejam simétricas, chama-se o mesmo de grafo direcionado, ou dígrafo, e as arestas de arcos. Nesse caso os arcos indicam a direção em que a relação ocorre. Grafos são muito úteis na representação de problemas nas mais diversas áreas, podendo modelar de maneira muito intuitiva malhas viárias e redes telefônicas por exemplo.

No Problema do Caminho Mínimo (*SPP - Shortest Path Problem*), provavelmente um dos mais conhecidos na Teoria dos Grafos, deseja-se encontrar um caminho entre dois vértices dados, de modo que a soma dos valores associados às arestas que o constituem

*Orientando – `paulocosta@inf.ufg.br`
†Orientador – `longo@inf.ufg.br`
[1]Revisado pelo orientador.

seja a menor possível. Esses valores podem ter vários significados, como o tempo que se leva para percorrer tal caminho, a quantidade de vértices no mesmo ou a distância total. O problema tem uma grande variedade de aplicações, as mais óbvias aparecendo em áreas como transportes e telecomunicações.

O SPP tem sido estudado por um longo período de tempo, o que possibilitou o surgimento de técnicas eficientes que o solucionam, como os algortimos de Dijkstra [9], Bellman-Ford [17, 5] e Floyd-Warshall [12], os quais têm complexidade de tempo polinomial no número de vértices e arestas. Contudo, modificações aparentemente simples na estrutura do problema podem aumentar bastante a dificuldade de resolução do mesmo, geralmente transformando-o em um problema $\mathcal{NP}$-Difícil. Por exemplo, restrições topológicas, ao restringir a combinação de vértices ou arestas, limitam a maneira que os caminhos podem ser compostos; enquanto restrições de recursos restringem os caminhos quando são associados recursos ao grafo e limites são estabelecidos sobre os níveis que os mesmos podem assumir.

A adição de restrições de recursos ao SPP gera variantes como o Problema do Caminho Mínimo com Restrição de Capacidade (*SPPCC - Shortest Path Problem with a Capacity Constraint*), onde são associados valores aos vértices e a soma dos valores associados aos vértices que fazem parte da solução não podem ser superiores a um certo limite; ou o Problema do Caminho Mínimo com Janelas de Tempo (*SPPTW - Shortest Path Problem with Time Windows*), no qual é definido o tempo gasto para percorrer cada arco e para cada vértice é estipulado um limite superior e inferior de quando o mesmo pode vir a fazer parte de uma solução.

Esses problemas (SPPCC e SPPTW) são casos particulares do Problema do Caminho Elementar Mínimo com Restrições de Recursos (*ESPPRC - Elementary Shortest Path Problem with Resource Constraints*), o qual consiste em encontrar, dentre todos os caminhos de um vértice origem s a um vértice destino t, um caminho elementar, ou seja, um caminho no qual cada vértice é visitado apenas uma vez. Além disso, o custo desse caminho deve ser mínimo e o mesmo deve satisfazer um conjunto de restrições definidas sobre um conjunto de recursos.

Nesse contexto, um recurso corresponde a uma quantidade, como o tempo ou a carga de um veículo, que varia ao longo de um caminho de acordo com funções chamadas Funções de Extensão de Recursos (*REF - Resource Extension Functions* [8]). Uma REF é definida para cada arco (u,v) no grafo sobre cada recurso considerado. Uma tal função limita inferiormente o nível que o recurso correspondente pode ter no vértice v, dado o nível do recurso no vértice u. Assim, as restrições de recursos são dadas como intervalos, que correspondem a níveis mínimo e máximo de variação dos recursos em cada vértice do grafo para cada recurso considerado. Os níveis de variação devem ser respeitados ao longo dos caminhos.

O ESPPRC é um problema fortemente $\mathcal{NP}$-Difícil, provado por Dror [10], e tem sido estudado por vários autores [7, 11, 16]. Até o momento, a maioria das soluções propostas são baseadas em algoritmos pseudo-polinomiais de correção de rótulos. Righini e Salani [21, 20] propõem algoritmos de programação dinâmica para o ESPPRC que fazem uso de técnicas avançadas, tais como técnicas de programação dinâmica bidirecional e relaxação decrescente do espaço de estados (*DSSR - Decremental State Space Relaxation*).

A adição de restrições topológicas ao SPP gera variantes como o Problema do Caminho Mínimo com Pares Impossíveis (*SPPIP - Shortest Path Problem with Impossible Pairs*), onde para alguns pares de vértices apenas um deles pode fazer parte da solução; ou o Problema do Caminho Mínimo com Subcaminhos Proibidos (*SPPFP - Shortest Path Problem with Forbidden Paths*), em que alguns trechos de caminhos não podem estar presentes na solução.

O Problema do Caminho Elementar Mínimo com Subcaminhos Proibidos (*ESPPFP - Elementary Shortest Path Problem with Forbidden Paths*) é uma variante do problema de caminho mínimo com restrições no qual as restrições adicionais vem em forma de um conjunto de caminhos (sequências de arcos) proibidos, que não podem fazer parte de nenhuma solução viável. Ou seja, o problema consiste em encontrar um caminho de custo mínimo de um vértice origem s ao vértice destino t, sem que algum caminho proibido seja percorrido durante o processo.

Szeider [22] provou que o problema de encontrar um caminho mínimo simples com restrição de subcaminhos é $\mathcal{NP}$-Difícil para determinadas classes de grafos, mesmo considerando caminhos proibidos pequenos. Se todos os caminhos proibidos têm tamanho menor ou igual a dois, o ESPPFP pode ser resolvido em tempo polinomial através de reduções para o SPP, como descritas por Giesel [14]. Embora com uma quantidade considerável de pesquisas a seu respeito, o ESPPFP ainda necessita de estudos. Villeneuve e Desaulniers [23] propuseram uma solução para o ESPPFP onde o algoritmo de reconhecimento de padrões de Aho e Corasick [4] é usado para filtrar os caminhos produzidos por um algoritmo de caminho mínimo (para o SPP). Se o caminho é aceito pelo filtro, este não possui trechos proibidos e é válido. Caso não seja aceito, o mesmo possui algum subcaminho proibido sendo tal trecho identificado pelo algoritmo de Aho e Corasick. Para contornar este caminho inválido o algoritmo de Villeneuve e Desaulniers utiliza a abordagem de desvio de caminho de Martins [18] para construir uma rota alternativa.

Uma outra solução, proposta por Ahmed e Lubiw [3], também utiliza a abordagem de Martins para não percorrer um caminho proibido quando este é encontrado em um caminho mínimo. Entretanto, diferente da solução de Villeneuve e Desaulniers, a solução de Ahmed e Lubiw não faz uso de um conjunto de caminhos proibidos previamente conhecido. Para que sejam encontrados tais caminhos esta solução utiliza a técnica de tentativa-e-erro: quando uma tentativa de percorrer um caminho acarreta em algum erro, tal caminho é tratado como caminho proibido e um desvio é criado.

Neste projeto foi abordada uma variação do problema do caminho mínimo com restrições, discutida pela primeira vez por Foulds e Longo [13], o Problema do Caminho Elementar Mínimo com Restrições de Recursos e Subcaminhos Proibidos (*ESPPRCFP - Elementary Shortest Path Problem with Resource Constraints and Forbidden Paths*). Este problema mescla restrições dos dois problemas, ESPPRC e ESPPFP, descritos anteriormente:

(i) Cada arco tem associado a si o consumo de vários recursos, os quais devem ser acumulados ao longo do caminho e, para cada recurso, os níveis do mesmo em cada vértice devem ser respeitados;

(ii) Há um conjunto finito de subcaminhos, dos quais nenhum deve fazer parte de uma solução para o problema.

2 Motivação

O estudo do ESPPRCFP foi motivado principalmente pela possibilidade de sua aplicação em certas situações práticas, como a programação de veículos para realizar coletas ou entregas de mercadorias em diversos pontos de uma cidade.

Esse cenário é tipicamente enfrentado por gerentes de cadeias de abastecimento ao planejar rotas de veículos de entrega de suprimentos. Geralmente, estes veículos fazem viagens de longa distância atendendo várias cidades e em cada cidade devem entregar vários tipos diferentes de suprimentos para bares, restaurantes, supermercados e outros pontos de varejo. Ao chegar numa cidade o veículo deve fazer entregas em um número de locais ao menor custo possível e sujeito a várias restrições.

Cada tipo de suprimento, por exemplo, está sujeito a uma restrição de quantidade. Desta forma, os suprimentos podem ser representados por vetores de recursos multidimensionais, que são acumulados ao longo do caminho de visita aos clientes pelo veículo de entrega. A quantidade total de cada suprimento entregue deve estar sempre entre limites inferior e superior dados. Além disso, o caminho por onde o veículo pode passar na cidade é restrito, uma vez que certas sequências de ruas não podem ser usadas por tais veículos. Por exemplo, sequências que implicam numa conversão proibida ou ruas estreitas por onde o veículo não é capaz de passar.

Ao terminar as entregas agendadas, com o menor custo possível e sujeito às restrições mencionadas, o veículo deixa a cidade e retorna à rodovia. Como o ponto de saída da cidade geralmente não coincide com o ponto de entrada, o problema acaba se tornando uma questão de encontrar um caminho de custo mínimo entre um acesso e uma saída da cidade, sujeito a restrições de recurso e subcaminhos proibidos.

Outra motivação para o estudo do problema é a possibilidade futura de utilização do mesmo durante a geração de colunas em algoritmos *Branch-and-Price* [8] voltados à solução de Problemas de Roteamento de Veículos (*VRP - Vehicle Routing Problems*).

Como exposto por Pecin [19], a natureza combinatorial dos VRPs sugere que os mesmos sejam formulados e resolvidos como um problema de Programação Linear Inteira (PLI). Muitos métodos exatos para PLIs baseiam-se no paradigma *Branch-and-Bound*, no qual a busca de uma solução ótima é estruturada em uma árvore, onde cada nó representa uma parte do espaço de soluções. Relaxações do problema podem ser usadas em cada nó da árvore *Branch-and-Bound* para o cálculo de limites inferiores (ou superiores no caso de maximização) para o valor de uma solução ótima. Assim, se o limite inferior para algum nó da árvore for maior que o menor limite superior conhecido (ou seja, a melhor solução viável encontrada até o momento), então esse nó pode ser seguramente descartado da busca.

Quando a formulação para um problema de PLI é bem estruturada, isto é, quando há determinados conjuntos de variáveis onde algumas restrições não são sobrepostas, é possível usar a técnica de decomposição de Dantzig-Wolfe [6] para decompor o problema original em subproblemas menores, e combinar suas soluções em um problema mestre. O problema mestre é, em geral, pequeno no número de restrições, mas pode ter um grande número de variáveis. Uma relaxação linear do problema mestre é uma forma diferente de calcular o limite inferior, podendo ser, em muitos casos, um limite melhor que o obtido com a relaxação linear do problema original. Para calcular este limite o subproblema deve ser resolvido, iterativamente, até encontrar o valor ótimo da função objetivo do problema mestre. Esse processo é conhecido como geração de colunas, uma vez que a solução dos subproblemas dá origem a novas colunas no problema mestre e quando o mesmo é inserido em um algoritmo *Branch-and-Bound*, tem-se um algoritmo *Branch-and-Price*.

3 Modelagem

Nesta seção é feita uma descrição mais formal de como os problemas apresentados anteriormente podem ser modelados em grafos. Nas subseções 3.1 e 3.2 são descritas modelagens para o ESPPRC e ESPPFP, respectivamente. Na subseção 3.3 são descritas modelagens para o ESPPRCFP.

As subseções seguintes fazem uso de um dígrafo simples $G = (V, A)$, onde $V = \{v_1, v_2, ..., v_n\}$ é o conjunto de vértices e inclui um vértice origem s e um vértice destino

t, e $A = \{(v_i, v_j) \mid v_i, v_j \in V\}$ é o conjunto de arcos. A função $c : A \to \mathbb{R}$ define os custos dos arcos e c_{ij} representa o custo do arco (v_i, v_j). Um caminho em G contém pelo menos um arco e seu custo é dado pela soma dos custos dos arcos que o constituem.

3.1 ESPPRC

O objetivo do ESPPRC é encontrar em G um caminho elementar de custo mínimo, de um vértice origem s a um vértice destino t, que satisfaça um conjunto de restrições de recursos ao longo de seu trajeto. Dados R recursos, a REF $d^r : A \to \mathbb{R}$ define o consumo do recurso r ($1 \leq r \leq R$) associado a cada arco de G. A notação d^r_{ij} representa o consumo do recurso r no arco (v_i, v_j). Assume-se que os valores d^r_{ij} satisfazem a desigualdade triangular para todo recurso r, ou seja, $d^r_{ij} < d^r_{ik} + d^r_{kj}$, $\forall\ v_i, v_j, v_k \in V$. Para cada vértice $v_i \in V$ e recurso r, associam-se dois valores não negativos a^r_i e b^r_i, tal que o consumo acumulado do recurso r ao longo de um caminho de s para v_i deve pertencer ao intervalo $[a^r_i, b^r_i]$.

3.2 ESPPFP

No ESPPFP o objetivo é encontrar em G um caminho elementar de custo mínimo, de um vértice origem s a um vértice destino t, sob a restrição de que nenhum subcaminho proibido seja parte do mesmo. O conjunto F de subcaminhos proibidos em G é definido por sequências não-vazias de arcos $(v_{(1)}, v_{(2)})$, $(v_{(2)}, v_{(3)})$, ..., $(v_{(k-1)}, v_{(k)})$, com $1 < k \leq |V|$ e $v_{(i)} \in V$, e cada uma define um caminho de um vértice $v_{(1)}$ a um vértice $v_{(k)}$.

3.3 ESPPRCFP

No ESPPRCFP, além dos R recursos existentes no ESPPRC (seção 3.1) e consumos d^r_{ij} associados, é dado um conjunto F de subcaminhos proibidos como no ESPPFP (seção 3.2). O objetivo do ESPPRCFP é encontrar em G um caminho elementar de custo mínimo, de um vértice origem s a um vértice destino t que satisfaça todas as restrições de recursos ao longo de seu trajeto e tal que nenhum $f \in F$ seja parte do mesmo.

3.3.1 Modelagem PLI

O ESPPRCFP também pode ser modelado como um problema de PLI, como descrito nas equações (1) a (6). Neste modelo cada arco $(v_i, v_j) \in A$ é associado a uma variável binária x_{ij} que indica se o mesmo faz ou não parte de um caminho mínimo viável. A variável t^r_i corresponde ao consumo acumulado do recurso r num caminho da origem s ao vértice v_i. As constantes d^r_{ij} e c_{ij} representam, respectivamente, o consumo do recurso r

no arco (v_i, v_j) e o custo do mesmo, enquanto as constantes a_i^r e b_i^r representam os limites inferior e superior aceitáveis de consumo do recurso r ao chegar no vértice v_i.

$$
\begin{aligned}
\text{Minimizar} \qquad & Z = \sum_{(v_i,v_j)\in A} c_{ij}x_{ij}; && (1)\\
\text{sujeito a} \quad & \sum_{(v_j,v_i)\in A} x_{ji} - \sum_{(v_i,v_k)\in A} x_{ik} = \begin{cases} -1, & i = s, \\ 0, & i \neq s, \\ 1, & i = t; \end{cases} && (2)\\
& t_i^r + d_{ij}^r - t_j^r + Mx_{ij} \leq M, \qquad r = 1, 2, \ldots, R; && (3)\\
& \qquad\qquad\qquad\qquad\qquad\quad (v_i, v_j) \in A; &&\\
& a_i^r \leq t_i^r \leq b_i^r \qquad r = 1, 2, \ldots, R; && (4)\\
& \qquad\qquad\qquad\qquad\qquad\quad v_i \in V; &&\\
& \sum_{(v_i,v_j)\in f} x_{ij} \leq |f| - 1, \qquad \forall\, f \in F; && (5)\\
& x_{ij} \in \{0, 1\}, \qquad (v_i, v_j) \in A. && (6)
\end{aligned}
$$

onde

$$
\begin{aligned}
x_{ij} &= \begin{cases} 1, & \text{se } (v_i, v_j) \text{ faz parte de um caminho mínimo viável;} \\ 0, & \text{caso contrário; e} \end{cases}\\
M &= \text{um número real qualquer, suficientemente grande.}
\end{aligned}
$$

A equação (1) é a função que descreve o custo da solução e portanto deve ser minimizada. As restrições (2) garantem a elementaridade da solução. Para cada vértice, se um arco que termina no mesmo faz parte da solução, um arco saindo deste também deve fazer parte da solução, com exceção dos vértices origem s e destino t, para os quais deve-se ter, respectivamente, um arco saindo e um arco terminando apenas. As restrições (3) são necessárias para que o valor de t_i^r sejam calculados de forma correta, ou seja, se o arco (v_i, v_j) faz parte da solução deve-se ter $t_j^r = t_i^r + d_{ij}^r$. As restrições (4) garantem que os níveis de cada recurso estão de acordo com os limites impostos. As restrições (5) garantem que nenhum caminho proibido $f \in F$ faz parte da solução. As restrições (6) garantem a integralidade das variáveis x_{ij}.

4 Resultados Obtidos

O modelo descrito na seção 3.3.1 foi implementado e testado com as ferramentas *Ilog CPLEX Optimizer* [1] e *Gurobi Optimizer* [2]. O *CPLEX Optimizer* é uma ferramenta baseada no método *Simplex* que fornece solucionadores flexíveis e de alto desempenho para problemas de programação linear, programação inteira mista, programação quadrática e problemas de programação quadrática limitadas. O *Gurobi Optmizer* é um solucionador no estado da arte para problemas de programação linear, programação quadrática e programação inteira mista, desenvolvido desde o ínicio de modo que pudesse explorar ao máximo as arquiteturas *multi-core* atuais e fazer uso eficiente do paralelismo.

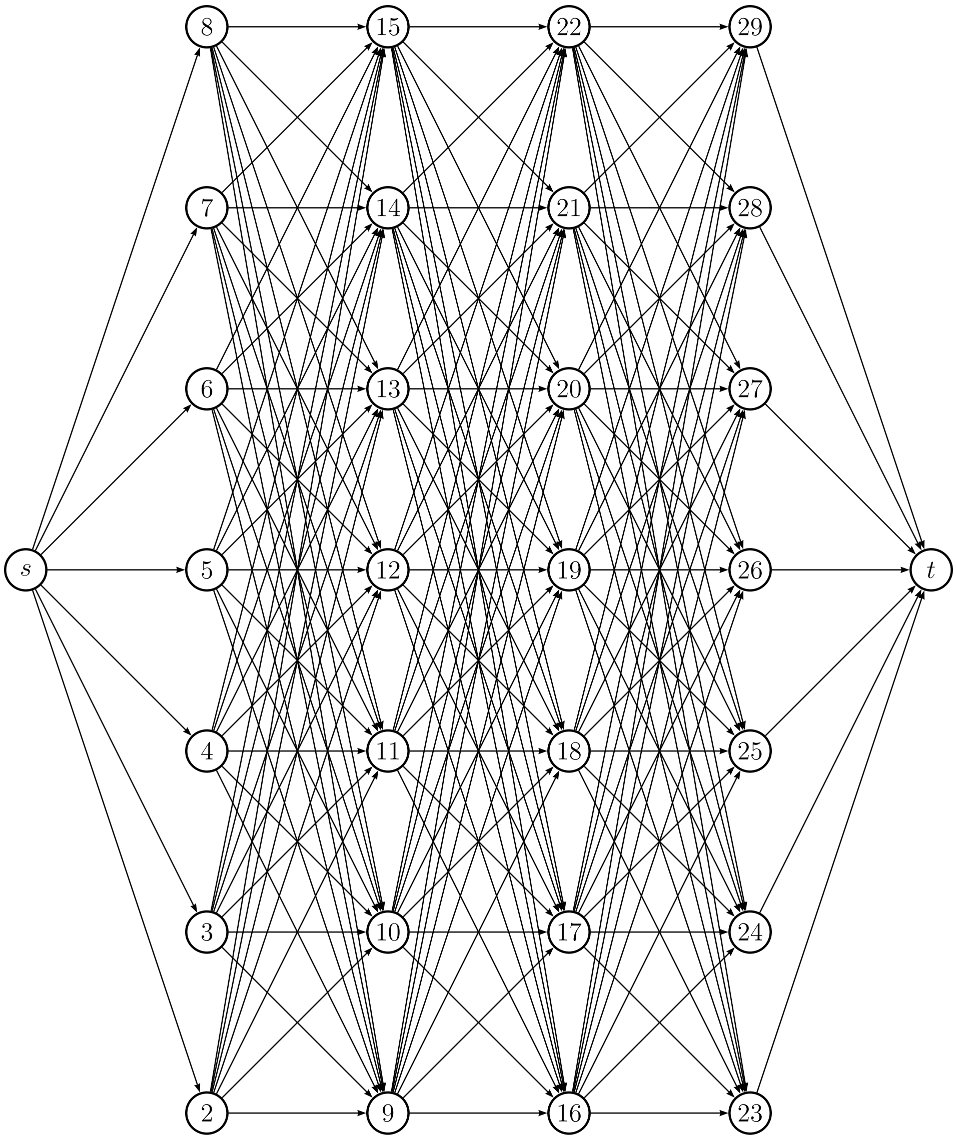

Figura 1: Instância de teste do ESPPRCFP.

O modelo foi testado com a instância mostrada na figura 1. Dado o grafo $G = (V, A)$, $V = \{s, 2, 3, \ldots, 28, 29, t\}$ é o conjunto de vértices e os mesmos são agrupados em: $G_1 = \{s\}$, $G_2 = \{2, 3, ..., 8\}$, $G_3 = \{9, 10, ..., 15\}$, $G_4 = \{16, 17, ..., 22\}$, $G_5 = \{23, 24, ..., 29\}$ e $G_6 = \{t\}$. $A = \{(v_i, v_j) \mid v_i \in G_x \; e \; v_j \in G_{x+1}, 1 \leq x \leq 5\}$ é o conjunto de arcos. São considerados $R = 3$ recursos, $s = 1$ é o vértice origem e $t = 30$ é o vértice destino.

$F = \{\langle 17, 23\rangle\}$ contém o caminho proibido. As constantes são calculadas de acordo com:

$c_{ij} = i + j,\ (i,j) \in A,$ são os custos dos arcos;

$d^r_{ij} = i + j + r,\ (i,j) \in A,\ r = 1, 2, \ldots, R,$ é o consumo do recurso r no arco (i,j);

$a^r_i = i + r,\ i \in V,\ r = 1, 2, \ldots, R;$

$b^r_i = i + 10r,\ i$ não primo, $i \neq 30,\ i \in V,\ r = 1, 2, \ldots, R;$

$b^r_i = i + 120r,\ i$ primo, $i = 30,\ i \in V,\ r = 1, 2, \ldots, R.$

O menor caminho sem considerar restrições é $\langle s, 2, 9, 16, 23, t\rangle$ e tem custo 131. Entretanto, esse caminho viola a restrição (4): $a^1_{16} = 16 + 1 = 17$, $b^1_{16} = 16 + 10 = 26$ e $t^1_{16} = 42$. O menor caminho que satisfaz a restrição (4) é $\langle s, 2, 9, 17, 23, t\rangle$ e tem custo 133. Entretanto, ele viola a restrição (5), uma vez que contém o subcaminho proibido $\langle 17, 23\rangle$. Um caminho de custo mínimo que satisfaz as restrições é: $\langle s, 2, 9, 19, 23, t\rangle$ com custo 137.

As implementações usando as duas ferramentas obtiveram bons resultados, encontrando com sucesso a solução ótima em tempos satisfatórios.

5 Considerações Finais

A elaboração de novas instâncias de testes acabou se mostrando um processo complicado que continua em execução.

Além da abordagem escolhida para resolver o problema existem outras técnicas que possivelmente também podem ser usadas para encontrar uma solução, por exemplo:

- Algoritmos de programação dinâmica, como os estudados por Pecin para solucionar o ESPPRC, provavelmente podem ser adaptadas para levar em consideração as restrições de subcaminhos proibidos adicionais;

- Uso de algoritmos de enumeração de caminhos mínimos até que um caminho viável seja encontrado;

- Modificações nas técnicas usadas nas referências bibliográficas para a solução do SPPFP e SPPRC.

Como trabalho futuro, pretende-se estudar e desenvolver técnicas alternativas à PLI para a resolução do ESPPRCFP, em particular, algoritmos de programação dinâmica.

Referências

[1] IBM ILOG CPLEX Optimizer 12.12. `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

[2] Gurobi Optimizer 4.5. `http://www.gurobi.com/`.

[3] M. Ahmed and Anna Lubiw. Shortest paths avoiding forbidden subpaths. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63–74, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[4] Alfred V. Aho and Margaret J. Corasick. Efficient String Matching: an Aid to Bibliographic Search. *Communications of the ACM*, 18(6):333–340, June 1975.

[5] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[6] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[7] G. Desaulniers, J. Desroisiers, I. Ioachim, M. M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. *Fleet Management and Logistics*, pages 57–93, 1998.

[8] G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column Generation*. Springer, 2005.

[9] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[10] M. Dror. Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW. *Operations Research*, 42:977–978, 1994.

[11] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to some Vehicle Routing Problems. *Networks*, 44(3):216–229, 2004.

[12] Robert W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, June 1962.

[13] L. R. Foulds and H. Longo. Shortest Path Problem with Resource Constraints and Forbidden Subpaths. Trabalho não publicado, 2010.

[14] Marcelo R. Giesel. O problema do caminho mínimo com restrição de subcaminhos de comprimento dois, 2010.

[15] S. Irnich and D. Villeneuve. The shortest path problem with resource constraints and k-cycle elimination for k $\geq$ 3. Technical Report or-2003-01, Lehr- und Forschungsgebiet Operations Research und Logistik Management, RWTH Aachen, RWTH Aachen University, Templergraben 64, 52062 Aachen, 2003.

[16] S. Irnich and D. Villeneuve. The shortest path problem with resource constraints and k-cycle elimination for k $\geq$ 3. *INFORMS Journal on Computing*, 18(3):391–406, 2006.

[17] Ford Jr. and Lester Randolph. Network flow theory. Paper P-923, The RAND Corperation, Santa Monica, California, August 1956.

[18] Ernesto Q. V. Martins. An algorithm for ranking paths that may contain cycles. *European Journal of Operational Research*, 18(1):123–130, October 1984.

[19] Diego G. Pecin. Uso de rotas elementares no cvrp. Master's thesis, Universidade Federal de Goiás, 2010.

[20] G. Righini and Matteo Salani. Symmetry Helps: Bounded Bi-Directional Dynamic Programming for the Elementary Shortest Path Problem with Resource Constraints. *Discrete Optimization*, 3(3):255–273, 2006.

[21] G. Righini and Matteo Salani. New Dynamic Programming Algorithms for the Resource Constrained Elementary Shortest Path Problem. *Networks*, 51(3):155–170, 2008.

[22] Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. *Discrete Applied Mathematics*, 126(2-3):261 – 273, 2003.

[23] D. Villeneuve and G. Desaulniers. The shortest path problem with forbidden paths. *European Journal of Operational Research*, 165(1):97 – 107, 2005.