UM MÉTODO PRIMAL-DUAL PARA O PROBLEMA DO CAMINHO MÍNIMO EM DIGRAFOS NA PRESENÇA DE CICLOS ABSORVENTES

Rafael Castro de Andrade
Departamento de Estatística e Matemática Aplicada - Universidade Federal do Ceará
Campus do Pici, Bloco 910. CEP 60.455-760 - Fortaleza, Ceará - Brasil
rca@lia.ufc.br

Kennedy Anderson Guimarães de Araújo
Bacharelado em Matemática Industrial - Universidade Federal do Ceará
Campus do Pici, Bloco 910. CEP 60.455-760 - Fortaleza, Ceará - Brasil
kennedy@lia.ufc.br

Rommel Dias Saraiva
Programa de Mestrado e Doutorado em Ciência da Computação - Universidade Federal do Ceará
Campus do Pici, Bloco 910. CEP 60.455-760 - Fortaleza, Ceará - Brasil
rommel@lia.ufc.br

RESUMO

Seja $G = (V, A)$ um digrafo, em que V representa o conjunto de vértices, A o conjunto de arcos, e $c_{ij} \in \mathbb{R}$ o custo de um arco $ij \in A$. O problema do caminho mínimo em digrafos contendo ciclos absorventes (PCMDCA) é $\mathcal{NP}$-difícil e consiste em identificar um caminho elementar de custo mínimo entre um vértice-origem $s \in V$ e um vértice-destino $t \in V$ evitando ciclos de custo estritamente negativo. Propomos um novo modelo primal-dual compacto para o PCMDCA. Resultados computacionais para instâncias *benchmark* e para novas instâncias geradas aleatoriamente evidenciam que nossa abordagem possibilita resolver o problema de forma mais eficiente, fornecendo soluções ótimas para todas as instâncias em tempos de execução menores que os obtidos pelos modelos da literatura para o PCMDCA.

PALAVRAS CHAVE. Otimização combinatória, Programação matemática, Caminho mínimo em digrafos com ciclos absorventes.

Área Principal: Programação Matemática, Otimização Combinatória.

ABSTRACT

Consider a digraph $G = (V, A)$, with set of nodes V and set of weighted arcs A. Let $c_{ij} \in \mathbb{R}$ denotes the cost of an arc $ij \in A$. The shortest path problem in digraphs with negative cycles (SPPDNC) is $\mathcal{NP}$-hard and consists in finding an elementary path with minimum cost between an origin node $s \in V$ and a destination node $t \in V$ avoiding negative cycles. In this work, we propose a novel compact primal-dual method for the SPPDNC and report numerical experiments on both benchmark and random generated instances of the problem. The new approach produces optimal solutions for all instances in less computational time than existing ones for the SPPDNC.

KEYWORDS. Combinatorial optimization, Mathematical programming, Shortest path problem in digraphs with negative cycles.

Main Area: Mathematical Programming, Combinatorial Optimization.

1. Introdução

O problema clássico do caminho mínimo consiste em determinar um caminho de custo mínimo entre dois vértices de um dado grafo. Na literatura, encontramos algoritmos especializados em resolver esse problema, como os de Ford Jr (1956) e Dijkstra (1959). No entanto, a aplicação desses algoritmos depende das características de cada grafo. O algoritmo de Ford requer que o grafo não contenha ciclos absorventes (isto é, de custo estritamente negativo), enquanto o algoritmo de Dijkstra se aplica apenas em grafos com custos não-negativos em suas arestas.

Neste trabalho, investigamos o problema do caminho mínimo em digrafos com ciclos absorventes (PCMDCA), comumente presente em operações logísticas de crédito cambial, de roteamento em redes de telecomunicações, dentre outras. Esse problema, para grafos com todos os arcos tendo custos negativos, equivale a encontrar um caminho Hamiltoniano de custo mínimo, sendo assim $\mathcal{NP}$-difícil (Haouari et al., 2013). Seja dado um digrafo $G = (V, A)$, com conjunto de vértices representado por V e, de arcos, por A. Cada arco $ij \in A$ tem um custo $c_{ij} \in \mathbb{R}$ que pode ser negativo. Considere um vértice-origem $s \in V$ e um vértice-destino $t \in V$ diferente de s. O objetivo do PCMDCA é determinar um s, t-caminho elementar (sem repetição de vértices) de custo mínimo.

Em uma breve revisão do estado da arte, encontramos poucos métodos exatos para o PCMDCA. Na verdade, grande parte dos trabalhos se concentra em detectar ciclos absorventes em grafos (Gu et al., 2009; Hougardy, 2010; Subramani et al., 2010) do que resolver o problema propriamente dito. É de Ibrahim et al. (2009) o primeiro modelo de programação linear inteira mista (PLIM) para o PCMDCA, com $O(|V|^2)$ variáveis binárias, $O(|V|^3)$ variáveis contínuas e $O(|V|^3)$ restrições. Já no recente trabalho de Haouari et al. (2013), encontramos uma outra formulação matemática, que contém $O(|V|^2)$ variáveis binárias, $O(|V|^2)$ variáveis contínuas e $O(|V|^2)$ restrições. Esta última era a mais eficiente até o momento na literatura do problema.

O principal objetivo deste trabalho é implementar um novo método primal-dual compacto para o PCMDCA. Mostramos, a partir de resultados numéricos sobre um conjunto de 88 instâncias, que nossa abordagem mostrou ser a mais eficiente em termos computacionais para lidar com esse problema. A prova de corretude de nosso modelo e o desenvolvimento teórico que possibilitou seu desenvolvimento serão abordados em uma versão estendida deste trabalho.

O restante do texto é organizado da seguinte forma. Na Seção 2, reproduzimos a formulação de Haouari et al. (2013). Na Seção 3, descrevemos o novo modelo primal-dual compacto para o PCMDCA. Na Seção 4, relatamos os experimentos computacionais. Por fim, na Seção 5, apresentamos uma breve conclusão e sugestões de trabalhos futuros.

2. Um modelo não-linear para o PCMDCA

Nesta seção, apresentamos o modelo de Haouari et al. (2013) para o PCMDCA. Para uma melhor compreensão, considere a seguinte notação:

- δ_j^+ e δ_j^-: conjunto de vértices sucessores e predecessores do vértice j, respectivamente;
- x_{ij}: variável binária que assume o valor 1 (um) caso o arco ij pertença à solução; e 0 (zero), caso contrário;
- u_j: variável representando o número de arcos induzidos pelas componentes não nulas de x entre o vértice-origem s e um dado vértice j pertencente ao caminho, com $u_s = 0$.

$$(P) \quad \min \quad \sum_{ij \in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.a.} \quad \sum_{j \in \delta_s^+} x_{sj} = 1 \tag{2}$$

$$\sum_{j \in \delta_t^-} x_{jt} = 1 \tag{3}$$

$$\sum_{i \in \delta_j^-} x_{ij} - \sum_{i \in \delta_j^+} x_{ji} = 0, \quad \forall j \in V \backslash \{s, t\} \tag{4}$$

$$\sum_{i \in \delta_j^-} x_{ij} \leq 1, \quad \forall j \in V \backslash \{s, t\} \tag{5}$$

$$u_j x_{ij} = (u_i + 1) x_{ij}, \quad \forall j \in V \backslash \{s, t\}, i \in \delta_j^- \backslash \{s\} \tag{6}$$

$$u_j x_{sj} = x_{sj}, \quad \forall j \in \delta_s^+ \tag{7}$$

$$1 \leq u_j \leq |V| - 1, \quad \forall j \in V \backslash \{s\} \tag{8}$$

$$x_{ij} \in \{0, 1\}, \quad \forall ij \in A \tag{9}$$

No modelo não-linear (P), a função objetivo (1) minimiza o custo do caminho da origem s ao destino t. As restrições (2)–(4) e (9) estabelecem que x induz um caminho que conecta s a t. Em Haouari et al. (2013) os autores consideram $\delta_s^- = \delta_t^+ = \emptyset$. A restrição (5) impõe que cada vértice $j \in V \backslash \{s, t\}$ tem, no máximo, um arco chegando nesse vértice. Apesar de redundante, ela é usada para fortalecer a relaxação linear do modelo. As restrições (6)–(8) garantem a não existência de ciclos (ou *subtours*) na solução do modelo. Em particular, a restrição (6) representa o fato de que se um arco ij pertence à solução, então o número de arcos u_j até chegar ao vértice j deve ser uma unidade a mais que o número de arcos u_i até chegar ao vértice i. Podemos mostrar que essas variáveis u não necessitam de restrição de integralidade, sendo portanto tratadas como variáveis contínuas. Note que (6) e (7) podem ser linearizadas com a técnica de Sherali and Adams (1990) a partir da seguinte transformação de variáveis:

$$\alpha_{ij} = u_j x_{ij}, \quad \forall j \in V \backslash \{s\}, i \in \delta_j^- \backslash \{s\} \tag{10}$$

$$\beta_{ij} = u_i x_{ij}, \quad \forall j \in V \backslash \{s\}, i \in \delta_j^- \backslash \{s\} \tag{11}$$

Dessa forma, o modelo $(P1)$ a seguir é equivalente ao modelo (P) linearizado com a substituição de variáveis acima (vide Haouari et al. (2013) para mais detalhes).

$$(P1) \quad \min \sum_{ij \in A} c_{ij} x_{ij} \tag{12}$$

s.a. (2), (3), (4), (5), (9)

$$\alpha_{ij} = \beta_{ij} + x_{ij}, \quad \forall j \in V \backslash \{s\}, i \in \delta_j^- \backslash \{s\} \tag{13}$$

$$x_{sj} + \sum_{i \in \delta_j^- \backslash \{s\}} \alpha_{ij} - \sum_{i \in \delta_j^+} \beta_{ji} = 0, \quad \forall j \in \delta_s^+ \tag{14}$$

$$\sum_{i \in \delta_j^-} \alpha_{ij} - \sum_{i \in \delta_j^+} \beta_{ji} = 0, \quad \forall j \in V \backslash (\delta_s^+ \cup \{s, t\}) \tag{15}$$

$$x_{ij} \leq \alpha_{ij} \leq (|V| - 1) x_{ij}, \quad \forall j \in V \backslash \{s\}, i \in \delta_j^- \backslash \{s\} \tag{16}$$

$$x_{ij} \leq \beta_{ij} \leq (|V| - 1) x_{ij}, \quad \forall j \in V \backslash \{s\}, i \in \delta_j^- \backslash \{s\} \tag{17}$$

No modelo linear $(P1)$, as restrições (13)–(17) impedem a presença de ciclos na solução, enquanto as demais, como visto anteriormente, asseguram a existência de um caminho que conecta s a t. Podemos provar que a aplicação do modelo $(P1)$ é limitada, uma vez que ele resolve apenas instâncias onde s (resp. t) não possui predecessores (resp. sucessores) em G. Caso contrário, *subtours* aparecem na solução do problema. Outra limitação de $(P1)$ surge quando $st \in A$. Em tal situação, a aplicação do modelo apresentado se torna inviável. Ambos os casos relatados podem ser contornados em $(P1)$ com a seguinte proposição.

Proposição 1 *O modelo $(P1)$, para ser correto em instâncias apresentando arcos chegando em s ou saindo de t, ou com $st \in A$, necessita de substituir as restrições* (14) *e* (15) *por:*

$$\sum_{i \in \delta_j^-} \alpha_{ij} - \sum_{i \in \delta_j^+} \beta_{ji} = 0, \quad \forall j \in V \backslash \{s, t\} \tag{18}$$

Um exemplo de falha do modelo $(P1)$ para o caso acima relatado, bem como a prova da Proposição 1, serão apresentadas em uma versão completa deste artigo.

3. Formulação primal-dual compacta

Nesta seção, propomos um método primal-dual compacto para o PCMDCA. Considere a seguinte observação.

Observação 1 *Seja $\{(s, s_1), (s_1, s_2), \cdots, (s_{p-1}, s_p), (s_p, t)\}$ um caminho elementar de s a t em $G = (V, A)$ e u_v a distância em arcos de s a v nesse caminho, para todo $v \in \{s, s_1, s_2, \cdots, s_p, t\}$. Então $u_s = 0$, $u_{s_1} = 1$, $u_{s_j} = u_{s_{j-1}} + 1$, para $j \in \{2, 3, \cdots, p\}$, e $u_t = u_{s_p} + 1$.*

A ideia é explorar a observação acima para obter um novo modelo que conecta variáveis primais x e duais u em um único modelo, preocupando-se exclusivamente com as variáveis implicadas no caminho. Assim sendo, considere a notação das variáveis defindas anteriormente para a apresentação do novo modelo primal-dual compacto que se segue.

$$(Q) \quad \min \sum_{ij \in A} c_{ij} x_{ij} \tag{19}$$

$$\text{s.a.} \quad \sum_{j \in \delta_s^-} x_{js} = 0 \tag{20}$$

$$\sum_{j \in \delta_t^+} x_{tj} = 0 \tag{21}$$

$$\sum_{i \in \delta_j^-} x_{ij} - \sum_{i \in \delta_j^+} x_{ji} = \begin{cases} -1, & \text{se } j = s, \\ +1, & \text{se } j = t, \\ \;\;\,0, & \text{caso contrário.} \end{cases}, \quad \forall j \in V \tag{22}$$

$$\sum_{i \in \delta_j^-} x_{ij} \leq 1, \quad \forall j \in V \backslash \{s, t\} \tag{23}$$

$$u_j - u_i \leq 1 + |V|(1 - x_{ij}), \quad \forall\, ij \in A \tag{24}$$

$$u_j - u_i \geq 1 - |V|(1 - x_{ij}), \quad \forall\, ij \in A \tag{25}$$

$$x_{ij} \in \{0, 1\}, \quad \forall\, ij \in A \tag{26}$$

$$u_s = 0,\ u_j \geq 0, \quad \forall j \in V \tag{27}$$

O modelo (Q) apresenta $O(|V|^2)$ variáveis binárias, $O(|V|)$ variáveis contínuas e $O(|V|^2)$ restrições. A restrição (20) (resp. restrição (21)) impõe que s (resp. t) não deve conter predecessores (resp. sucessores) em uma solução viável. As restrições (22) e (23) asseguram que todo vértice contido no caminho entre s e t é visitado apenas uma vez. As restrições (24) e (25), adaptadas de (Andrade, 2013), asseguram que se um arco ij está na solução, então $u_j - u_i = 1$. Podemos mostrar que as restrições (24) são redundantes, podendo ser descartadas do modelo. Já as restrições (25) equivalem às conhecidas restrições de potencial MTZ, podendo ser encontradas na formulação do problema do caixeiro viajante (Miller et al., 1960). Estas últimas restrições, juntamente com (23) e a integralidade das variáveis (26), são usadas para evitar a presença de *subtours* na solução do problema. Por fim, pela descrição de (Q), note que tanto (23) quanto (27) são também redundantes. Entretanto, essas restrições são inseridas no modelo para melhorar a qualidade da solução das relaxações lineares na árvore de busca do *Branch-and-Bound* (B&B).

Proposição 2 *Dado um grafo G com possíveis circuitos absorventes, o modelo (Q) é correto e fornece, se existir, um caminho elementar de custo mínimo entre s e t.*

4. Experimentos computacionais

Implementamos os modelos $(P1)$, de Haouari et al. (2013), e (Q), proposto neste trabalho, em C++ junto ao *solver* IBM ILOG CPLEX 12.6. Os experimentos foram realizados em um PC Intel Core i7, 8 $\times$ 3.40 GHz, 16GB de RAM, no ambiente Linux Ubuntu. As instâncias testadas são divididas em duas classes: as *benchmark* propostas em (Haouari et al., 2013) e as novas instâncias geradas aleatoriamente. Em ambas, a origem e o destino do caminho desejado são os vértices 1 e $|V|$, respectivamente. O primeiro conjunto de instâncias é obtido como segue. Para cada vértice $j \in \{1, \ldots, |V| - 1\}$ é atribuído ao grau de saída, denotado por d_j^+, um valor inteiro no intervalo $[1, min\{3, |V| - j - 1\}]$. Após isso, são escolhidos aleatoriamente d_j^+ vértices de $\{j + 1, \ldots, |V|\}$, representando os sucessores de j. Se no grafo resultante existir um vértice $j \in \{2, \ldots, |V|\}$ sem vizinhos, então um arco ij é adicionado, com $i \in \{1, \ldots, j - 1\}$. Visando gerar ciclos, para cada vértice $j \in \{2, \ldots, |V|\}$, um arco jk é criado com probabilidade 0.5, onde k é escolhido aleatoriamente em $\{2, \ldots, j - 1\}$. Por fim, é atribuído um custo c_{ij} no intervalo inteiro $[1, 50]$ a cada arco $ij \in A$, sendo o valor multiplicado por -1 com probabilidade $\frac{2}{3}$. Já o segundo conjunto de instâncias é gerado da seguinte forma. Para cada par de vértices $i, j \in |V|$, com $i \neq j$, $i \neq |V|$ e $j \neq 1$, um arco ij de custo c_{ij} é adicionado ao grafo dada uma probabilidade ρ, onde ρ representa a densidade do grafo dada como entrada do algoritmo e c_{ij} é um valor inteiro escolhido aleatoriamente dentro do intervalo $(-49, 50]$.

As Tabelas 1 e 2 detalham os resultados obtidos nos experimentos computacionais. Nelas, n, m e z denotam, nessa ordem, o número de vértices, o número de arcos e a solução ótima de uma determinada instância, enquanto $t(s)$ representa o tempo de execução (em segundos) do CPLEX para cada modelo. As tabelas indicam igualmente o número de iterações $iter$, que denota a quantidade total de iterações do CPLEX para resolver as instâncias e o de nós bb avaliados na árvore de B&B do CPLEX. É importante destacar que, em diversas ocorrências, realizamos um pré-processamento sobre instâncias geradas aleatoriamente com o objetivo de eliminar arcos chegando em s e saindo de t, ou direcionados de s para t, viabilizando a aplicação do modelo $(P1)$ para os novos casos de testes elaborados. Note, para efeito de comparação com os resultados de Haouari et al. (2013), que ordenamos as linhas das tabelas por número de vértices e por número de arestas.

Como observamos nas tabelas, tanto o modelo $(P1)$ quanto o modelo (Q) encontram a solução ótima para todas as 88 instâncias abordadas. Analisando globalmente os dois conjuntos de instâncias em relação ao tempo de execução, notamos que o modelo (Q) foi melhor que o modelo $(P1)$ em 74 casos. Em apenas 10 instâncias o tempo de execução do modelo $(P1)$ foi menor que o do modelo (Q). No restante das 4 instâncias, os tempos de execução de ambos os modelos foram equivalentes. Levando em consideração o tempo de execução médio total, o do modelo $(P1)$ foi de 20.24 segundos, enquanto o do modelo (Q) foi de 4.86 segundos. Quanto ao número de subproblemas (*bb*) resolvidos pelo B&B do CPLEX, em 23 instâncias o número de nós resolvidos pelo modelo (Q) foi menor que o do modelo $(P1)$. Em 21 instâncias o número de nós resolvidos pelo modelo $(P1)$ foi menor que o do modelo (Q). No restante das 44 instâncias, o número de nós resolvidos por ambos os modelos foi equivalente. Já em termos de número de iterações ($iter$), em 53 instâncias o número de iterações do modelo (Q) foi menor que o do modelo $(P1)$. Em apenas 35 instâncias o número de iterações do modelo $(P1)$ foi menor que o do modelo (Q). Isso é um forte indício para explicar a melhor performance em termos de tempo de execução do modelo (Q) frente ao modelo $(P1)$.

Tabela 1: Resultados - instâncias benchmark (Haouari et al., 2013).

Instância			***(P1)***			***(Q)***		
n	***m***	***z***	***bb***	***iter***	***t(s)***	***bb***	***iter***	***t(s)***
10	29	-185	0	30	0.01	0	16	0.01
10	29	-232	0	29	0.01	0	20	0.00
20	70	-392	0	82	0.01	0	43	0.01
20	74	-446	0	70	0.01	0	55	0.01
30	110	-744	0	110	0.01	0	67	0.01
30	114	-815	0	214	0.04	0	172	0.03
40	154	-1057	0	569	0.09	25	632	0.05
40	158	-928	0	855	0.21	68	1579	0.09
50	193	-1214	0	355	0.08	0	448	0.06
50	196	-1281	0	448	0.07	96	1557	0.06
60	234	-1623	0	469	0.11	0	442	0.08
60	242	-1519	0	872	0.18	263	3232	0.13
70	276	-1824	0	551	0.10	127	2721	0.09
70	285	-1912	0	1120	0.17	0	338	0.04
80	318	-2176	0	646	0.11	0	308	0.07
80	323	-2012	0	1632	0.34	0	968	0.20
90	353	-2339	0	1678	0.35	0	464	0.11
90	357	-2293	0	517	0.10	0	399	0.06
100	407	-2628	0	1110	0.29	0	411	0.09
100	414	-2768	0	912	0.21	0	362	0.07
200	821	-5133	560	37998	1.25	70	4760	0.43
200	827	-5288	682	36400	1.21	82	3545	0.41
300	1227	-7856	10	6495	1.21	0	1974	0.36
300	1229	-7988	34	3785	1.56	5922	209888	6.24
400	1657	-10327	0	1683	2.00	0	2567	0.83
400	1660	-10276	0	5048	2.99	0	2476	0.70
500	2078	-12999	112	16515	2.68	0	4248	1.36
500	2085	-13086	802	151146	7.09	232	31164	2.13
600	2486	-15169	413	60129	4.46	602	91218	3.76
600	2469	-15573	1002	258524	18.90	676	83933	3.40
700	2907	-18188	642	179710	14.79	1834	153645	8.23
700	2932	-17841	939	328494	33.91	226	29506	2.59
800	3280	-20510	854	156628	12.24	101	34120	4.28
800	3329	-20269	1264	559680	62.73	415	67153	5.73
900	3693	-23049	388	316206	37.76	830	158195	8.89
900	3712	-23719	4876	1425781	260.40	773	147755	11.01
1000	4158	-25680	644	388121	82.20	2274	245111	18.54
1000	4176	-26076	145	90868	12.62	553	164268	10.87

5. Conclusão

Neste artigo, introduzimos um método primal-dual compacto para resolver o problema do caminho mínimo em digrafos com ciclos absorventes. Além disso, constatamos limitações na formulação de programação linear inteira mista de Haouari et al. (2013), até então a mais eficiente para tratar o problema em questão. Experimentos computacionais realizados em uma bateria de 88 instâncias de densidades de arco variadas mostram que, de uma forma geral, o modelo primal-dual proposto é mais eficiente que os demais encontrados na literatura. Como trabalhos em preparação, estamos implementando um algoritmo de plano de cortes usando separação de desigualdades generalizados de ciclos para o problema, bem como um algoritmo especializado de *Branch-and-Bound*

Tabela 2: Resultados - instâncias aleatórias.

Instância			***(P1)***			***(Q)***		
n	***m***	***z***	***bb***	***iter***	***t(s)***	***bb***	***iter***	***t(s)***
100	1899	-3968	0	3444	0.85	384	15007	0.91
100	1913	-4085	0	927	0.25	0	415	0.14
100	1937	-4118	475	50107	2.08	94	4345	0.72
100	1985	-4158	0	1255	0.87	0	764	0.56
100	1983	-4192	15	11536	1.29	0	998	0.62
100	3878	-4501	2847	215192	9.04	0	1384	1.07
100	3778	-4450	0	699	1.45	220	6239	1.41
100	3822	-4518	0	2398	2.65	55	3928	1.37
100	3892	-4495	23	10812	4.51	0	516	0.51
100	3870	-4492	661	316129	8.27	0	1090	0.79
100	5835	-4639	66	5435	4.99	0	986	0.82
100	5772	-4626	0	557	1.11	0	1057	1.10
100	5804	-4653	0	22	0.97	0	601	0.22
100	5797	-4635	0	19	0.79	0	606	0.25
100	5830	-4603	0	4782	3.84	0	737	0.83
100	7759	-4724	793	446312	28.50	0	2825	3.41
100	7771	-4685	0	2791	2.57	0	784	1.80
100	7761	-4692	0	2205	2.48	0	830	1.01
100	7793	-4709	0	27	1.35	0	730	0.40
100	7758	-4688	0	2540	4.01	0	2197	3.33
100	9703	-4718	0	3101	3.43	0	1083	1.36
100	9703	-4741	0	18	2.27	0	927	0.61
100	9703	-4725	0	6528	9.03	163	10536	3.60
100	9703	-4744	0	7456	6.66	0	1002	1.11
100	9703	-4729	0	1052	5.54	126	8964	3.40
200	7870	-9022	719	624407	37.59	282	34882	5.12
200	7832	-9036	310	19220	4.49	474	56088	8.54
200	7782	-9040	0	10393	5.35	0	1120	1.42
200	7919	-9072	0	1222	2.09	0	1378	2.36
200	7948	-9130	0	72	1.62	0	1327	1.96
200	15872	-9468	104	56362	54.82	0	6382	9.05
200	15659	-9429	0	499	3.22	0	2713	4.48
200	15584	-9475	0	18407	16.09	0	1939	6.96
200	15813	-9446	290	711939	102.14	0	2736	8.73
200	15868	-9432	96	355528	70.45	378	61184	31.64
200	23819	-9567	0	20	6.96	0	2676	1.77
200	23499	-9567	0	4529	7.63	0	2563	5.77
200	23767	-9574	55	18967	8.61	0	2383	10.16
200	23683	-9567	818	1182882	329.27	40	14305	8.69
200	23601	-9564	0	23	7.54	0	2427	1.83
200	31462	-9642	0	20	7.84	0	3215	2.57
200	31499	-9657	75	21453	12.11	237	34894	43.59
200	31703	-9637	465	1664242	339.77	0	3764	12.69
200	31538	-9643	0	221	13.57	112	25136	42.40
200	31490	-9647	0	20	8.78	0	2741	2.81
200	39403	-9685	0	20	12.67	0	3313	5.11
200	39403	-9681	0	21	13.49	0	3561	3.37
200	39403	-9677	0	419	19.87	234	42381	70.53
200	39403	-9691	0	20	10.67	0	3372	3.26
200	39403	-9683	0	22	12.90	0	3737	10.42

que explora técnicas de $upper$-tolerância no processo de ramificação de nós na árvore de busca. Por fim, pretendemos investigar o impacto da densidade de arcos na dificuldade de resolução das instâncias desse problema.

Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Edital Universal, Processo 449254/2014-3. Somos gratos pelos comentários construtivos de revisores anônimos para melhorar a qualidade deste trabalho.

Referências

Andrade, R. C. (2013). Elementary shortest-paths visiting a given set of nodes. *Simpósio Brasileiro de Pesquisa Operacional*, pages 2378–2388.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Ford Jr, L. R. (1956). Network flow theory. Technical report, DTIC Document.

Gu, X., Madduri, K., Subramani, K., and Lai, H.-J. (2009). Improved algorithms for detecting negative cost cycles in undirected graphs. In *Frontiers in Algorithmics*, pages 40–50. Springer.

Haouari, M., Maculan, N., and Mrad, M. (2013). Enhanced compact models for the connected subgraph problem and for the shortest path problem in digraphs with negative cycles. *Computers & Operations Research*, 40(10):2485–2492.

Hougardy, S. (2010). The floyd–warshall algorithm on graphs with negative cycles. *Information Processing Letters*, 110(8):279–281.

Ibrahim, M., Maculan, N., and Minoux, M. (2009). A strong flow-based formulation for the shortest path problem in digraphs with negative cycles. *International Transactions in Operational Research*, 16(3):361–369.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329.

Sherali, H. D. and Adams, W. P. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430.

Subramani, K., Tauras, C., and Madduri, K. (2010). Space–time tradeoffs in negative cycle detection–an empirical analysis of the stressing algorithm. *Applied Mathematics and Computation*, 215(10):3563–3575.