Resumo do artigo: *The Parameterized Approximability of TSP with Deadlines*, de autoria de Bockenhauer, Hromkovic, Kneis e Kupke (2007)

Mauro Henrique Mulati[1]

[1]IC–UNICAMP

18 de dezembro de 2016

Resumo

Resumo apresentado na disciplina *MO829B – Tópicos em Teoria da Computação – Algoritmos Parametrizados* da pós-graduação do Instituto de Computação no 2º semestre de 2016, ministrada pelo prof. Lehilton L. C. Pedrosa. Objetiva-se enfatizar o tratamento do artigo à versão parametrizada pelo número de *deadlines* do Problema do Caixeiro Viajante métrico com *deadlines*. Utiliza-se uma combinação de técnicas de algoritmos de aproximação e de complexidade parametrizada. Uma fpt-$2,5$-aproximação é mostrada para o problema, e além disso, mostra um *lower bound* de $2-\epsilon$ para o fator de aproximação, para qualquer $\epsilon > 0$, a menos que $P = NP$.

1 Introdução

No Problema do Caixeiro Viajante (TSP) são dados como entrada um grafo não orientado $G = (V, E)$ e pesos $c : E \to \mathbb{Q}_{>0}$, onde $n = |V|$. O objetivo é encontrar um ciclo de peso mínimo que passe por cada vértice exatamente uma vez e retorne ao vértice inicial. No TSP com *Deadlines* DlTSP alguns vértices possuem *deadlines*, de modo que vértice com *deadline* não pode ser visitado após esse ponto. Também usaremos *deadline* para nos referirmos a um vértice com tal característica. Seja o prefixo Δ indicativo de versão métrica.

No artigo é mostrado que não existe aproximação com fator constante para o ΔDlTSP, a menos que $P = NP$. Entretanto, combinando métodos de *algoritmos de aproximação* com ideias de *complexidade parametrizada*, os autores aplicaram o conceito de *aproximação parametrizada*: obtiveram uma 2,5-aproximação para o ΔDlTSP com tempo de execução $k! \cdot n^{O(1)}$, onde k é o número de vértices com *deadline* i.e., uma 2,5-aproximação para o k-ΔDlTSP. Além disso, no trabalho é provado que não existe algoritmo fpt com aproximação $2 - \epsilon$ para qualquer $\epsilon > 0$, a menos que $P = NP$. Este limite inferior vale para ΔDlTSP com 2 ou mais *deadlines*. Desse modo, concluímos que ΔDlTSP não é *fixed-parameter tractable*, se o parâmetro for o número de *deadlines*).

Sabemos que a dificuldade em resolver o ΔDlTSP é pelo menos a de se resolver o ΔTSP (ΔTSP $\leq_p$ ΔDlTSP), e o trabalho mostra inclusive, que não existe $o(|V|)$-aproximação para o ΔDlTSP. Assim, o ΔDlTSP é mais difícil que o ΔTSP. Todos esses resultados de inaproximabilidade se transferem para o ΔTSP com janela de tempo.

As seções, figuras, algoritmos, definições, teoremas e colorários possuem os mesmos números que seus correspondentes no artigo [BHKK07].

2 Definições

A seguir, vamos definir formalmente aproximação parametrizada.

Def. 3. *Seja (U, κ) um problema de otimização parametrizado. Um algoritmo A é uma fpt-α-aproximação para (U, κ) se e somente se:*

- *O algoritmo A é uma α-aproximação para U; e*
- *Existe uma função f e um polinômio p de modo que para toda as instâncias de entrada x de U, o tempo de execução $T_A(x)$ de A em x pode ser limitado da forma $T_A(x) \leq f(\kappa(x)) \cdot p(|x|)$.*

Onde, pela Def. 1 (do artigo) κ é uma parametrização de U e por k-U denotamos restrição em U de modo que somente as instâncias de entrada x tal que U que $\kappa(x) \leq k$ são admissíveis.

Vamos definir formalmente DLTSP. A primeira definição discorre sobre a entrada e a saída.

Def. 4. *Seja $G = (V, E)$ um grafo completo com pesos nas arestas $c : E \to \mathbb{Q}_{>0}$. Denominamos (s, D, d) um deadline set para G se $s \in V$, $D \subseteq V \setminus \{s\}$ e $d : D \to \mathbb{Q}_{>0}$. Um vértice $v \in D$ é chamado deadline vertex. Um caminho $(v_0, v_1, \ldots, v_n)$ satisfaz os deadlines sse $s = v_0$ e para todo $v_i \in D$ temos*

$$\sum_{j=1}^{i} c(\{v_{j-1}, v_j\}) \leq d(v_i).$$

Um ciclo $(v_0, v_1, \ldots, v_n, v_0)$ satisfaz os deadlines sse contém um caminho $(v_0, v_1, \ldots, v_n)$ satisfazendo os deadlines.

Dados G e um *deadline set*, o objetivo do DLTSP é encontrar um ciclo hamiltoniano de que satisfaça todos os *deadlines* e tenha peso total mínimo. No caso do ΔDLTSP, a entrada os pesos das arestas satisfazem a desigualdade triangular.

3 Algoritmo fpt-aproximado para o ΔDLTSP (k-ΔDLTSP)

Como k é constante, podemos testar todas as permutações possíveis de ordens de visitação dos *deadlines*. Considere então que sabemos a ordem em que os *dealines* são visitados em uma solução ótima: podemos inserir corretamente os vértices restantes nesta ordem para obter a solução ótima. Porém, inserir corretamente os vértices restantes ainda é difícil. Desse modo, o algoritmo simplesmente os coloca ao fim da visitação feita nos *deadlines*.

Teo. 1. ALG *resolve* ΔDLTSP *com fator de aproximação $\leq 2,5$ em tempo $O(n^3 + k! \cdot k)$.*

Demonstração. Vamos apresentar um esboço da ideia. Seja I uma entrada para o ΔDLTSP.

Sobre o tempo de execução. Computa X por CHRISTOFIDES em tempo $O(n^3)$ [PS82]. Despois computa e verifica factibilidade das permutações dos *deadlines*: leva tempo $k! \cdot O(k)$. Em seguida, obtém E_π concatenando π e X; e depois remove o vértice s do meio para obter H_π: toma tempo constante.

Algorithm 1

$\textsc{Alg}(\mathbf{G} = (V, E, c),\ (\mathbf{s}, \mathbf{D}, \mathbf{d}),\ \mathbf{k} = |D|)$

$X = (\mathbf{s}, x_1, \ldots, x_{n-k-1}, s) = \textsc{Christofides}(G[V \setminus D])$ // [Chr76]

for each permutação $\pi = (s, d_1, d_2, \ldots, d_k)$ de $D \cup \{s\}$

  if π satisfaz os *deadlines*

    $E_\pi = (s, d_1, d_2, \ldots, d_k, \mathbf{s}, x_1, x_2, \ldots, x_{n-k-1}, s)$ // Seja $C = (s, d_1, d_2, \ldots, d_k, \mathbf{s})$

    $H_\pi = (s, d_1, d_2, \ldots, d_k, x_1, x_2, \ldots, x_{n-k-1}, s)$

return H_π de menor custo satisfazendo *deadlines*, se algum foi encontrado

O algoritmo claramente retorna uma solução ótima, se existir. Mais detalhes no artigo.

Vamos provar o *upper bound* do fator de aproximação. Para uma instância I do problema sejam $\text{SOL} = \text{sol}(I) = \textsc{Alg}(I)$, $\text{OPT} = \text{opt}(I)$ e π_{OPT} a ordem dos *deadlines* em OPT. Temos que $\text{cost}(\text{SOL}) \leq \text{cost}(H_\pi)$, e de fato, são iguais.

O *tour* H_π é E_π com um vértice s removido e arestas em torno dele corrigidas. Temos que E_π consiste de dois ciclos: C (composto dos vértices em $D \cup \{s\}$ ordenados de acordo com π) e X (composto por todos os vértices em $V \setminus D$).

Temos que $\text{cost}(C) \leq \text{cost}(\text{OPT})$, pois C é OPT com vértices removidos e arestas corrigidas: como o grafo é completo, isto é válido; e não nos esqueçamos que todos os *deadlines* estão em C, e naturalmente em OPT também.

Temos que X é uma $\frac{3}{2}$-aproximação do ciclo hamiltoniano de peso mínimo de algum subgrafo G' de G, e de fato $G' = G[V \setminus D]$. Assim

$$\begin{aligned}
\text{cost}(X) &= \text{cost}(\textsc{Christofides}(G[V \setminus D])) \\
&\leq \frac{3}{2}\text{cost}(\text{opt}_{\Delta\text{TSP}}(G[V \setminus D])) \\
&\leq \frac{3}{2}\text{cost}(\text{opt}_{\Delta\text{TSP}}(G)) \\
&\leq \frac{3}{2}\text{cost}(\text{OPT}),
\end{aligned}$$

notando que, em X há somente vértices que *não* são *deadlines*, e obviamente, todos esses vértices também estão em OPT.

Portanto

$$\begin{aligned}
\text{cost}(\text{SOL}) &= \text{cost}(H_\pi) \\
&\leq \text{cost}(C) + \text{cost}(X) \\
&\leq \frac{3}{2}\text{cost}(\text{OPT}) + \text{cost}(\text{OPT}) \\
&= \frac{5}{2}\text{cost}(\text{OPT}).
\end{aligned}$$

■

Cor. 1. $\textsc{Alg}$ *é uma fpt-2,5-aproximação para o* ΔDLTSP.

4 *Lower bound* para k-ΔDLTSP

Seja o prefixo Δ_β indicativo de versão com métrica relaxada, de maneira que teríamos uma β-desigualdade triangular, onde para algum $\beta > 1$ e quaisquer vértices u, v e x temos que

$$c(\{u,v\}) \leq \beta\left(c(\{u,x\}) + c(\{x,v\})\right).$$

Depois de mostrar o *lower bound* para o k-Δ_βDLTSP, o artigo parte para o k-ΔDLTSP. Aqui vamos direto para o k-ΔDLTSP, adaptando o que for necessário.

Pelo que o artigo expõe sobre o k-Δ_βDLTSP pode-se chegar que a um *lower bound* de $\frac{3}{2} - \epsilon$ para o fator de aproximação para o 1-ΔDLTSP. Lembremos que o ALG tem fator $\frac{5}{2}$.

Vamos mostrar que não existe, a menos que $P = NP$, algoritmo $(2-\epsilon)$-aproximado para o ΔDLTSPcom dois ou *deadlines*: $2-\epsilon$ é um *lower bound* da aproximabilidade do 2-ΔDLTSP.

Teo. 3. *Não existe algoritmo de tempo polinomial para o* 2-ΔDLTSP *com fator de aproximação* $2-\epsilon$ *para qualquer* $\epsilon > 0$*, a menos que* $P = NP$.

Demonstração. Vamos apresentar um esboço da ideia. Seja $\epsilon > 0$. Vamos mostrar que tal algoritmo de aproximação para o 2-ΔDLTSPpode ser usado para resolver o problema do Caminho Hamiltoniano (HP). Seja $G' = (V', E')$ uma instância para o HP com $|V'| = n$. Vamos construir um grafo completo $G = (G, E, c)$ para o ΔDLTSPcomo segue.

Seja $V' = \{v_1, \ldots, v_n\}$. Construamos dois cópias disjuntas V_1 e V_2 de V'. Façamos $V = V_1 \cup V_2 \cup \{s, D_1, D_2\}$ e para arestas com as duas extremidades em V_1 e V_2, atribuamos

$$c(\{v_i^1, v_j^1\}) = \begin{cases} 1 & \text{se } \{v_i, v_j\} \in E' \\ 2 & \text{se } \{v_i, v_j\} \notin E' \end{cases}$$

$$c(\{v_i^2, v_j^2\}) = \begin{cases} 1 & \text{se } \{n, v_j\} \in E' \\ 2 & \text{se } \{2n, v_j\} \notin E' \end{cases}$$

Agora, temos de tratar arestas que se ligam a s, D_1 e D_2. Façamos

$$c(e) = \begin{cases} \gamma & \text{se } e \in & \{\{s, v^1\} | v^1 \in V_1\} \cup \\ & & \{\{v^1, D_1\} | v^1 \in V_1\} \cup \\ & & \{\{D_1, v^2\} | v^2 \in V_2\} \cup \\ & & \{\{v^2, D_2\} | v^2 \in V_2\} \\ n^2 & \text{se } e \in & \{s, D_2\}. \end{cases}$$

onde γ pode ser escolhido arbitrariamente, desde que

$$\gamma > \frac{(4n^2 - 2)/\epsilon - 2n^2 + 1}{4}.$$

Podemos ver que este grafo satisfaz desigualdade triangular. Por fim, atribuímos às demais arestas o peso máximo possível que respeita a desigualdade triangular. Um esquema do grafo G construído é representado na Fig. 3 a seguir.

Note que:

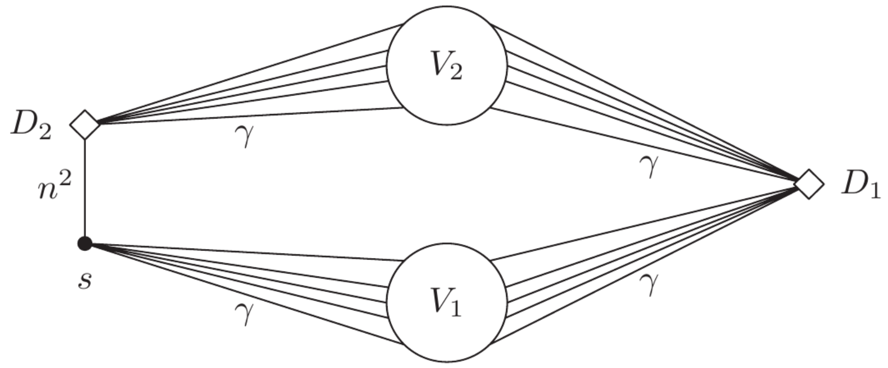

Figura 3: Construção para 2-ΔDlTSP

- A distância entre $v^1 \in V_1$ e $v^2 \in V_2$ é exatamente 2γ;
- Uma aresta entre $v^1 \in V_1$ e D_2 custa $\gamma + 2$.

Fazemos:

- $d(D_1) = 2\gamma + n - 1$; e
- $d(D_2) = 4\gamma + n^2 - 1$.

Vamos assumir que G' *contém* um HP denominado p. Uma *solução ótima* para o 2-ΔDlTSP começa em s, segue p em V_1 e chega em D_1 exatamente no "tempo" $2\gamma + n - 1$, sendo que 2γ é o custo da aresta $\{s, v^1\}$ para um $v^1 \in V_1$ somado ao da aresta $\{v^2, D_1\}$ para um $v^2 \in V^2$; e $n - 1$ é o custo de se percorrer p em V_1. Depois disso, o ciclo vai para V_2 e segue p dentro dele, e chega em D_2 no "tempo" $4\gamma + 2n - 2$ (dobramos do "tempo" de chegada em D_1), valor que é menor que $d(D_2)$. Depois retornamos para s, resultando num custo total de $4\gamma + 2n - 2 + n^2$. Temos que $4\gamma + 2n - 2 + n^2 \leq 4\gamma + 2n^2 - 1$ para $n \geq 2$ (por se tratar do problema com pelo menos dois *deadlines*, naturalmente teremos $n \geq 2$).

Agora vamos assumir que G' *não contém* um HP. Nesse caso, não é possível visitar todos os vértices em V_1 gastando apenas $n-1$ antes de se chegar em D_1. Consequentemente, uma *solução ótima* pode atingir D_1 em "tempo" $2\gamma+l$, visitando $l+1$ vértices em V_1, mas $l+1 < n \therefore l < n-1$. Note que o impedimento é o *deadline* de D_1, que precisa que l não passe de $n-1$. Desse modo, esta solução tem de visitar algum vértice $v^1 \in V_1$ mais tarde, entretanto, não pode visitar v_1 logo após D_1, pois isto violaria o *deadline* de D_2, dado que, pela desigualdade triangular, a aresta $\{D_1, D_2\}$ possui custo de pelo menos $2\gamma + l + 2\gamma + n^2 > 4\gamma + n^2 - 1$.

Então, na verdade, após s, esta solução tem de ir direto para D_2, visitar alguns vértices em V_2, mas não todos eles (pois isto acumularia o custo de pelo menos $\gamma + (n-2)n + 2n + \gamma = 2\gamma + n^2$, e violaria o *deadline de* D_1). Depois disso vai para D_1 e depois de visitar $\mathbf{D_1}$, esta solução deve voltar à V_1 e V_2 e retornar para s no final. Note que D_2 não será visitado antes do "tempo" 4γ e o custo de volta é pelo menos 4γ. A solução total tem custo de pelo menos 8γ.

Sejam OPT-NO e OPT-YES as soluções ótimas retornadas pelo algoritmo hipotético para o 2-ΔDlTSP nos casos de podermos inferir que *existe* um HP em G' e no caso de podermos inferir que *não existe*, respectivamente.

Por fim, obtemos o fator de aproximação

$$\frac{\text{cost(OPT-NO)}}{\text{cost(OPT-YES)}} \geq \frac{8\gamma}{4\gamma + 2n^2 - 1} = \frac{\gamma + n^2 - 2}{4\gamma + 2n^2 - 1} - \frac{4n^2 - 2}{4\gamma + 2n^2 - 1} > 2 - \epsilon.$$

Desse modo, um algoritmo $2-\epsilon$-aproximado para o 2-ΔDlTSP poderia ser usado para resolver o problema HP. ■

Cor. 3. *Não existe fpt-aproximação para o* ΔDlTSP *com fator de aproximação* $2 - \epsilon$ *para qualquer* $\epsilon > 0$*, a menos que* $P = NP$.

5 Usando somente aproximação: um *lower bound* para o ΔDlTSP

O trabalho mostra que, nesse caso, não existe algoritmo de tempo polinomial para o ΔDlTSP* (que é uma versão facilitada do ΔDlTSP) com fator de aproximação $((1 - \epsilon)/2)|V|$, para qualquer $0 < \epsilon < 1$, a menos que $P = NP$.

6 Considerações finais

A parametrização é crucial para obter uma aproximação com fator constante: sem a parametrização, o fator de aproximação não pode ser constante.

O trabalho relacionado [BHKK06] enfatiza na aproximabilidade de genelizações relacionadas do ΔTSP, como com janelas de tempo.

7 Questão

Por que o Teo. 1 não se aplica ao Δ_βDlTSP?

Referências

[BHKK06] Hans-Joachim Böckenhauer, Juraj Hromkovič, Joachim Kneis, and Joachim Kupke. *On the Approximation Hardness of Some Generalizations of TSP*, pages 184–195. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[BHKK07] Hans-Joachim Bockenhauer, Juraj Hromkovic, Joachim Kneis, and Joachim Kupke. The parameterized approximability of tsp with deadlines. *Theory of Computing Systems*, 41(3):431–444, 2007.

[Chr76] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[PS82] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity.* Courier Corporation, 1982.