Parameterized Complexity of k-Chinese Postman Problem

Gregory Gutin, Gabriele Muciaccia e Anders Yeo

MO829 - Resumo do Artigo

Francisco Jhonatas 20 de dezembro de 2016

Resumo

Este resumo é referente ao artigo *Parameterized Complexity of k-Chinese Postman Problem* proposto em [3], onde é apresentado o primeiro *kernel* com $O(k^2 \log k)$ vértices para o problema.

1 Introdução

Considere um grafo $G = (V, E)$ com custos não-negativos nas arestas, $|V| = n$ e $|E| = m$. Um passeio fechado é multiconjunto não vazio $T = \{e_1, \cdots, e_r\}$ de arestas $e_i = x_i y_i$ tal que existe uma permutação σ de $\{1, \cdots, r\}$, onde $y_{\sigma(i)} = x_{\sigma(i+1)}$, para todo $1 \leq i \leq r$ (onde $\sigma(r+1) = \sigma(1)$). O problema *k-Chinese Postman Problem* (k-CPP) é uma generalização do *Chinese Postman Problem* (CPP), definidos a seguir:

Chinese Postman Problem (CPP)	
Entrada:	Um grafo $G = (V, E)$ conexo com pesos nas arestas e um inteiro p.
Problema de Decisão:	Existe um passeio fechado em G tal que toda aresta de G está contida no caminho e o peso total das arestas no caminho é no máximo p?

k-Chinese Postman Problem (k-CPP)	
Entrada:	Um grafo $G = (V, E)$ conexo com pesos nas arestas e inteiros p e k
Parâmetro:	k
Problema de Decisão:	Existe um conjunto de k passeios fechados tal que toda aresta de G está contida em pelo menos um dos caminhos e o peso total das arestas nos caminhos é no máximo p?

Os autores propuseram uma nova redução para mostrar que o k-CPP é NP-Difícil a partir do problema 3-*Cycle Partitioning Problem*, onde dado um grafo G, deve-se decidir se as arestas de

G podem ser particionadas em ciclos de tamanho de 3. A redução é a seguinte: defina $k = m/3$, onde m é o número de arestas de G, e o peso de cada aresta de G igual a 1, uma solução para o $(m/3)$-CPP possui custo m se e somente se as arestas de G podem ser particionadas em ciclos de tamanho de 3.

Este artigo apresenta o primeiro algoritmo FPT para o k-CPP, mostrando que o problema apresenta um *kernel* com $O(k^2 \log k)$ vértices. Os autores também provaram que a versão orientada do k-CPP é NP-Difícil, e deixaram como questionamento se a versão orientada também é FPT, quando parametrizada por k.

2 Kernel para o k-CPP

Considerando que $G = (V, E)$ é um grafo conexo com peso nas arestas e que para uma solução $T = \{T_1, \cdots, T_k\}$, onde T_i representa um passeio fechado, para o k-CPP em G, com $k \geq 1$, definimos $G_T = (V, E_T)$ onde E_T é um multiconjunto contendo todas as arestas de E e a quantidade de uma aresta $e \in E_T$ é dada pela quantidade de vezes que ela é usada por $T_1 \cup \cdots \cup T_k$. Observe que se tivermos k caminhos fechados que cobrem todas as arestas de um grafo conexo, a união deles é um caminho fechado cobrindo todas as arestas e, portanto, é uma solução para o CPP. Com isso, temos a seguinte proposição:

Proposição 1: O custo de uma solução ótima para o k-CPP em G não é menor que o custo de uma solução ótima para o CPP em G.

A partir da Proposição 1, temos o seguinte lema:

Lema 1: Seja T uma solução ótima para o CPP em G. Se G_T contém pelo menos k ciclos de arestas disjuntas, então uma solução ótima para o k-CPP em G possui o mesmo custo que T. Além disso, se k ciclos de arestas disjuntas em G_T forem fornecidas, então uma solução ótima para o k-CPP pode ser encontrada em tempo polinomial.

Prova. Considere uma coleção de k ciclos de arestas disjuntas C em G_T (T é uma solução ótima para o CPP em G). Remova todas as arestas de C em G_T, e observe que após a remoção, temos que todo vértice no multigrafo resultante G' possui grau par. Encontre uma solução ótima F para o CPP em cada componente de G' e anexe cada F a um ciclo C no qual possui vértices em comum com F. Como consequência, obtemos uma coleção Q de k caminhos fechados para o k-CPP em G com o mesmo custo de T. Então Q é ótimo pela *Proposição 1*. □

Seja V_1 o conjunto de vértices de grau 1, V_2 o conjunto de vértices de grau 2 e $V_{\geq 3}$ o conjunto de vértices com grau pelo menos 3. Será mostrado que, em tempo polinomial, podemos resolver o k-CPP ou limitar $|V|$ ($V = V_1 \cup V_2 \cup V_{\geq 3}$).

Seja u um vértice com exatamente dois vizinhos v e w. Definimos a operação *bypassing u* como sendo a operação que remove as arestas uv e uw e adiciona uma aresta vw cujo custo é a soma dos custos das arestas uv e uw. Note que esta operação pode criar arestas paralelas.

O seguinte lema foi baseado em [1], cuja prova os autores omitiram. É importante observar que o resultado desse lema também é válido para multigrafos, uma vez que arestas paralelas formam um ciclo de tamanho dois.

Lema 2: Existe uma constante c tal que todo grafo H com grau mínimo 3 e $|V|$ pelo menos $ck\log_2 k$ contém k ciclos de arestas disjuntas. Tais ciclos podem ser encontrados em tempo polinomial.

Precisaremos do seguinte fato para a prova do próximo lema: Existe uma solução ótima T para o CPP em G que usa no máximo duas cópias de cada aresta de G e tal solução pode ser encontrada em tempo polinomial [2].

Lema 3: Se $|V_1| \geq k$ ou $|V_3| \geq ck\log_2 k + k$, onde c é obtido pelo *Lema 2*, então uma solução ótima para o k-CPP em G possui o mesmo custo que uma solução ótima para o CPP em G. Além disso, uma solução ótima para o k-CPP pode ser obtida em tempo polinomial.

Prova. Seja $V_1 = \{v_1, \cdots, v_r\}$ com $r \geq k$ e seja w_i um vizinho de v_i. Encontre em tempo polinomial uma solução ótima T para o CPP em G. Temos que em G_T todo vértice possui grau par, portanto G_T contém duas cópias da aresta v_iw_i para todos $1 \leq i \leq r$, o que representa que existem pelo menos k ciclos de arestas disjuntas de tamanho 2. E pelo *Lema 1*, podemos encontrar em tempo polinomial uma solução ótima para o k-CPP em G de mesmo custo que T.

Agora, vamos assumir que $|V_1| \leq k$ e $|V_{\geq 3}| \geq ck\log_2 k + k$. Remova todos os vértices de grau 1 e realize a operação de *bypassing* em todos os vértices de grau 2. O multigrafo resultante G' contém pelo menos $ck\log_2 k$ vértices e tem grau mínimo de pelo menos 3, então pelo *Lema 2*, G'

contém pelo menos k ciclos de arestas disjuntas que podem ser encontrados em tempo polinomial. De novo, podemos concluir pelo *Lema 1*. $\square$

A partir do *Lema 3* podemos assumir que $|V_1 \cup V_{\geq 3}| = O(k \log_2 k)$. Falta limitar $|V_2|$ e para isso usaremos uma *Reduction rule*, mas antes disso mostraremos o seguinte lema:

Lema 4: Existe uma solução ótima T para o k-CPP em G tal que nenhuma aresta de G, exceto possivelmente a de menor custo, é usada mais que duas vezes em T.

Prova. Seja xy a aresta de menor custo em G e seja T uma solução ótima para o k-CPP em G tal que xy é usada quantas vezes forem necessárias. Assuma, por absurdo, que a aresta uv, diferente de xy, é usada ao menos 3 vezes em T. Observe que G_T é euleriano e contém k ciclos de arestas disjuntos.

Em T, remova duas cópias de uv e adicione duas cópias de xy, obtendo assim T'. Se C_1 e C_2 são dois ciclos de arestas disjuntos usando cópias diferentes de uv em G_T, podemos observar que uvu é um ciclo em G_T e mais, existe um ciclo contendo apenas arestas de C_1 e C_2, distintas de uv. Isto implica que removendo duas cópias de uv de T, diminui o número máximo de ciclos de arestas disjuntas em G_T de um. E adicionar duas cópias de xy aumenta o número máximo de ciclos de arestas disjuntas de um. Portanto, como G_T contém k ciclos de arestas disjuntas, $G_{T'}$ também. Mas isto implica que $G_{T'}$ contém uma solução para o k-CPP com custo de no máximo o de T, o que é um absurdo. $\square$

Regra de Redução 1: Seja $P = v_0 v_1 \cdots v_r v_{r+1}$ um caminho em G tal que v_i é um vértice de grau 2 para $1 \leq i \leq r$. Se $r > k$, realize a operação de *bypassing* em um vértice v_i tal que $2 \leq i \leq r-1$. Escolha v_i de tal forma que *bypassing* não mude o custo mínimo de uma aresta em G.

Esta regra é segura pois em P uma solução para o k-CPP duplicará todas as arestas ou não duplicará nenhuma (com exceção da aresta de custo mínimo, conforme *Lema 4*), e isto é verdade tanto para G como para G' obtido após a aplicação da regra. Portanto, duplicar uma aresta contraída e (resultante da operação *bypassing*) corresponde a duplicar todas as arestas que foram contraídas para a construção de e. Além disso, duplicar o caminho em G' criará pelo menos k ciclos de arestas disjuntas, o que também aconteceria se fosse realizado em G.

A partir de agora vamos assumir que G foi reduzida pela *Reduction Rule 1*. Seja H um multigrafo onde $V(H) = V_1 \cup V_{\geq 3}$ (então $|V(H)| = O(k \log_2 k)$) e adicionaremos r arestas entre dois vértices u e v em H se e somente se existem exatamente r caminhos distintos de u para v em G onde todos os nós internos do caminho possuem grau 2 em G.

Se existirem vértices u e v em H tal que existam $2k$ arestas paralelas entres eles, então G contém pelo menos k ciclos de arestas disjuntas, e com isso podemos aplicar o *Lema 1*. Portanto, vamos assumir que este não seja o caso e que $|E(H)| = O(k^2 \log_2 k)$ e como G foi reduzido pela *Reduction Rule 1*, temos que $|E(G)| = O(k^3 \log_2 k)$. Contudo, podemos melhorar o limitante em $|E(H)|$ e com isso oferecer um *kernel* melhor.

Lema 5: Seja $k \geq 2$ e seja c_1 qualquer constante. Existe uma constante c_2 tal que todo multigrafo H com no máximo $c_1 k \log_2 k$ vértices e pelo menos $c_2 k \log_2 k$ arestas contém k ciclos de arestas disjuntas. Além disso, tais ciclos podem ser encontrados em tempo polinomial.

A partir do *Lema 5* podemos assumir que $|E(H)| \leq c_2 k \log_2 k$ para alguma constante c_2 e como G foi reduzido pela *Reduction Rule 1*, temos que $|V_2| = O(k^2 \log_2 k)$, o que implica o seguinte teorema:

Teorema 1: O k-CPP admite um *kernel* com $O(k^2 \log_2 k)$ vértices.

3 Exercício

Seja T uma solução ótima para o k-CPP. Seguindo a definição apresentada da construção do grafo G_T, explique por que G_T é euleriano e por que G_T apresenta pelo menos k ciclos de arestas disjuntas. (Dica: observe que G_T pode ser um multigrafo e que arestas paralelas formam um ciclo de tamanho 2)

References

[1] Hans L Bodlaender, Stéphan Thomassé, and Anders Yeo. "Kernel bounds for disjoint cycles and disjoint paths". In: *Theoretical Computer Science* 412.35 (2011), pp. 4570–4578.

[2] William J Cook et al. *Combinatorial optimization*. Springer, 2009.

[3] Gregory Gutin, Gabriele Muciaccia, and Anders Yeo. “Parameterized complexity of k-Chinese postman problem”. In: *Theoretical Computer Science* 513 (2013), pp. 124–128.