Conexão de Terminais com Número Restrito de Roteadores e Elos

Mitre C. Dourado[1]
UFRJ - Depto. de Ciência da Computação
Rio de Janeiro, RJ
E-mail: [1]mitre@nce.ufrj.br ,

Rodolfo A. Oliveira[2]
UFRJ - PPGI/NCE
Rio de Janeiro, RJ
E-mail: [2]r.oliveira@ppgi.ufrj.br ,

Fábio Protti[3]
UFF - Instituto de Computação
Niterói, RJ
E-mail: [3]fabio@ic.uff.br ,

Uéverton S. Souza[4]
CEFET-RJ - Departamento de Informática
Rio de Janeiro, RJ
E-mail: [4]ueverton.souza@cefet-rj.br

RESUMO

O PROBLEMA DA ÁRVORE GERADORA e o PROBLEMA DA ÁRVORE STEINER visam a obtenção de um subgrafo acíclico que conecta um conjunto de vértices terminais e satisfaz algumas propriedades. Ambos os problemas têm aplicações importantes em várias áreas que envolvem interconexões de elementos. Seja G um grafo. A *árvore de conexão* de um subconjunto $W \subseteq V(G)$ é um subgrafo acíclico e conexo T de G tal que $W \subseteq V(T)$ e todas as folhas de T estão em W. Em uma árvore de conexão, existem três tipos de vértices: (1) *vértices terminais* , que são aqueles que pertencem a W; (2) vértices não-terminais com grau dois em T, chamados *elos*; (3) vértices não-terminais com grau pelo menos três em T, chamados *roteadores*. Motivados por seu grande potencial de aplicação, propomos um novo problema, denominado PROBLEMA DE CONEXÃO DE TERMINAIS (TCP), onde o número de vértices não-terminais (elos e/ou roteadores) é limitado. Neste trabalho, provamos casos NP-completos e polinomiais para variantes do TCP. Além disso, analisamos a complexidade do problema TCP quando os vértices terminais são folhas, e para o mesmo problema apresentamos um algoritmo parametrizado, o qual provamos pertencer à classe *FPT*, quando o grau máximo, o número de rotadores e o número de elos são limitados.

Palavras-chave: Árvore de Steiner. Árvore Geradora. Ávore de Conexões. Complexidade Computacional. FPT. Teoria e Algoritmos em Grafos.

ABSTRACT

The SPANNING TREE PROBLEM and the STEINER TREE PROBLEM aim at obtaining an acyclic subgraph connecting a set of terminal points and satisfying some properties. Both problems have several important network applications. Let G be a graph. A *connection tree* of a subset $W \subseteq V(G)$ is an acyclic, connected subgraph T of G such that $W \subseteq V(T)$ and all leaves of T are in W. In a connection tree, there are three types of vertices: (1) *terminal vertices*, i.e., those belonging to W; (2) non-terminal vertices with degree two in T, called *linkers*; (3) non-terminal vertices with degree at least three in T, called *routers*. Motivated by its large potential applicability, we propose a new problem in graphs, called TERMINAL CONNECTION PROBLEM (TCP), where the number of non-terminal vertices (linkers and/or routers) is bounded. In this work, we prove NP-complete and polynomial cases for variants of the TCP. Furthermore, we analyze the complexity of the TCP when the terminal vertices are leaves, and we present an FPT algorithm when the maximum degree, the number of routers, and the number of linkers are limited.

Keywords: Steiner tree. Spanning tree. Connection Tree. Computational Complexity. FPT. Theory and Algorithms on Graphs.

1 Introdução

Em [Prim (1957)], discute-se a relevância de conectar pontos terminais com custo mínimo de traços, ou seja, para um determinado conjunto de pontos terminais, o problema consiste em conectar todos os terminais com conexões terminal-a-terminal com o menor comprimento total possível (soma dos comprimentos do traços). Este problema é chamado de PROBLEMA DA ÁRVORE GERADORA e tem várias aplicações em projeto de redes, em particular em comunicação, distribuição e redes de transporte.

Outro problema importante é o PROBLEMA DA ÁRVORE STEINER [Hakimi (1971), Levin (1971)]: dado um grafo G conexo e ponderado e um subconjunto $W \subseteq V(G)$, o problema consiste em encontrar um subgrafo de custo mínimo contendo W, onde o uso de pontos não-terminais adicionais, chamados *vértices de Steiner*, é permitido. Para grafos não ponderados, uma árvore de Steiner é um subgrafo conexo T de G tal que $W \subseteq V(T)$ e $|E(T)|$ é mínimo.

Veja a Figura 1, que ilustra os dois problemas citados.

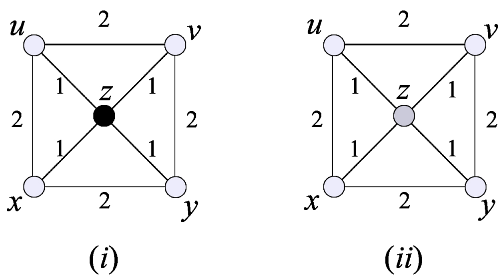


Figura 1. (i) **Temos que qualquer árvore geradora para** $W = \{u, v, x, y\}$ **tem custo mínimo** 6; (ii) **A árvore de Steiner para** $W = \{u, v, x, y\}$ **é a árvore geradora de custo mínimo para** $W \cup \{z\}$**, cujo valor é** 4.

Uma *árvore de conexão* para um subconjunto $W \subseteq V(G)$ é um subgrafo acíclico e conexo T de G, onde $W \subseteq V(T)$ e todas as folhas de T estão em W. Claramente, toda árvore Steiner para W é uma árvore de conexão. Em uma árvore de conexão, existem três tipos de vértices : (1) vértices de W , chamados *terminais*; (2) vértices em $V(T) \backslash W$ com grau dois em T, chamados *elos*; (3) vértices em $V(T) \backslash W$ com grau pelo menos três em T, chamados *roteadores* .

Motivados por potenciais aplicações, estudamos a complexidade de determinar se um grafo não ponderado G admite uma árvore de conexão T para um dado subconjunto $W \subseteq V(G)$ satisfazendo as seguintes condições: (i) $V(T)$ contém no máximo r roteadores; (ii) $V(T)$ contém no máximo ℓ elos. A ideia de limitar o número de vértices não-terminais surgiu de algumas questões em segurança da informação em redes, uma vez que os vértices não-terminais usados para estabelecer a conexão podem tentar acessar informações privadas compartilhadas apenas pelos terminais. Além disso, em algumas situações, estes limites representam o cenário real ao criar diferentes estruturas de conexão, tais como encontrar formas de construir estradas ou ferrovias para conectar um conjunto de localidades, ou decidir políticas de roteamento através da internet para o tráfego *multicast*.

Note que a determinação de uma árvore de conexão mínima satisfazendo (i) e (ii) é equivalente a determinar uma árvore geradora para $W = V(G)$, quando $r = 0$ e $\ell = 0$; e é equivalente a determinar uma árvore Steiner quando $r = \ell = |V(G)\backslash W|$.

Neste artigo, provaremos que decidir se existe uma árvore de conexão T satisfazendo (i) e (ii) é um problema NP-completo, mesmo quando: (a) o parâmetro ℓ é um valor constante fixo; (b) o parâmetro r é um valor constante fixo. Por outro lado, mostraremos que, quando os dois parâmetros são fixos, o problema pode ser resolvido em tempo polinomial. Ainda neste trabalho, analisaremos a complexidade do problema E-TCP, variação do problema TCP onde exige-se que todos os vértices de W sejam folhas da árvore de conexão. Dentre os resultados obtidos para o problema E-TCP, mostramos que o problema permanece NP-completo quando o grafo de entrada possui grau máximo 6 e o número de elos exigidos na árvore de conexão é limitado por uma constante. Além disso, apresentamos um algoritmo parametrizado que prova que o problema é *tratável por parâmetro fixo* quando o grau máximo, o número de roteadores e o número de elos são parâmetros.

2 Complexidade Computacional

O PROBLEMA DA ÁRVORE GERADORA é bem conhecido por ser solucionável em tempo polinomial [Prim (1957)]; em contrapartida, encontrar uma árvore de Steiner é NP-difícil [Garey (1990)]. Em [Dreyfus (1972)], Dreyfus e Wagner apresentaram um algoritmo que, utilizando como ferramenta a programação dinâmica, obtém uma árvore de Steiner em tempo $O(n^3 + n^2 2^{k-1} + n 3^{k-1})$, onde k é um parâmetro que representa o número de terminais; este resultado (obtido de forma independente por Levin [Levin (1971)]) implica em um algoritmo de tempo polinomial quando k é limitado por um valor constante ou por uma função de ordem $O(\log n)$.

Definimos o PROBLEMA DE CONEXÃO DE TERMINAIS da seguinte forma:

PROBLEMA DE CONEXÃO DE TERMINAIS – TCP
Instância: Um grafo conexo G, um subconjunto $W \subseteq V(G)$, e dois inteiros não negativos ℓ e r.
Questão: O grafo G contém uma árvore de conexão T com no máximo ℓ elos e r roteadores?

Claramente, o TCP está em NP, pois é fácil verificar se uma árvore T conecta W usando no máximo ℓ elos e r roteadores. Definindo valores constantes para alguns parâmetros do TCP, três variantes são formuladas:

- TCP(ℓ) - versão do TCP com ℓ limitado por uma constante;
- TCP(r) - versão do TCP com r limitado por uma constante;
- TCP (ℓ, r) - versão do TCP com ℓ e r limitados por constantes.

Note que TCP$(0, 0)$ é equivalente a encontrar uma árvore contendo apenas vértices de W.

Teorema 1. TCP(ℓ) *é NP-completo.*

Prova. A prova utiliza uma redução do problema 3-SAT. Mostramos que, dada uma fórmula booleana F com m cláusulas e n variáveis, existe uma atribuição de valores `verdadeiro` e `falso` às variáveis de F que a satisfaçam se e somente se, no grafo associado G, existe uma árvore de conexão para um subconjunto específico $W \subseteq V(G)$ com $r = 2n + 1$ roteadores e $\ell = 2$ elos. (Para outros valores de ℓ, mesmo para $\ell = 0$, a prova pode ser facilmente adaptada.)

Dada uma fórmula booleana F com m cláusulas e n variáveis onde cada cláusula contém exatamente 3 literais, construimos um grafo associado G da seguinte forma:

- Para cada cláusula C_j de f, crie três vértices c_j^1, c_j^2, c_j^3 em G;
- Para cada variável X_i de F, crie um subgrafo g_i, composto pelos vértices x_i, t_{x_i}, f_{x_i}, $W_{x_i}^1$, $w_{x_i}^2$ e pelas arestas $(w_{x_i}^1, x_i)$, (x_i, t_{x_i}), $(t_{x_i}, w_{x_i}^2)$, $(w_{x_i}^2, f_{x_i})$, (f_{x_i}, x_i);
- Crie o subgrafo g_f que consiste nos vértices f, w_f^1, w_f^2, l^1, l^2 e nas arestas (f, l_1), (f, l_2), (l_1, w_f^1), (l_2, w_f^2);
- Para todo i, adicione uma aresta (f, x_i) a G;
- Adicione as arestas (t_{x_i}, c_j^1), (t_{x_i}, c_j^2) e (t_{x_i}, c_j^3) se e somente se a cláusula C_j contém o literal X_i;
- Adicione as arestas (f_{x_i}, c_j^1), (f_{x_i}, c_j^2) e (f_{x_i}, c_j^3) se e somente se a cláusula C_j contém o literal $\bar{X}_i$;
- Inclua em W vértices w_f^1, w_f^2, $w_{x_i}^1$, $w_{x_i}^2$, c_j^1, c_j^2 e c_j^3 (para todos os i, j).

Inicialmente, provaremos que, se F é satisfatível, então o grafo associado G contém uma árvore de conexão T com $r = 2n + 1$ roteadores e $\ell = 2$ elos. Por construção, o subgrafo g_f deve estar contido em T. Para todo i adicionamos a T as arestas (f, x_i) e

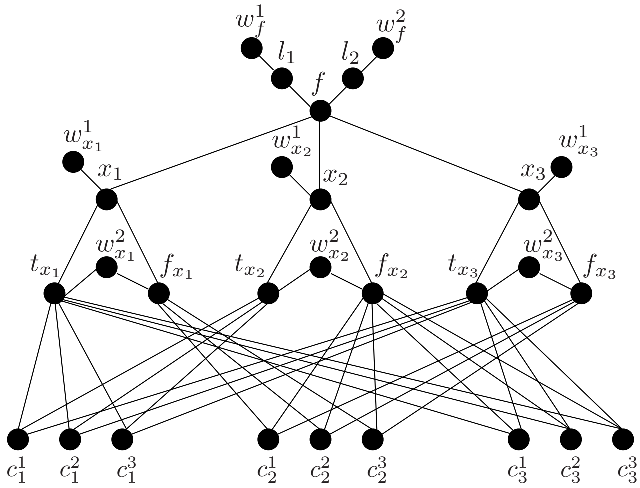
$(x_i, w^1_{x_i})$. Seja A uma atribuição de verdade para F. Suponha que cada variável torna pelo menos uma cláusula de F verdadeira (é fácil ver que tal atribuição sempre existe se F é satisfatível). Adicione a aresta (x_i, t_{x_i}) a T se a variável X_i =`verdadeiro` em A; caso contrário, adicione a aresta (x_i, f_{x_i}).

Se t_{x_i} (ou f_{x_i}) pertence a T, então adicione a aresta $(t_{x_i}, w^2_{x_i})$ (ou $(f_{x_i}, w^2_{x_i})$) a T. Para cada vértice t_{x_i} (ou f_{x_i}) em T, escolha um novo vértice c^k_j e insira uma aresta entre t_{x_i} (ou f_{x_i}) e c^k_j em T. (Os vértices c^k_j são folhas de T.) Finalmente, para cada vértice c^k_j ainda não incluído em T, insira em T uma aresta de G conectando c^k_j a algum vértice t_{x_i} ou f_{x_i} já em T. Neste ponto, T é uma árvore de conexão para W contendo os dois elos em g_f, e tendo como roteadores os vértices $f, x_1, x_2, \ldots, x_n$ e t_{x_i} ou f_{x_i} para cada x_i.

Por outro lado, se G contém uma árvore de conexão para W utilizando no máximo $r = 2n + 1$ roteadores e $\ell = 2$ elos, então por construção T usa l_1 e l_2 como elos. Como x_i é o único vizinho de $w^1_{x_i}$ em G, x_i é um roteador em T (para todo i). Cada vértice x_i em T é incidente a apenas uma das arestas (x_i, t_{x_i}) e (x_i, f_{x_i}), pois, caso contrário, isso implicaria a existência de mais do que $2n + 1$ roteadores. Em outras palavras, isso mostra que ou t_{x_i} ou f_{x_i} é um roteador para todo i por causa de $w^2_{x_i}$, pois esses vértices mais o vértice f e os vértices x_i totalizam no mínimo $2n + 1$ roteadores. Isto implica diretamente que f_{x_i} não está em T se t_{x_i} está, e vice-versa. Consequentemente, x_i tem grau 3 e é adjacente a f. Cada vértice c^k_j em T é adjacente a exatamente um vértice, caso contrário T conteria um ciclo. Assim, podemos construir uma atribuição A para F da seguinte forma: atribua X_i =`verdadeiro` se e somente se t_{x_i} pertence a T. Uma vez que cada c^k_j é adjacente a um vértice t_{x_i} ou f_{x_i}, pela construção A é uma atribuíção que satisfaz F. $\square$

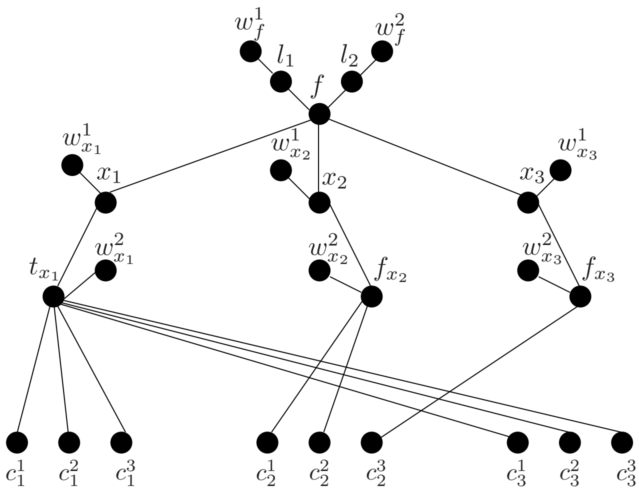
Note que para qualquer valor de l podemos construir um novo grafo para a redução. Para isso, basta adicionar novos vértices l_j e w^j_f para o mesmo grafo, conforme feito para os vértices l_1, l_2, w^1_f e w^2_f.

A Figura 2 mostra o grafo G construído a partir da fórmula $F = (x_1 + x_2 + x_3).(\overline{x_1} + \overline{x_2} + \overline{x_3}).(x_1 + \overline{x_2} + x_3)$, e a Figura 3 ilustra a árvore de conexão T de G para W usando $\ell = 2$ elos e $r = 7$ roteadores.



(a)

Figura 2. O grafo G associado à fórmula F.



(b)

Figura 3. Uma árvore de conexão T para W.

Teorema 2. TCP(r) *é NP-completo.*

Prova. Esta prova utiliza uma redução a partir de um problema muito interessante descrito em [Garey (1990)]:

DEGREE CONSTRAINED SPANNING TREE - DCST

Instância: Um conexo grafo H e um inteiro positivo k.

Pergunta: Existe uma árvore geradora para H em que nenhum vértice tem grau maior do que k?

De acordo com [Garey (1990)], DCST é NP-completo para qualquer inteiro fixo $k \geq 2$. Denote por DCST(2) a versão de DCST para $k = 2$, que é equivalente ao PROBLEMA DO CAMINHO HAMILTONIANO [Garey (1990)].

A partir de uma instância H de DCST(2), crie uma instância G de TCP(r) para $r = 0$ da forma a seguir. (A prova pode ser facilmente adaptada para outros valores de r).

(1) Inicialmente, faça $V(G) = V(H)$.
(2) Para cada vértice $v_i \in V(G)$ faça: (a) crie dois vértices v'_i e v''_i em G; (b) atribua a v'_i e v''_i em G a mesma vizinhança de v_i em H; (c) adicione as arestas (v_i, v'_i) e (v_i, v''_i) em G.
(3) Faça $\ell = 2n$, $r = 0$ e $W = \{v_i \mid 1 \leq i \leq n\}$.

Note que todos os vértices em W têm grau dois em G.

Se H contém uma árvore geradora T' em que nenhum vértice tem grau maior do que 2, podemos construir uma árvore de conexão T para W usando no máximo $\ell = 2n$ elos e $r = 0$ roteadores. Para cada aresta (v_i, v_j) em T', basta adicionar as arestas $(v_i, v'_i), (v'_i, v'_j), (v'_j, v_j)$ a T. (Se v'_i ou v'_j já pertence a T, substitua-o por v''_i ou v''_j, respectivamente).

Por outro lado, a partir de uma árvore de conexão T para W com no máximo $2n$ elos e sem roteadores, podemos construir uma árvore geradora T' de H (que é um caminho Hamiltoniano), adicionando uma aresta (v_i, v_j) a T' se e somente se existe um caminho em T entre v_i e v_j cujos vértices internos são todos elos. $\square$

Teorema 3. TCP(ℓ, r) *pode ser solucionado em tempo polinomial.*

Prova. A prova é baseada no seguinte algoritmo: para cada par L, R de subconjuntos tais que $L, R \subseteq V(G) \backslash W$, $L \cap R = \emptyset$, $|L| \leq \ell$ e $|R| \leq r$, siga os passos abaixo:

(1) Faça $G' = G[W \cup L \cup R]$
(2) **Para cada** floresta geradora T de G' obtida escolhendo duas arestas de G' incidentes a v para cada $v \in L$ e três arestas de G' incidentes a w para cada $w \in R$, onde $deg_T(v) = 2$ para cada $v \in L$ **faça**

Se T é conexo **então** retorne T **senão**
  Seja $S = E(G') \backslash (E(T) \cup E(G'[L]))$;
  Enquanto T não for conexo ou S não estiver vazio **faça**
    Remova uma aresta e de S e insira e em T, desde que $T + e$ não contenha ciclo.
    Se T é conexo **então** retorne T

O algoritmo acima retorna uma árvore de conexão para W utilizando os vértices em L como elos e os vértices em R como roteadores, caso esta árvore exista. O algoritmo considera $O(n^{\ell}n^{r})$ pares de subconjuntos L, R. A linha (2) considera $O(n^{2\ell}n^{3r})$ subgrafos T, onde um tempo $O(n)$ é necessário para verificar se cada T é um subgrafo gerador e acíclico. As restantes operações são facilmente feitas feitas em tempo $O(n+m)$. A complexidade geral é, portanto, $O(n^{3\ell+4r+1}m)$. $\square$

3 Conexão de Nós Estritamente Terminais

O problema TCP consiste em conectar um conjunto $W \subseteq V(G)$ utilizando no máximo ℓ elos e r roteadores. Sem perda de generalidade, podemos considerar que em uma árvore de conexão T todas as suas folhas são vértices pertencentes a W. Como podemos observar, não necessariamente todos os vértices de W são folhas em uma árvore de conexão; há árvores onde vértices em W possuem grau maior ou igual a dois, fazendo portanto a função de elos ou roteadores.

No entanto, há aplicações em que espera-se que os nós terminais sejam folhas da árvore de conexão, ou seja, estes não podem simular um elo ou um roteador. Denotamos por E-TCP uma versão do problema TCP onde deseja-se determinar se existe uma árvore de conexão T para um conjunto W de vértices utilizando no máximo ℓ elos e r roteadores, tal que todos os vértices de W são folhas em T.

CONEXÃO DE NÓS ESTRITAMENTE TERMINAIS – E-TCP
Instância: Um grafo conexo G, um subconjunto $W \subseteq V(G)$, e dois inteiros não negativos ℓ e r.
Questão: O grafo G contém uma árvore de conexão T com no máximo ℓ elos e r roteadores, tal que $V(T) \cap W$ é o conjunto das folhas de T?

Lema 4. TCP $\propto$ E-TCP.

Prova. Seja (G, W, r, ℓ) uma instância do problema TCP. Construímos uma instância (G', W', r', ℓ') do problema E-TCP da seguinte forma:

(i) Faça $G' = G$ e $W' = W$;
(ii) Para cada vértice $v_i \in W'$ crie um vértice v_i' com a mesma vizinhança de v_i em G', remova todas arestas incidentes a v_i e adicione a aresta (v_i, v_i') a G';
(iii) Para cada vértice v_i' recentemente criado, crie um segundo vértice v_i'', adicione a aresta (v_i', v_i''), e faça $W' = W' \cup \{v_i''\}$;
(iv) Faça $\ell' = \ell$ e $r' = |W| + r$.

Seja $S = V(G') \backslash (W \cup \{v_i'' : \forall v_i'' \in V(G')\})$. Note que $G[S]$ é um grafo isomorfo a G. Além disso, todo vértice v_i' será roteador em qualquer árvore de conexão para W' em G'. Sendo assim, é fácil ver que G possui uma árvore de conexão para W utilizando ℓ elos e r roteadores se e somente se G' possui uma árvore que conecte W' utilizando $\ell' = \ell$ elos e $r' = |W| + r$ roteadores. $\square$

Corolário 5. E-TCP(ℓ) *é NP-completo.*

Prova. Segue diretamente da prova apresentada no Lema 4, ou alternativamente, da prova apresentada no Teorema 1.

Lema 6. E-TCP(ℓ, r) *pode ser solucionado em tempo polinomial.*

Prova. O algoritmo apresentado no Teorema 3 pode ser facilmente adaptado para preservar a condição dos vértices em W serem folhas. □

Teorema 7. TCP(ℓ) *e* E-TCP(ℓ) *permanecem NP-completos em grafos com grau máximo* 6.

Prova. O Teorema 1 demonstrou por uma redução do problema 3-SAT que TCP(ℓ) e E-TCP(ℓ) são problemas NP-completos. No entanto, o problema 3-SAT permanece NP-completo mesmo quando cada variável ocorre no máximo três vezes e cada literal ocorre no máximo duas vezes [Garey (1990)]. Se considerarmos essa restrição em nossa construção, temos que o único vértice com grau maior que 6 no grafo obtido é o vértice f. Todavia, podemos substituir f na construção por uma subgrafo T_f composto por uma árvore binária completa com n (número de variáveis) folhas, sendo cada folha vizinha de um vértice x_i e a raiz de T_f vizinha de l_1 e l_2. Para finalizar a construção, cada vértice em T_f diferente da raiz será vizinho de dois novos vértices (vértices pendentes) que serão adicionados ao conjunto W.

Neste ponto, o grafo resultante G possui grau máximo 6 e possuirá uma árvore que conecta W utilizando $2n + |V(T_f) \setminus W|$ roteadores e 2 elos se e somente se a fórmula F associada for satisfatível. □

4 Algoritmo Parametrizado

A teoria de complexidade parametrizada [Downey (1995)], proposta em 1995, surgiu como uma alternativa promissora para se trabalhar com problemas NP-difíceis, que passam a ter um conjunto K de parâmetros fixos adicionais à entrada. O interesse em parametrizar tais problemas se deve ao fato de que, em muitos casos, somente uma pequena faixa de valores de parâmetro é realmente importante na prática. Logo, a intratabilidade (aparente) desses problemas no caso geral é indevidamente pessimista. Analisando-se mais profundamente a estrutura da entrada (considerando-a com um conjunto de parâmetros adicionais), tenta-se limitar a explosão combinatória aparentemente inevitável na solução do problema.

Muitas vezes, determinar o custo mínimo para a produção de uma empresa, ou o lucro máximo que esta poderá obter, por exemplo, são problemas difíceis de serem resolvidos. No entanto, na prática, as empresas geralmente necessitam determinar apenas se é possível efetuar a produção com um determinado orçamento ou verificar se é possível cumprir uma determinada meta de venda/lucro. Sendo assim, é possível adicionar aos problemas de custo mínimo ou lucro máximo destas empresas parâmetros fixos adicionais, como um orçamento ou uma meta, dando origem às suas versões parametrizadas, que podem ser mais fáceis de serem solucionadas (menos complexas) e ao mesmo tempo satisfazer as necessidades requeridas. Perante estes fatos, observa-se que a Teoria da Complexidade Parametrizada modela perfeitamente esta realidade; muitos problemas considerados intratáveis (a menos que $P = NP$) sob o ponto de vista teórico, na prática são tratáveis por parâmetro fixo.

Segundo a complexidade parametrizada [Flum (2006), Niedermeier (2006)], dado um conjunto K de parâmetros (fixados com valor constante), um problema NP-difícil Π é dito *tratável por parâmetro fixo* (FPT) para K se, uma vez adicionando K à sua formulação, for possível desenvolver um algoritmo que solucione Π em tempo $f(K)n^{\alpha}$, onde n é o tamanho da entrada, $f(K)$ é uma função arbitrária (possivelmente exponencial) em K e α é uma constante independente de n e K.

Definição 8. *Seja Π um problema. Então $\Pi \in FPT$ com relação a um conjunto de parâmetros K se e somente se existe um algoritmo A para solucionar Π com tempo de execução $f(K).n^{O(1)}$, onde $f()$ é uma função arbitrária.*

Dentre os métodos para construção de um algoritmo FPT, o método da árvore de busca limitada é provavelmente o mais fácil de ser aplicado, e é baseado no seguinte fato. Muitos problemas combinatórios podem ser solucionados através de algoritmos que podem ser decompostos em duas partes distintas:

- Primeiramente, com o algoritmo computamos, talvez de maneira ineficiente, algum espaço de busca que é com frequência uma árvore de busca de tamanho exponencial.
- Em seguida, executamos algum algoritmo relativamente eficiente para os ramos das árvores, em geral baseado na busca em profundidade.

A complexidade exponencial do pior caso de tais algoritmos provém de instâncias que necessitam percorrer completamente a árvore. Para o propósito de parametrização, a principal observação é que para muitos problemas parametrizados, o tamanho da árvore depende somente dos parâmetros. Desta forma, para parâmetros fixos, o espaço de busca possui tamanho constante e o algoritmo é então eficiente para cada parâmetro fixo k.

Teorema 9. E-TCP *pode ser solucionado em tempo $f(\Delta, r, \ell).n^{O(1)}$.*

Prova. Construímos uma árvore enraizada de altura $r + \ell + 1$ da seguinte forma.

Rotule a raiz da árvore com dois conjuntos vazios. Escolha um vértice $w_i \in W$; note que qualquer árvore de conexão de W (onde W consiste nas folhas) deve conter para o vértice $w_i \in W$ uma aresta $(w_i, v) \in E(G)$ tal que $v \notin W$. Sendo assim, criamos nós-filhos da raiz correspondendo a estas $d(w_i)$ possibilidades. Assim, cada nó filho é rotulado com os conjuntos $\{(w_i, v)\}$ e $\{w_i\}$. O conjunto de arestas do rótulo de um nó representa uma "possível" subárvore de conexão para W, e o conjunto de vértices representa os vértices que não precisam ser mais analisados na construção da árvore de conexão.

A construção da árvore de conexão seguirá uma busca em largura. Assim, no segundo passo, será analisado o primeiro nó filho criado a partir da raiz. Neste ponto, seja v o vértice tal que (w_i, v) pertence ao conjunto de arestas do rótulo do nó em análise e $v \notin W$. Como v não será terminal temos duas possibilidades: (i) v é um elo na árvore de conexão; neste caso v possuirá mais uma aresta no conjunto de arestas da árvore de conexão; (ii) v é um roteador na árvore de conexão; neste caso v possuirá pelo menos mais duas arestas no conjunto de arestas da árvore de conexão. Sendo assim, criamos nós-filhos do nó corrente para cada uma destas $2^{\Delta-1}$ possibilidades, onde cada nó-filho será rotulado com o conjunto de arestas de seu nó pai acrescido de um subconjunto distinto de arestas de v. O conjunto de vértices do rótulo deste nó criado será formado pelo conjunto de vértices do rótulo do seu nó-pai acrescido do vértice v.

De uma forma geral, para cada nó rotulado com um conjunto S de arestas e um conjunto X de vértices, escolhemos um nó em $V(S) \setminus X$ e criamos nós-filhos para cada uma das possibilidades deste nó (ser elo ou roteador), desde que as arestas adicionadas ao conjunto de arestas não gerem ciclos.

Se criamos um nó de altura no máximo $r + \ell + 1$ na árvore de busca e o conjunto de vértices do rótulo do nó contém W então o conjunto de arestas do rótulo deste nó forma uma árvore de conexão utilizando no máximo $r + \ell$ roteadores e elos. Desta forma, a construção da árvore de busca terá a seguinte condição de parada da ramificação de um nó: (a) atingir altura igual a $r + \ell + 1$; (b) o rótulo possuir mais do que r roteadores na árvore de conexão formada pelo seu conjunto de arestas; (c) o rótulo possuir mais do que ℓ elos na árvore de conexão formada pelo seu conjunto de arestas.

Dada a nossa árvore de busca, haverá uma árvore de conexão satisfazendo as condições exigidas para um determinado conjunto W de vértices se e somente se alguma das folhas desta árvore representa uma configuração contendo no máximo ℓ elos, r roteadores e conectando todos os vértices em W.

Como a árvore de busca possui altura no máximo $r + \ell + 1$ e grau máximo $2^{\Delta-1}$ a complexidade de tempo do nosso algoritmo será $O((2^{\Delta-1})^{r+\ell+1}m)$, como requerido. □

Corolário 10. E-TCP *pertence à classe FPT.*

5 Conclusão

Problemas relacionados a conexão de nós em uma rede são problemas combinatórios de grande interesse, e possuem diversas aplicações práticas.

Neste trabalho analisamos o problema de decidir, em uma determinada rede, se existe uma árvore de conexão que conecta um determinado número de terminais utilizando um número restrito de roteadores e elos.

Dentre os resultados obtidos, provamos que dado um grafo G e um conjunto $W \subseteq$

$V(G)$, determinar se existe em G uma árvore que conecta W utilizando no máximo r roteadores e ℓ elos é NP-completo quando r ou ℓ são constantes. Porém, quando ambos os valores são limitados por constantes, o problema por ser solucionado em tempo polinomial. Além disso, analisamos a variação do problema onde todos os vértices em W devem ser folhas na árvore de conexão. Dentre os resultados obtidos, provamos que o problema permanece NP-completo mesmo se o grafo de entrada possui grau máximo 6 e o número de elos é fixado por uma constante. Também apresentamos um algoritmo parametrizado que prova que esta variação do problema é tratável por parâmetro fixo quando o grau máximo, o número de roteadores e o numero de elos são utilizados como parâmetros.

Referências

Downey, R. G. and Fellows, M. R. (1995). Fixed-parameter tractability and completeness I: Basic results. *SIAM J. Comput.*, 24(4):873–921.

Dreyfus, S. and Wagner, R. (1972). The Steiner problem in graphs. *Networks*, 1:195–207.

Flum, J. and Grohe, M. (2006). *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Hakimi, S. (1971). Steiner's problem in graphs and its implications. *Networks*, pages 113–133.

Levin, A. Y. (1971). Algorithm for shortest connection of a group of graph vertices. *Soviet Mathematical Doklady*, 12:1477–1481.

Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *The Bell Systems Technical Journal*, 36(6):1389–1401.