



Variantes do Problema da Árvore de Steiner Aplicados à Internet das Coisas

Rodolfo Alves de Oliveira

UFF - Dep. de Ciências Exatas, Biológicas e da Terra
Santo Antônio de Pádua - RJ
rodolfo.oliveira@infes.uff.br

Marcelo Pitanga Alves

UNIABEU - Campus 6
Nova Iguaçu, RJ
mpitanga@gmail.com

Andressa Martins Moraes

UNIABEU - Campus 6
Nova Iguaçu, RJ
andressa.meryl@gmail.com

RESUMO

Este trabalho tem como objetivo o estudo de problemas análogos ao problema da árvore de Steiner em grafos com aplicações no roteamento de rede da Internet das Coisas. Assim, definimos problemas que garantam uma conexão eficiente entre os dispositivos eletrônicos numa rede e analisamos suas respectivas complexidades. Embora os problemas de decisão aqui estudados sejam NP-completos, como estão demonstrados, conseguimos resultados interessantes em relação ao tempo de execução, tornando-os de cunho polinomial ao fixarmos alguns parâmetros por constantes.

PALAVRAS CHAVE. Árvore de Steiner, Grafos, Internet das Coisas, Roteamento, Complexidade.

ABSTRACT

This paper aims at the study of problems analogous to the Steiner tree problem in graphs for applying in IoT network routing. Hence, we define problems that guarantee an efficient connection among electronic devices in the network, and we analyze their respective complexities. Although the decision problems studied here are NP-complete, we get satisfactory results regarding the execution time, turning them into polynomial time by defining some parameters as constants.

KEYWORDS. Steiner tree, Graphs, Internet of Things, Network Routing, Complexity.



1. Introdução

Imagine um mundo onde os dispositivos eletrônicos que já fazem parte do nosso cotidiano se comunicassem por uma rede de forma independente, sem ação humana, de modo a tornar o nosso dia a dia mais fácil e eficiente. Por exemplo, considere uma geladeira conectada ao seu smartphone por intermédio da internet, então com uma programação adequada poderíamos fazer com que a geladeira envie informações de aviso sobre produtos que estão prestes a vencer, ou até mesmo manter as informações dos produtos que nela contém. Além disso, a geladeira também poderia pesquisar na web quais mercados oferecem os melhores preços para os itens que estão faltando e enviá-los para o seu smartphone. Claro que a interatividade na troca de informações é interessante para um número considerável de dispositivos, como por exemplo, numa plantação onde podemos instalar dezenas de sensores capazes de medir o nível hídrico e níveis de nutrientes do solo, e dependendo dos índices, que sejam capazes de acionar outros dispositivos para o tratamento do solo. Esses dispositivos conectados em rede e outros que garantem a confiabilidade e a segurança na transmissão de dados constituem a Internet das Coisas. Para mais informações sobre o tema indicamos [Atzori et al., 2010].

De [Lin e Liu, 2010], na comunicação de rede sem fio se destacam diversos padrões, tais como a tecnologia de banda ultralarga que utiliza transmissão via rádio, a tecnologia Bluetooth e a tecnologia ZigBee. Este último, o padrão ZigBee, se destaca na área de rede pessoal (wireless personal area network - WPAN) e, conseqüentemente, na Internet das Coisas. Embora seja um pouco lento comparado com outras tecnologias, o padrão ZigBee tem a capacidade de conectar milhares de dispositivos numa única rede e possui um custo baixo, se comparado com as outras tecnologias citadas. O padrão é constituído por três elementos:

- ZigBee Dispositivo Final (ZDF): o dispositivo que se deseja conectar a rede, como um sensor, por exemplo;
- ZigBee Roteador (ZR): responsável por rotear o envio e recebimento de pacotes na rede;
- ZigBee Coordenador (ZC): responsável por criar, gerenciar, rotear dados, armazenar informações, comunicar com outras redes e, além disso, determinar parâmetros como o número máximo de elementos que podem ser conectados num roteador e a profundidade da rede (distância máxima do ZC a qualquer outro dispositivo ZDF).

Como podemos observar, o dispositivo ZC é muito importante na rede, principalmente por ele definir a faixa de endereços que limitam o número máximo de dispositivos ZR's e o número máximo de dispositivos ZDF's que ele e qualquer outro dispositivo ZR podem endereçar. Além disso, este padrão adota três tipos de topologias: *árvore*, *malha* e *cluster tree*. Veja na Figura 1 exemplos das topologias citadas.

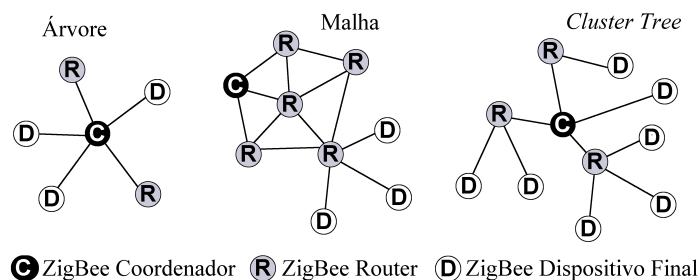


Figura 1: Topologias ZigBee.



A troca de informações eficiente de uma rede depende de diversos fatores. Um deles é o roteamento dos elementos que a compõe. Não há como não falar de conexão ótima sem mencionar o problema da árvore de Steiner. O problema da árvore de Steiner foi definido originalmente no espaço euclidiano, onde no princípio eram dados três pontos no espaço e desejava-se encontrar o quarto ponto, de modo que a soma das distâncias das semirretas ligando todos esses quatro pontos seja mínima. Esse problema clássico vem sendo discutido desde o século XVIII conforme em [Brazil et al., 2014], mas foi em 1943 que ganhou maior notoriedade após a publicação de *What Is Mathematics?* [Courant e Robbins, 1943]. Além disso, uma generalização do problema é definida como: dado um conjunto S de pontos no plano, desafia-se a encontrar um novo conjunto de pontos S' que interligue S , de forma que seja mínima a soma das distâncias das semirretas entre todos os pontos de $S \cup S'$. Isso realmente é um desafio, já que este problema é NP-difícil por [Garey e Johnson, 1979]. Os pontos de S são chamados de *terminais* e os novos pontos encontrados em S' são chamados de *pontos de Steiner*.

Conforme em [Hakimi, 1971], em grafos também há uma versão do problema da árvore de Steiner. O problema consiste em: dado um grafo conexo $G = (V(G), E(G))$ com pesos $w(e)$ não negativos associados a cada $e \in E$ e um subconjunto de vértices W de G , deve-se encontrar um subgrafo conexo T contendo W , onde $\sum_{e \in E(T)} w(e)$ seja mínimo. É fácil notar que T é sempre uma árvore. Análogo ao espaço euclidiano, os vértices de W são chamados *terminais* e os vértices $V(T) \setminus W$ são chamados de *vértices de steiner*. Na Figura 2 temos que dado o grafo em (i) com $W = \{a, b, c\}$, uma árvore de Steiner para W está em (iii), pois não há como conectar W com custo inferior a 9.

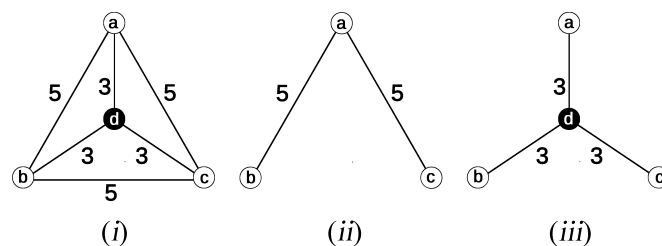


Figura 2: Em (i) temos um grafo, onde $W = \{a, b, c\}$. Em (ii) temos uma árvore geradora contendo W com custo total 10 ($5+5$), já em (iii) temos uma subgrafo conexo contendo W com custo total 9 ($3+3+3$).

O problema da árvore de Steiner possui uma versão de decisão definida como: *dado um grafo G conexo e ponderado, um subconjunto $W \subseteq V(G)$ e um inteiro não negativo λ , pergunta-se, Existe uma árvore T em G conectando W tal que $\sum_{e \in E(T)} w(e) \leq \lambda$?*

Infelizmente, o problema da árvore de Steiner é NP-difícil em grafos assim como no espaço euclidiano [Garey e Johnson, 1979]. No entanto, quando o número de terminais é fixado por uma constante, existe um algoritmo polinomial desenvolvido por [Dreyfus e Wagner, 1971] que resolve o problema da árvore de Steiner em grafos na ordem $O(n^3 + n^2 2^{k-1} + n 3^{k-1})$, onde n é o número de vértices do grafo G e k o tamanho conjunto W .

O problema da árvore de Steiner possui diversas aplicações na criação de conexões ótimas, isto é, conexões que possuem custo mínimo entre os elementos envolvidos. Como exemplos, na construção de VLSI [Kahng, 1991], estudos em aproximações genéticas com construção de árvores filogenéticas [Weng et al., 2012] e conexão em redes multicast [Hakimi, 1971].

Uma justificativa para introduzir o problema da árvore de Steiner ao padrão ZigBee se deve ao fato que os algoritmos implementados numa rede ZigBee geralmente utilizam para o roteamento algoritmos que se preocupam apenas com a profundidade da rede, não na conexão ótima entre seus elementos. Por exemplo, ao considerarmos ZC o vértice a , ZR o vértice d e dois ZDF's como os vértices b e c na Figura 2, e assumir a profundidade máxima de rede igual a 6, o algoritmo implementado no roteamento encontrará como solução a árvore do item (ii) para o grafo do item



(i), e como já mencionamos ela não é uma conexão ótima, na qual é dada uma árvore de Steiner no item (iii). Além disso, ainda observando a Figura 2, temos que a comunicação entre b e c dada pela conexão em (ii) é dispendiosa se compararmos com a conexão do item (iii).

Em [Dourado et al., 2014], os autores definiram uma árvore de conexão que possui similaridade com o problema da árvore de Steiner, definido como: dado um grafo G , um subconjunto de vértices $W \subseteq V(G)$ e dois inteiros não negativos ℓ e r , deseja-se verificar se existe uma árvore T em G conectando W , tal que em T existem no máximo ℓ vértices de grau 2 e r vértices de grau maior que 2, todos diferentes de W . Assim, de forma análoga, definimos um problema de conexão propício ao que desejamos.

PROBLEMA DA ÁRVORE DE CONEXÃO POR 2 PARÂMETROS (2PCT)

Instância: Um grafo G conexo e ponderado, um subconjunto $W \subseteq V(G)$, um vértice $c \in V(G) \setminus W$ e os inteiros não negativos α e β .

Pergunta: Existe uma árvore T em G conectando $W \cup \{c\}$ cujos vértices em T não possuem grau superior a α , a distância de c a qualquer outro vértice de W em T seja no máximo β e todos os vértices de W são folhas em T ?

Podemos observar que esse problema contempla em sua essência elementos comuns ao padrão ZigBee, com W representando os ZDF's, o vértice c representando o ZC e os vértices restantes sendo os ZR's disponíveis para a conexão de $W \cup \{c\}$. Já o inteiro não negativo α é o valor estipulado por ZC que limita o máximo de dispositivos conectados aos ZC e ZR's, e o inteiro não negativo β o valor de profundidade da rede. Assim, para um dado conjunto de elementos do ZigBee distribuídos numa região, poderíamos construir um grafo ponderado G representando esses elementos, onde os pesos poderiam corresponder ao tempo de resposta de um ponto ao outro, distância entre elementos etc., e disso encontraríamos uma conexão para alguns ZDF's representados por W conforme a topologia de árvore ou *cluster tree*, respeitando os valores de α e β dados.

Vale ressaltar que 2PCT é um problema de conexão similar ao problema de decisão da árvore de Steiner, porém não preconiza a obrigatoriedade de T ser de custo limitado, apenas limita o grau dos vértices em $V(T) \setminus W$ e a distância de c aos outros vértices em W , além dos vértices de W serem folhas de T . No entanto, o que desejamos são árvores de conexões ótimas, isto é, árvores de custo mínimo que satisfaçam o problema 2PCT. Assim sendo, considere a versão de decisão do problema de árvores de conexões com custo limitado a seguir:

PROBLEMA DA ÁRVORE DE STEINER POR 2 PARÂMETROS (2PST)

Instância: Um grafo G conexo e ponderado, um subconjunto $W \subseteq V(G)$, um vértice $c \in V(G) \setminus W$ e os inteiros não negativos α , β e λ .

Pergunta: Existe uma árvore T em G conectando $W \cup \{c\}$ tal que $\sum_{e \in E(S)} w(e) \leq \lambda$, cujos vértices em T não possuem grau superior a α , a distância de c a qualquer outro vértice de W em T seja no máximo β e todos os vértices de W são folhas em T ?

Todavia, uma versão de otimização apreciada para o 2PST é definida para valores mínimos de λ . Então, qualquer resultado de complexidade referente ao problema 2PST refletirá diretamente a sua versão de otimização.

2. Análises de Complexidade

Nesta seção veremos alguns resultados importantes para os problemas definidos. Mas antes, considere os grafos $G = (V(G), E(G))$ conexos e ponderados, onde $|V(G)| = n$ e $|E(G)| = m$. Denote também que para qualquer par de vértices u e v de G , $dist(u, v)$ representa a distância entre u e v no grafo G .



Proposição 1 2PCT está em NP.

Prova: Dado um certificado, isto é, uma árvore T , podemos efetuar a busca em largura em T a partir de c no conhecido tempo $O(n + m)$ e verificar se T respeita os limites máximos de α e β . Portanto, 2PCT está em NP. \square

Agora mostraremos que 2PCT é NP-completo. Para isso, faremos uso de argumentos que comprovam a NP-completude de 2PCT.

Teorema 2 O problema 2PCT é NP-completo para $\alpha = 3$ fixo.

Prova: O problema 2PCT está em NP pela **Proposição 1**. Para redução polinomial, faremos uso do problema do caminho hamiltoniano (HP) em grafos que é conhecido ser NP-completo [Garey e Johnson, 1979]. O problema consiste em: *dado um grafo G , pergunta-se se existe um caminho¹ que passe por todos os vértices?*

Seja $\langle G \rangle$ uma entrada genérica do HP. Usaremos a função transformação $f : \langle G \rangle \rightarrow \langle G', W, c, \alpha, \beta \rangle$ descrita a seguir:

- i.* Conjunto de Vértices: $V(G') = V(G) \cup \{x_u : \text{para } u \in V(G)\}$;
- ii.* Conjunto de Arestas: $E(G') = E(G) \cup \{x_u u : \text{para } u \in V(G)\}$;
- iii.* Atribua: $w(e) = 1, \forall e \in E(G')$;
- iv.* Conjunto de Terminais: $W = \{x_u : u \in V(G)\}$;
- v.* O vértice c pode ser qualquer vértice de $V(G)$;
- vi.* $\alpha = 3$;
- vii.* $\beta = n$.

Na Figura 3, temos uma ideia da construção do grafo G' de G e a obtenção da solução de um caminho hamiltoniano P a partir de uma árvore de conexão T encontrada em G' .

Agora, provaremos que HP diz *sim* para uma instância $\langle G \rangle$ se, e somente se, também for *sim* para 2PCT com a entrada $\langle G', W, c, \alpha, \beta \rangle$:

(HP \Rightarrow 2PCT)

Se para $\langle G \rangle$ existe um caminho hamiltoniano P , temos que $V(G) = V(P)$ e que cada vértice em P tem grau no máximo 2. Então, como existe a aresta $x_u u$ em $E(G')$, para cada u em P , podemos conectar todos os vértices W através de P acrescentando exatamente uma unidade no grau de cada vértice em P . Disso, temos uma árvore que conecta W com no máximo grau 3 e todos os vértices de W são folhas. Além disso, o vértice $c \in V(G) = V(P)$, então qualquer outro vértice $u \in V(G)$ tem distância no máximo $n - 1$ de c , caso ambos seja extremos distintos de P . Então, a distância de c a x_u é limitada por n . Portanto, temos um certificado de *sim* para 2PCT.

(HP \Leftarrow 2PCT)

Agora, considere que para $\langle G', W, c, \alpha, \beta \rangle$ temos uma árvore T em G' que conecta todos os vértices de $W \cup \{c\}$ com grau no máximo 3. Por construção, todos os vértices de W são folhas. Então, $V(T) \setminus W$ é um caminho, pois é conexo e com exatamente um grau a menos cada um dos vértices em T . Portanto, temos um certificado *sim* para HP.

¹Um passeio que não se repete vértice.

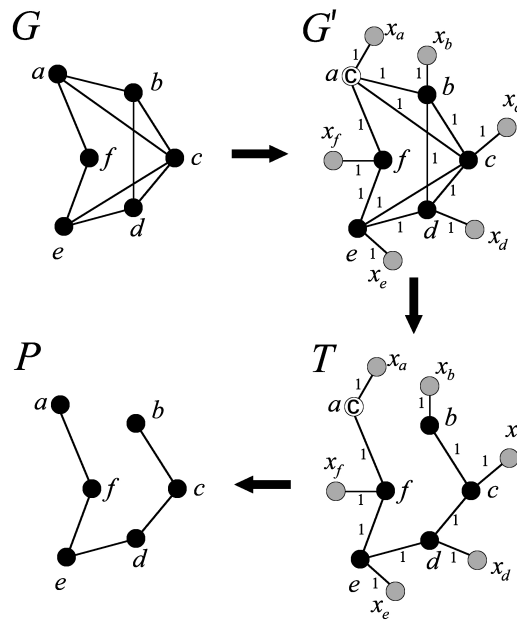


Figura 3: Ideia da construção de G' , onde os vértices de W estão em cinza e vértice c está em branco (o vértice a) e a solução do caminho hamiltoniano P em G obtido de uma árvore de conexão T existente em G' .

Assim, o problema 2PCT é NP-completo. \square

Além disso, podemos concluir o seguinte resultado:

Corolário 3 *O problema 2PCT é NP-completo para qualquer que seja $\alpha \geq 3$ e inteiro não negativo β fixos.*

Prova: Para $\alpha = 3$, segue do **Teorema 2**. Agora, para $\alpha > 4$, basta seguir o mesmo raciocínio de construção do **Teorema 2**, onde desejado um $\alpha > 3$, crie os vértices $x_u^1, x_u^2, \dots, x_u^{\alpha-2}$ adjacente a cada u de $V(G)$ e os insiram em W . Com isso, usando os mesmos argumentos do **Teorema 2** para a existência de um caminho hamiltoniano, podemos concluir a NP-completude para todos esses α 's.

No caso do β , podemos usar a mesma construção do **Teorema 2**, mas atribuindo a cada aresta peso zero, já que podemos atribuir valores não negativos para as arestas. Com isso, para qualquer $\alpha \geq 3$ e inteiro não negativo β fixos, temos que o problema é NP-completo. \square

Assim, do **Corolário 3**, podemos concluir que o problema é NP-completo para valores fixos de $\alpha \geq 3$. No entanto, o problema 2PCT é polinomial quando o valor de α é ilimitado. No algoritmo a seguir mostramos o Algoritmo de Dijkstra [Dijkstra, 1959] modificado para encontrar uma árvore de conexão por 2 parâmetros.



Algoritmo 1 2PCT com α ilimitado.

Entradas: O conjunto de vértices $V(G)$;
A matriz de adjacência A , onde para $i, j \in V(G)$ temos $A[i, j] = w((i, j))$;
O subconjunto de vértices W ;
O vértice $c \in V(G) \setminus W$;
O valor constante de β .

Est. Auxiliares: Um vetor d , onde para cada $i \in V(G)$ temos que $d[i] = \text{dist}(c, i)$;
Um Min-Heap H de vértices de G organizado de forma não decrescente para valores de d ;
Um vetor ant , onde cada $i \in V(G)$ temos que $ant[i] = u$ implica u antecessor de i no caminho de c a i ;
Um conjunto S para sinalizar os vértices já verificados pelo algoritmo;

Saída: Se existir, uma árvore de conexão T , onde todo vértice $z \in W$ é folha e $\text{dist}(c, z) \leq \beta$, caso contrário \emptyset .

Início do Algoritmo

```
1.  para cada  $u \in V(G) \setminus \{c\}$  faça
2.       $d[u] \leftarrow \infty$  e  $ant[u] \leftarrow \emptyset$ 
3.   $d[c] \leftarrow 0$ ,  $ant[c] \leftarrow c$ 
4.   $H \leftarrow \text{INSERT}(V(G))$            % insere todos os vértices no Min-Heap H
5.  enquanto  $H \neq \emptyset$  faça
6.       $u \leftarrow \text{EXTRACT}(H)$        % extrai o elemento no topo do Min-Heap H
7.       $S \leftarrow S \cup \{u\}$ 
8.      se  $u \notin W$  então
9.          para cada  $v$  vizinho de  $u$  e  $v \notin S$  faça
10.             se  $d[u] + A[u, v] < d[v]$  então
11.                 se  $d[u] + A[u, v] \leq \beta$  então
12.                      $d[v] \leftarrow d[u] + A[u, v]$ ,  $ant[v] \leftarrow u$ 
13.                     RECTIFY( $H, v$ )           % corrige o Min-Heap H após as atualizações de  $d[v]$ 
14.              $V(T) \leftarrow \{c\}$ ,  $E(T) \leftarrow \emptyset$            % aqui se inicia a construção da árvore T
15.         para cada  $z \in W$  faça
16.             se  $ant[z] \neq \emptyset$  então           % verifica se z foi alcançado por um caminho no máximo  $\beta$ 
17.                  $v \leftarrow z$ 
18.                 senão retorne  $\emptyset$            % não temos a árvore T desejada
19.             enquanto  $v \notin V(T)$  faça
20.                  $V(T) \leftarrow V(T) \cup \{v\}$ ,  $E(T) \leftarrow E(T) \cup \{(ant[v], v)\}$ ,  $v \leftarrow ant[v]$ 
21.         retorne  $T$ 
```

Fim do Algoritmo

Teorema 4 O Algoritmo 1 funciona no tempo $O(m \log n + n \log n + 3n - 1)$.

Prova: Como já foi mencionado, o Algoritmo 1 é o algoritmo de Dijkstra com leves modificações. Uma delas está na linha 8 do algoritmo que nos garante que teremos uma árvore T contendo $W \cup \{c\}$, de modo que os vértices de W são folhas de T e na linha 11 temos a garantia esses vértices não distam mais do que β do vértice c em suas ramificações.

Assim, se para alguma entrada do problema 2PCT com α ilimitado existir uma árvore T que o satisfaça, então o algoritmo será capaz de retorna T ou outra árvore que também seja solução. Essa justificativa se deve ao fato que T possui ramificações de c até aos vértices de W , e cada ramificação é uma caminho que satisfaz a condição de não passar por nenhum outro vértice em W e satisfaz o comprimento β . Então, a linha 11 produzirá verdade no teste lógico para essa ramificação



ou para qualquer outra de igual ou menor valor. Portanto, o algoritmo de Dijkstra será capaz de alcançar todos os vértices de W , produzindo assim os caminhos que formam as ramificações de T ou outra árvore que satisfaz todas as condições do problema com α é ilimitado.

Agora, em relação ao tempo temos que da linha 1 a 3 são necessários n passos. Na linha 4, a inserção dos vértices no Min-Heap também acarreta a construção do mesmo, que é feito no tempo $O(n)$. A extração dos elementos de um Min-Heap na linha 6 tem o custo $O(\log n)$ para a organização dos elementos do Heap e é executado n vezes por causa do tamanho inicial de H que é igual ao tamanho de $V(G)$. Então podemos concluir que o tempo de execução desta linha é $O(n \log n)$. Já que retificação dos elementos no Heap feita na linha 12 também tem o custo $O(\log n)$ e se repete para toda vizinhança de cada vértice apenas uma vez, podemos concluir que esta linha tem custo $O(m \log n)$. Por último, vem a construção da árvore que possui no pior caso $n - 1$ arestas, onde cada elemento de W é verificado sem repetição de arestas. Assim, o tempo totalizado é de $O(m \log n + n \log n + 3n - 1)$. \square

Outro fato importante é que o problema também permanece polinomial com os parâmetros α e β sejam fixados por uma constante, caso o grafo G de entrada possua arestas cujos pesos sejam inteiros positivos. Uma técnica simples para tal feito é construir de forma exaustiva uma árvore de conexão por 2 parâmetros usando *Backtracking*. Na Figura 4, temos uma árvore de busca usando o *Backtracking*, onde a altura da árvore é limitada por β e cada nó diferente da raiz localizado no determinado nível possui no máximo $\alpha - 1$ arestas ao próximo nível, já que possui 1 incidente do nível anterior. No caso da raiz c , por não possuir nenhuma aresta incidente do nível anterior, tem no máximo α arestas saindo para o próximo nível. Como um grafo G nem sempre é regular, podem ter graus distintos para os seus vértices, então podemos garantir que o grau de todos os seus vértices não é maior que o grau máximo Δ do grafo. Além disso, podemos assumir sem perda de generalidade que $\alpha \leq \Delta$. Portanto, como devemos escolher α vértices do nível 2 adjacentes a raiz e $\alpha - 1$ vértices no nível $i + 1$ adjacentes a vértices no nível i , para i diferente do nível da raiz, temos que as possibilidades para a raiz é C_{Δ}^{α} e $C_{\Delta-1}^{\alpha-1}$ para os vértices restantes, conforme demonstrado na Figura 4.

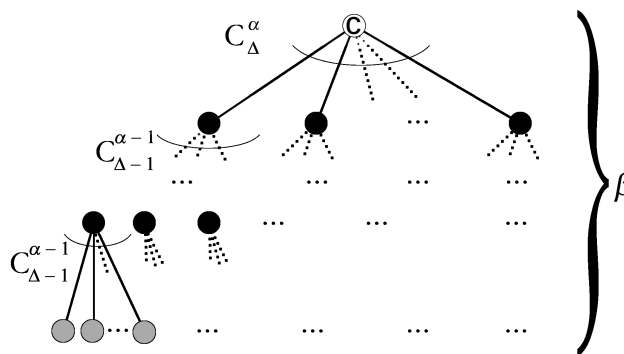


Figura 4: Árvore de busca baseado no *Backtracking*.

Assim, o *Backtracking* aplicado no grafo G com arestas cujos pesos são inteiros positivos, teria que no pior caso verificar todas as possíveis árvores candidatas a solução. O custo para isso é dado pelo princípio multiplicativo das arestas escolhidas em cada nível. Se garantirmos que os pesos das arestas são inteiros positivos, então temos que a árvore de busca exaustiva possui no máximo β níveis. Assim, temos o seguinte resultado:

$$C_{\Delta}^{\alpha} \times \underbrace{C_{\Delta-1}^{\alpha-1} \times \dots \times C_{\Delta-1}^{\alpha-1}}_{\beta-1} \quad (1)$$



Como $C_{\Delta}^{\alpha} = \frac{\Delta}{\alpha} \times C_{\Delta-1}^{\alpha-1}$, aplicando em (1) temos:

$$\frac{\Delta}{\alpha} \times C_{\Delta-1}^{\alpha-1} \times \underbrace{C_{\Delta-1}^{\alpha-1} \times \dots \times C_{\Delta-1}^{\alpha-1}}_{\beta-1} = \frac{\Delta}{\alpha} \times (C_{\Delta-1}^{\alpha-1})^{\beta} \quad (2)$$

Podemos notar que $C_{\Delta-1}^{\alpha-1}$ é $O((\Delta - 1)^{\alpha-1})$. Além disso, sabemos que $\Delta \leq n - 1$. Portanto, podemos concluir que a Equação (2) é $\frac{\Delta}{\alpha} O((n - 2)^{\alpha-1})^{\beta} = O(\frac{\Delta}{\alpha} (n - 2)^{\alpha\beta - \beta})$. Assim, concluímos o seguinte resultado:

Teorema 5 *Dado como entrada um grafo de arestas cujos pesos são inteiros positivos, o problema 2PCT pode ser verificado polinomialmente na ordem $O(\frac{\Delta}{\alpha} (n - 2)^{\alpha\beta - \beta})$, caso α e β sejam fixos.*

Agora mostraremos que o problema 2PST tem o mesmo grau de dificuldade ou até pior que o problema 2PCT.

Corolário 6 *O problema 2PST é NP-completo para qualquer que seja $\alpha \geq 3$ e β fixos.*

Prova: O resultado segue da construção do grafo G' conforme o **Teorema 2**, onde atribuímos a λ o valor $2n - 1$ ($n - 1$ arestas do caminho P e n arestas dos vértices extras x_u). Com isso, mostramos para $\alpha = 3$ e $\beta = n$. Agora, com os mesmos argumentos mostrados no **Corolário 3**, onde, para um grafo de arestas nulas, podemos atribuir $\lambda = 0$. Assim, concluímos a prova. \square

Além disso, diferentemente do caso 2PCT, o problema 2PST, permanece NP-completo mesmo com α ilimitado.

Teorema 7 *O problema 2PST é NP-completo com α ilimitado.*

Prova: O problema está em NP, pois basta usar os mesmos argumentos da **Proposição 1**, onde se inclui apenas a verificação do custo total da árvore. Para redução, faremos uso para redução do problema da Cobertura Exata por 3-conjuntos (X3C) definido como: *dado conjunto finito X com $|X| = 3q$ e uma coleção \mathcal{C} de subconjuntos de X com 3 elementos cada, isto é, $\mathcal{C} = \{C_1, \dots, C_r\}$, $C_i \subseteq X$, $|C_i| = 3 \forall 1 \leq i \leq r$, pergunta-se, a coleção \mathcal{C} contém uma cobertura exata para X , isto é, uma subcoleção $\mathcal{C}' \subseteq \mathcal{C}$ tal que todo elemento de X ocorre em exatamente um membro de \mathcal{C}' , isto é, $|\mathcal{C}'| = q$?*

O problema X3C é NP-completo [Garey e Johnson, 1979]. Então considere $\langle X, \mathcal{C} \rangle$ uma entrada genérica do X3C. Usaremos a função transformação $f : \langle X, \mathcal{C} \rangle \rightarrow \langle G, W, c, \alpha, \beta, \lambda \rangle$ descrita a seguir:

- i.* Conjunto de Vértices: $V(G) = \{x_1, x_2, \dots, x_{3q}\} \cup \{c_1, \dots, c_r, c\}$;
- ii.* Conjunto de Arestas: $E(G) = \{cc_j : \text{para todo } C_j \in \mathcal{C}\} \cup \{x_i c_j \mid x_i \in C_j\}$;
- iii.* Atribua: $w(e) = 1, \forall e \in E(G)$;
- iv.* Conjunto de Terminais: $W = \{x_1, x_2, \dots, x_{3q}\}$;
- v.* O vértice c definido em $V(G)$;
- vi.* $\alpha = \max\{q, 4\}$;
- vii.* $\beta = 2$;

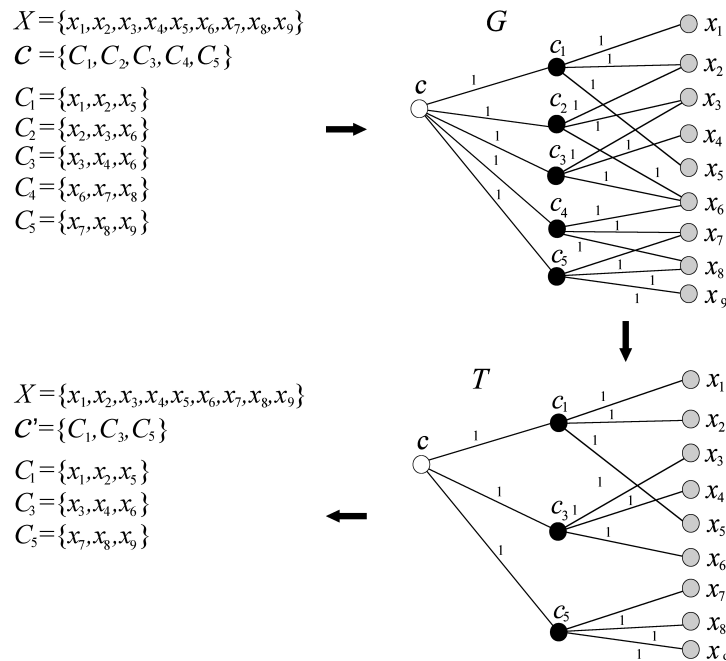


Figura 5: Ideia da construção do grafo G , onde os vértices de W estão em cinza, o vértice c está em branco. A solução da cobertura em X obtida de uma árvore de conexão T existente em G .

viii. $\lambda = 4q$.

Na Figura 5, temos uma ideia da construção do grafo G e a obtenção da solução de uma cobertura exata \mathcal{C}' a partir de uma árvore de conexão T encontrada em G .

Agora, provaremos que X3C diz *sim* para uma instância $\langle X, \mathcal{C} \rangle$ se, e somente se, também for *sim* para 2PST com a entrada $\langle G, W, c, \alpha, \beta, \lambda \rangle$.

(X3C \Rightarrow 2PST)

Se para $\langle X, \mathcal{C} \rangle$ existe uma cobertura $\mathcal{C}' \subseteq \mathcal{C}$ dos elementos de X de tamanho $3q$. Então, para cada $C_j \in \mathcal{C}'$, podemos escolher q vértices c_j de $V(G)$, onde cada um desses vértices é adjacente c , e consumindo assim q arestas. Agora, para cada $\{x_a, x_b, x_c\} = C_j$, podemos escolher os vértices x_a, x_b e x_c adjacente ao vértice c_j no grafo G , consumindo 3 arestas para cada C_j , totalizando $3q$ arestas. Por fim, temos uma árvore T conectando $W \cup \{c\}$ com $4q$ arestas ao adotarmos o processo descrito anteriormente, pois produz um grafo conexo e acíclico, já que que nenhum x_i pertence a mais de um elemento em \mathcal{C}' . Podemos notar também que T possui $\alpha = \max\{q, 4\}$ e $\beta = 2$. Portanto, temos um certificado de *sim* para 2PST.

(X3C \Leftarrow 2PST)

Agora, considere que temos uma árvore T em G que conecta todos os vértices de $W \cup \{c\}$ com no máximo $4q$ arestas, $\alpha = \max\{q, 4\}$ e $\beta = 2$. Note que não há como conectar $W \cup \{c\}$ com $4q - 1$ arestas, pois cada c_j só pode ser adjacente a três vértices que representam os elementos de X e, portanto, são necessários $3q$ arestas. Então, esses $3q$ vértices só podem ser incidentes a no mínimo q vértices c_j 's, onde esses obrigatoriamente teriam que ser adjacentes a c , pois os vértices c_j 's não são adjacentes entre si. Então não há como conectar $W \cup \{c\}$ com menos de que $4q$ arestas. Assim, temos que T possui exatamente $4q$ arestas. Então, se removermos as q arestas incidentes aos vértices c_j 's de T que partem de c , temos que nos restam $3q$ arestas incidentes a $3q$ vértices, que por sua vez representam os elementos de X .



Portanto, se compormos $\mathcal{C}' = \bigcup_{c_j \in V(T)} \{C_j\}$, temos que \mathcal{C}' é uma cobertura de X e com isso temos um certificado `sim` para X3C.

Assim, o problema 2PST é NP-completo. \square

O resultado direto do teorema anterior é mostrado a seguir:

Corolário 8 *O problema 2PST é NP-completo com α ilimitado e $\beta \geq 2$ fixo.*

Prova: Segue da mesma redução do **Teorema 7**. Note que β sempre será satisfeito pela construção, concluindo assim a prova. \square

Além disso, temos o seguinte resultado:

Corolário 9 *Dado como entrada um grafo de arestas cujos pesos são inteiros positivos, o problema 2PST pode ser verificado polinomialmente na ordem $O(\frac{\Delta}{\alpha}(n-2)^{\alpha\beta-\beta})$, caso α e β sejam fixos.*

Prova: Segue do **Teorema 5**, pois gerando todas as arvores podemos escolher aquela de menor custo na ordem $O(\frac{\Delta}{\alpha}(n-2)^{\alpha\beta-\beta})$. \square

Agora, considere α_R o número máximo de dispositivos ZR's e α_D o número máximo de dispositivos ZDF's que podem ser conectados a cada dispositivo ZC ou ZR. Note que $\alpha = \alpha_R + \alpha_D$. Então podemos deduzir alguns resultados:

Corolário 10 *O problema 2PST é NP-completo para qualquer que seja $\alpha_R \geq 2$, $\alpha_D \geq 1$ e β fixos.*

Prova: Segue do **Corolário 6**. \square

Corolário 11 *O problema 2PST é NP-completo para qualquer que seja $\alpha_D \geq 3$ e $\beta \geq 2$ fixos, e α_R ilimitado.*

Prova: Segue do **Teorema 7**. \square

Corolário 12 *Dado como entrada um grafo de arestas cujos pesos são inteiros positivos, o problema 2PST pode ser verificado polinomialmente, caso α_R , α_D e β sejam fixos.*

Prova: Segue do mesmo raciocínio usado no **Teorema 5**. \square

3. Conclusão

Portanto, com as definições e os resultados obtidos neste trabalho, acreditamos em ter dado o primeiro passo na análise de conexões ótimas no que se referem aos elementos que compõe o padrão ZigBee, que por sua vez são aplicados na Internet das Coisas. Com isso, esperamos ter motivado bem os problemas para fomentar novos trabalhos sobre o tema.

Embora os autores em [Lin e Liu, 2010] abordarem roteamento de árvore de corte mínimo (*Shortcut Tree Routing in ZigBee Networks*) para melhorar a conexão ponto a ponto na árvore, o nosso trabalho é de fato o primeiro em focar na conexão ótima entre todos os elementos ZigBee.

Mesmo com o resultado de NP-completude para 2PST, acreditamos que este problema é polinomial quando λ é fixado por uma constante, algo que pretendemos demonstrar em trabalhos futuros.



Referências

- Atzori, L., Iera, A., e Morabito, G. (2010). The internet of things: A survey. *Comput. Netw.*, 54: 2787–2805.
- Brazil, M., Graham, R. L., Thomas, D. A., e Zachariasen, M. (2014). On the history of the euclidean steiner tree problem. *Archive for History of Exact Sciences*, 68(3):327–354.
- Courant, R. e Robbins, H. (1943). *What is mathematics?: An elementary approach to ideas and methods*. Oxford University Press.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271.
- Dourado, M. C., Oliveira, R., Souza, U. S., e Protti, F. (2014). Design of connection networks with bounded number of non-terminal vertices. *Matematica Contemporanea*, 42:39–48.
- Dreyfus, S. E. e Wagner, R. A. (1971). The steiner problem in graphs. *Networks*, 1(3):195–207.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- Hakimi, S. L. (1971). Steiner's problem in graphs and its implications. *Networks*, 1(2):113–133.
- Kahng, A. B. (1991). A steiner tree construction for vlsi routing. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume i, p. 133–139 vol.1.
- Lin, C.-M. e Liu, T.-J. (2010). Data communications inside vehicular environments. In Huang, C.-M. e Chen, Y.-S., editors, *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications*. IGI Global.
- Weng, J., Mareels, I., e Thomas, D. (2012). Probability steiner trees and maximum parsimony in phylogenetic analysis. *Journal of Mathematical Biology*, 64(7):1225–1251.