

Posicionamento do Número Mínimo de Recursos que Maximizam Caminhos Vértice-Disjuntos em uma Rede de Topologia Arbitrária

Henrique Hepp¹, Jaime Cohen², Elias P. Duarte Jr.¹

¹Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19018 – CEP 81531-980 – Curitiba – PR – Brazil.

²Departamento de Informática – Universidade Estadual de Ponta Grossa (UEPG)
Av. General Carlos Cavalcanti 4748 – Bloco L - Sala 103 CEP 84030-900 – PR – Brazil.

hhepp@inf.ufrpr.br, jaimecohen@gmail.com, elias@inf.ufrpr.br

Abstract. *Resource placement is a problem that has several variants in computer networks, from server placement in the traditional client-server architecture to the allocation of controllers in SDN networks, or caches in CDN networks, among many others. This work presents the problem of placing the minimum number of resources in order to maximize the number of vertex-disjoint paths between a resource and its clients. One of the contributions of the paper is the proof that the problem of finding the minimum number of resources under these conditions is NP-complete. An exact solution to this problem was implemented and experiments showed its feasibility when it was run in several arbitrary topology networks. We present results comparing the connectivity based resource location problem with the classic problem in which the sum of the distances between clients and their resources is minimized. Experimental results show both the connectivity gains and the impact on the sum of the distances when the proposed solution is applied.*

Resumo. *O posicionamento de recursos em redes é um problema que encontra diversas variantes, desde o posicionamento de servidores na arquitetura tradicional cliente-servidor, passando pelo posicionamento de controladores em redes SDN, ou caches em redes CDN, entre vários outros. Este trabalho apresenta o problema de posicionar o número mínimo de recursos de modo a maximizar o número de caminhos vértice-disjuntos entre um recurso e seus clientes. Uma das contribuições do trabalho é a prova de que o problema de encontrar o número mínimo de recursos sob essas condições é NP-completo. Uma solução exata para o problema foi implementada e sua execução se mostrou viável em diversas redes de topologia arbitrária. Apresentamos os resultados comparando-os com o problema clássico em que é minimizada a soma das distâncias entre os clientes e seus recursos. Resultados experimentais avaliam o ganho de conectividade e o aumento da soma das distâncias quando a solução proposta é aplicada.*

1. Introdução

O problema de posicionar recursos adquire as mais diversas formas nas redes de computadores [Rochman et al. 2013]. Na sua versão na tradicional arquitetura cliente-

servidor, trata do posicionamento de servidores com o objetivo de minimizar a latência para os clientes [Rodolakis et al. 2006] ou realizar o balanceamento de carga entre múltiplos servidores ou ainda aumentar a chance dos clientes acessarem servidores corretos após a ocorrência de falhas. Outra versão relacionada é o posicionamento de proxies, entidades intermediárias entre clientes e servidores, por exemplo na Web [Li et al. 1999]. Nas redes CDN (*Content-Delivery Network*) o problema é semelhante, mas consiste em posicionar réplicas de conteúdos específicos dado um padrão de tráfego [Qiu et al. 2001] e pode levar em conta fatores como a popularidade do conteúdo. Nas redes P2P estruturadas o problema se refere ao posicionamento de super-peers que tornam a rede mais efetiva, de acordo com critérios diversos [Rao et al. 2010]. Nas redes de sensores, o problema pode tomar ainda diversas formas, como a localização de sensores em um campo para maximizar a cobertura com o menor número de sensores [Younis and Akkaya 2008]. O problema assume, em geral, uma rede de topologia arbitrária, mas há variações para topologias específicas [Bae 1997]. Recentemente, este problema ganhou ainda uma nova aplicação nas redes SDN, para o posicionamento de controladores [Heller et al. 2012, Muller et al. 2014].

Neste trabalho definimos uma nova versão do problema, que visa tirar partido da máxima conectividade da rede. Para facilitar a exposição, usaremos neste artigo a arquitetura cliente-servidor, mas o problema pode ser formulado de forma equivalente em outros contextos. O problema consiste em posicionar o número mínimo de servidores de modo que cada cliente esteja conectado a um servidor com o número máximo de caminhos vértice-disjuntos que esse cliente pode formar com qualquer outro nodo da rede. Desse modo, garante-se a maior disponibilidade possível de caminhos alternativos para a conexão entre um cliente e seu servidor, tendo em vista a ocorrência de falhas de conexão por falhas de nodos entre um cliente e seu servidor.

Uma das contribuições do trabalho é a prova de que o problema de encontrar o número mínimo de recursos sob essas condições é NP-completo. Uma solução exata para o problema foi implementada usando programação linear inteira. Ela foi testada em grafos arbitrários conexos com número de vértices entre 60 e 200. Os resultados foram comparados com o caso em que é minimizada a soma das distâncias entre os clientes e seus servidores, problema conhecido como p -mediana. Com relação a esse caso, observou-se um aumento médio de 20% para a soma das distâncias. Quanto à conectividade, o problema da p -mediana apresentou clientes com uma defasagem média de 36% de caminhos vértice-disjuntos com relação à conectividade máxima.

O restante do artigo está organizado da seguinte maneira. A seção 2 apresenta as definições preliminares. A seção 3 descreve trabalhos relacionados com o problema proposto. A seção 4 apresenta a definição formal do problema e a demonstração de que é NP-completo. A seção 5 descreve as especificações dos problemas usados nos experimentos como problemas de programação linear inteira. A seção 6 apresenta os resultados experimentais obtidos. Por fim, a seção 7 conclui o trabalho.

2. Definições Preliminares

Nesta seção são dadas definições preliminares que serão usadas nas próximas seções.

2.1. Grafos e Conectividade

Um *grafo*, ou *multigrafo*, $G = (V, E)$ é um par no qual V é um conjunto finito de elementos chamados de *vértices* e E é um multiconjunto de elementos chamados de *arestas* pertencentes ao conjunto dos subconjuntos de V de tamanho 2. Dada uma aresta $e = \{u, v\} \in E$, dizemos que e é *incidente* em u e v .

Um *caminho* em um grafo G entre dois vértices v_1 e v_k , ou v_1 - v_k -caminho, é uma sequência de vértices distintos e arestas $P = (v_1, e_1, v_2, e_2, \dots, e_{k-1}, v_k)$ tal que $v_1, v_2, \dots, v_k \in V$, $e_1, e_2, \dots, e_{k-1} \in E$ e $e_i = \{v_i, v_{i+1}\}$, para $i = 1, 2, \dots, k - 1$. Um grafo G é dito *conexo* se existe um caminho entre todo par de vértices de G .

Dados dois vértices $s, t \in V$, um conjunto de s - t -caminhos é *vértice-disjunto* se cada par de caminhos do conjunto tem como intersecção apenas os vértices s e t . A vértice-conectividade local entre s e t , denotada por $\kappa(s, t)$, é o número de caminhos vértice-disjuntos entre s e t .

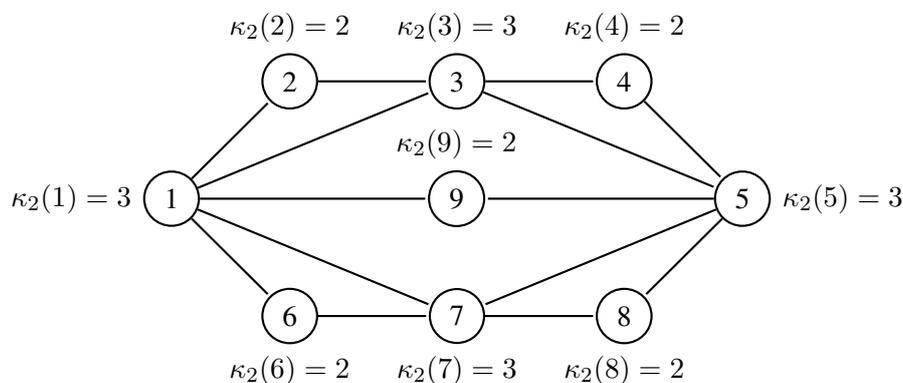


Figura 1. Grafo com os valores da 2-vértice-conectividade [Pires et al. 2011].

2.2. Medidas de Conectividade Baseadas em Cortes de Vértices

A medida de conectividade baseada em corte de vértices, vértice-conectividade [Pires et al. 2011] é definida a seguir:

Definição 1. A *vértice-conectividade* de um conjunto $X \subseteq V$ é a menor *vértice-conectividade local* entre quaisquer pares de vértices em X e é denotada por $\kappa(X)$.

Definição 2. Seja G um grafo e $i \in \mathbb{N}$, $2 \leq i \leq |V|$; a i -*vértice-conectividade* de um vértice v , denotada por $\kappa_i(v)$, é a maior *vértice-conectividade* de um conjunto $X \subseteq V$ satisfazendo $v \in X$ e $|X| \geq i$.

Em particular, a medida 2-vértice-conectividade $\kappa_2(v)$ de um grafo G é o maior número de caminhos vértice-disjuntos entre v e qualquer outro vértice de G . A figura 1 mostra um exemplo. O vértice 1 tem 3 caminhos vértice-disjuntos com o vértice 5, como o vértice 1 não tem um número maior de caminhos vértice-disjuntos com qualquer outro vértice, $\kappa_2(1) = 3$.

3. Trabalhos Relacionados

Problemas de localização (*location problems*) são problemas que procuram encontrar as melhores localizações para instalações (*facilities*) de modo a otimizar alguma propriedade, como, por exemplo, a soma das distâncias entre as instalações e seus clientes, que é o problema da p -mediana.

Para compreendermos o problema da p -mediana de forma intuitiva, consideremos o seguinte exemplo. Em um sistema cliente-servidor, devem ser posicionadas réplicas do servidor em até p vértices de um grafo com n vértices, os demais vértices correspondem a clientes. O objetivo é fazer o posicionamento das réplicas de forma a minimizar a soma das distâncias de cada cliente à réplica mais próxima [Sahoo et al. 2017, Reese 2006].

Outro problema de localização é o da localização de origens, fontes de dados (*source location*). Nesse caso são consideradas as conectividades e , ou, as quantidades de fluxo entre as instalações e os clientes. Uma formulação geral para esse problema é: dado um grafo capacitado $G(V, E, c)$, uma função de custo $w : V \rightarrow R_+$ (R_+ denota o conjunto de reais não negativos) e uma função de demanda $d : V \rightarrow R_+$, queremos minimizar $\sum_{x \in S} w(x)$, sob às restrições $S \subseteq V$ e $\psi(S, x) \geq d(x) \forall x \in V - S$. A função $\psi(S, x)$ pode ser a aresta-conectividade entre S e x , isto é, $\lambda(S, x)$, a vértice-conectividade $\kappa(S, x)$ ou uma função de fluxo entre o conjunto S e o vértice x [Ito et al. 2002].

Diversas variantes desse problema já foram estudadas [Makino 2012]. Para este artigo citamos apenas o seguinte caso: dado um grafo $G = (V, E)$ não capacitado e não direcionado e uma constante inteira k , queremos minimizar $|S|$, sob às restrições $S \subseteq V$ e $\kappa(S, x) \geq k, \forall x \in V$.

Foi provado em [Ito et al. 2002] que esse problema é NP-completo para $k \geq 3$. Não encontramos um resultado publicado para o caso particular em que a restrição seja a vértice-conectividade entre um vértice s e um vértice x , ($\kappa(s, x) \geq k, \forall x \in V$). No entanto, é possível demonstrar usando a mesma prova em [Ito et al. 2002] que esse caso em particular também é NP-completo para $k \geq 3$.

4. Alocação de Recursos com a Máxima Conectividade

Neste trabalho propõe-se estudar o caso em que cada cliente está conectado a um servidor com o número máximo de caminhos vértice-disjuntos que o cliente é capaz de estabelecer com qualquer outro nodo da rede. Denominamos esse problema como o *problema de alocação de recursos com a máxima conectividade*.

Tendo sido definido o maior número possível de caminhos vértice-disjuntos para um vértice v como sendo a medida 2-vértice-conectividade ou $\kappa_2(v)$, podemos definir esse problema como: dado um grafo não direcionado $G = (V, E)$, queremos encontrar o conjunto mínimo de servidores $S \subseteq V$, de modo que para todo $v \in V$, existe $s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$.

Estendemos a definição de $\kappa(u, v)$ de forma que $\kappa(v, v) = \kappa_2(v)$. Essa convenção garante que um servidor poderá ser servidor dele mesmo. Podemos também formular o problema como:

Entrada: $G = (V, E)$

Saída: $S \subseteq V$

Função Objetivo: $\min |S|$

Restrições: $\forall v \in V, \exists s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$

Reformulamos o problema de otimização para o de decisão e provamos que este é NP-completo. No problema de decisão queremos saber se existe uma solução de máxima conectividade de tamanho menor ou igual a uma constante inteira p .

Entrada: $G = (V, E)$ e $p \in \mathbb{N}$

Saída: sim, se $\exists S \subset V$ tal que $|S| \leq p$ e $\forall v \in V, \exists s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$
não, caso contrário

Teorema 3. *O problema de decisão da máxima conectividade é NP-completo.*

Para provar que o problema de decisão da máxima conectividade é NP-completo precisa-se provar primeiro que ele pertence a NP e depois que ele é NP-difícil. A primeira parte é provada em seguida no lema 1 e a segunda no lema 2.

Lema 1. *O problema de decisão de máxima conectividade pertence a NP.*

Demonstração. O problema de decisão de máxima conectividade pertence a NP, pois dado um conjunto de vértices S pode-se verificar em tempo polinomial se o mesmo é solução para o problema da seguinte maneira. O algoritmo de verificação afirma que $|S| \leq p$ e verifica para cada $s \in S$ com quais $v \in V$ é satisfeita a expressão $\kappa(s, v) = \kappa_2(v)$. No final é calculada a união de todos os vértices v que satisfizeram essa igualdade. Se essa união for igual ao conjunto V então é constatado que $\forall v \in V, \exists s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$. Como os cálculos de $\kappa(s, v)$ e $\kappa_2(v)$ são polinomiais, essa verificação pode ser feita em tempo polinomial. \square

Lema 2. *O problema de decisão da máxima conectividade é NP-difícil.*

Esse lema pode ser provado mostrando que existe uma redução de tempo polinomial de um problema NP-completo para o problema de conectividade máxima. Nesse artigo é feita a redução do problema de decisão da cobertura de vértices para o problema de decisão de máxima conectividade. Primeiramente é mostrada como é feita a redução entre os problemas e depois mostramos sua validade provando que o grafo tem uma cobertura por vértices de tamanho p se, e somente se, o grafo reduzido tem um conjunto de servidores de máxima conectividade de tamanho p . Isso é demonstrado pelos lemas 3 e 4. O lema 3 demonstra que uma solução para o problema de cobertura de vértices também é uma solução do problema de máxima conectividade. O lema 4 demonstra que uma solução do problema de máxima conectividade também é uma solução do problema de cobertura por vértices.

O problema de decisão da cobertura por vértices pode ser definido como: dado um grafo $G_0 = (V_0, E_0)$ e uma constante p , é perguntado se existe um subconjunto $V_c \subseteq V_0$ de tamanho p tal que se $\{x, y\} \in E_0$ então $x \in V_c$ ou $y \in V_c$ (ou ambos), ou seja, os vértices de V_c “cobrem” todas as arestas do grafo [Cormen et al. 2009].

Entrada: $G_0 = (V_0, E_0)$ e uma constante p

Saída: sim, se $\exists V_c$, com $|V_c| = p$ e $V_c \subseteq V_0$ e $\{x, y\} \cap V_c \neq \emptyset, \forall \{x, y\} \in E_0$
não, caso contrário

A redução da cobertura de vértices para a máxima conectividade pode ser feita pelo seguinte algoritmo:

Dado $G_0 = (V_0, E_0)$ construa um grafo $G = (V, E)$ de modo que:

$$V = V_0 \cup V' \quad (1)$$

$$V' = \bigcup_{\{x_i, x_j\} \in E_0} \{a_{i,j}, b_{i,j}\} \quad (2)$$

$$E = E' \cup E'' \quad (3)$$

$$E' = \bigcup_{\{x_i, x_j\} \in E_0} \{\{x_i, x_j\}, \{x_i, a_{i,j}\}, \{x_i, b_{i,j}\}, \{x_j, a_{i,j}\}, \{x_j, b_{i,j}\}, \{a_{i,j}, b_{i,j}\}\} \quad (4)$$

A redução até aqui para cada subgrafo composto pela aresta $\{x_i, x_j\}$ pode ser observada na figura 2.

As arestas de E'' são definidas pelo seguinte algoritmo:

1. Encontre o maior $\kappa_2(v)$, $v \in V_0$, com relação ao grafo $G' = (V, E')$
2. Para toda aresta $\{x_i, x_j\} \in E_0$ calcule $\kappa(x_i, x_j)$ com relação ao grafo $G' = (V, E')$
3. Para toda aresta $\{x_i, x_j\} \in E_0$ adicione a E'' o número de arestas entre os vértices x_i e x_j necessários para que $\kappa(x_i, x_j)$ seja igual ao maior $\kappa_2(v)$ encontrado no passo 1, ou seja, adicione $\max \kappa_2(v) - \kappa(x_i, x_j)$ arestas entre x_i e x_j .

Podemos ver um exemplo de uma redução na figura 3. Primeiro, a partir do grafo $G_0 = (V_0, E_0)$ todas as arestas $\{x_i, x_j\}$ são convertidas de acordo com as equações 1 a 4 para formar o grafo $G' = (V, E')$. Depois, calcula-se o maior $\kappa_2(v)$ para os vértices $\{x_1, x_2, x_3, x_4, x_5\}$ referente ao grafo $G' = (V, E')$. O maior $\kappa_2(v)$ é igual a 4 para os vértices $\{x_2, x_3, x_4\}$ e o valor de $\kappa(x_i, x_j)$ entre esses vértices também é igual a 4. Já para os vértices x_1 e x_5 o valor de $\kappa_2(v)$ é igual a 3 e os valores de $\kappa(x_1, x_2)$ e $\kappa(x_4, x_5)$ também são iguais a 3. Conclui-se a redução adicionando uma aresta entre x_1 e x_2 e outra entre x_4 e x_5 . Assim para toda aresta $\{x_i, x_j\} \in E_0$ os valores de $\kappa(x_i, x_j)$ no grafo $G(V, E)$ são iguais.

Lema 3. *Se V_c é uma solução do problema de cobertura de vértices para o grafo G_0 , então V_c também é uma solução do problema de máxima conectividade para o grafo G .*

Demonstração. Pela definição do problema de cobertura de vértices, a solução V_c “cobre” todas as arestas $\{x_i, x_j\} \in E_0$, e para tal V_c contém x_i ou x_j , ou ambos. Essa mesma solução no grafo $G(V, E)$ é uma solução para o problema de máxima conectividade, pois todos os servidores do conjunto V_c atendem os demais vértices de acordo com a condição $\forall v \in V, \exists s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$ como mostrado a seguir:

1. Tanto x_i como x_j podem ser servidores dos vértices $\{a_{i,j}, b_{i,j}\}$ obtidos na redução da aresta $\{x_i, x_j\}$, pois, como observado na figura 2, os vértices $a_{i,j}$ e $b_{i,j}$ têm $\kappa_2(v) = 3$ e $\kappa(x_i, v) = \kappa(x_j, v) = 3$ para qualquer $v \in \{a_{i,j}, b_{i,j}\}$.
2. x_i pode ser servidor de x_j ou vice-versa. O último passo da redução, a adição das arestas E'' , garante que $\kappa_2(x_i) = \kappa_2(x_j) = \kappa(x_i, x_j)$.

□

Lema 4. *Se S é uma solução do problema de máxima conectividade para o grafo G , então S também é uma solução do problema de cobertura de vértices para o grafo G_0 .*

Demonstração. Para a solução S do problema de máxima conectividade para o grafo G também ser solução para o problema de cobertura de vértices em G_0 , o conjunto S deverá conter x_i ou x_j ou ambos para toda aresta $\{x_i, x_j\} \in E_0$.

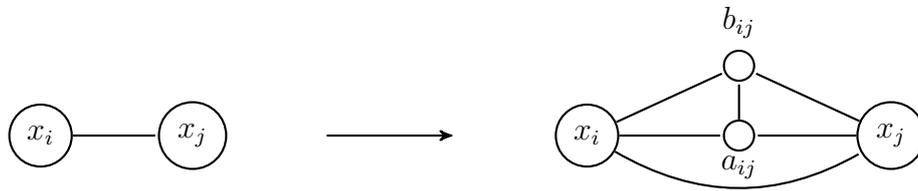


Figura 2. Redução de uma aresta (x_i, x_j) .

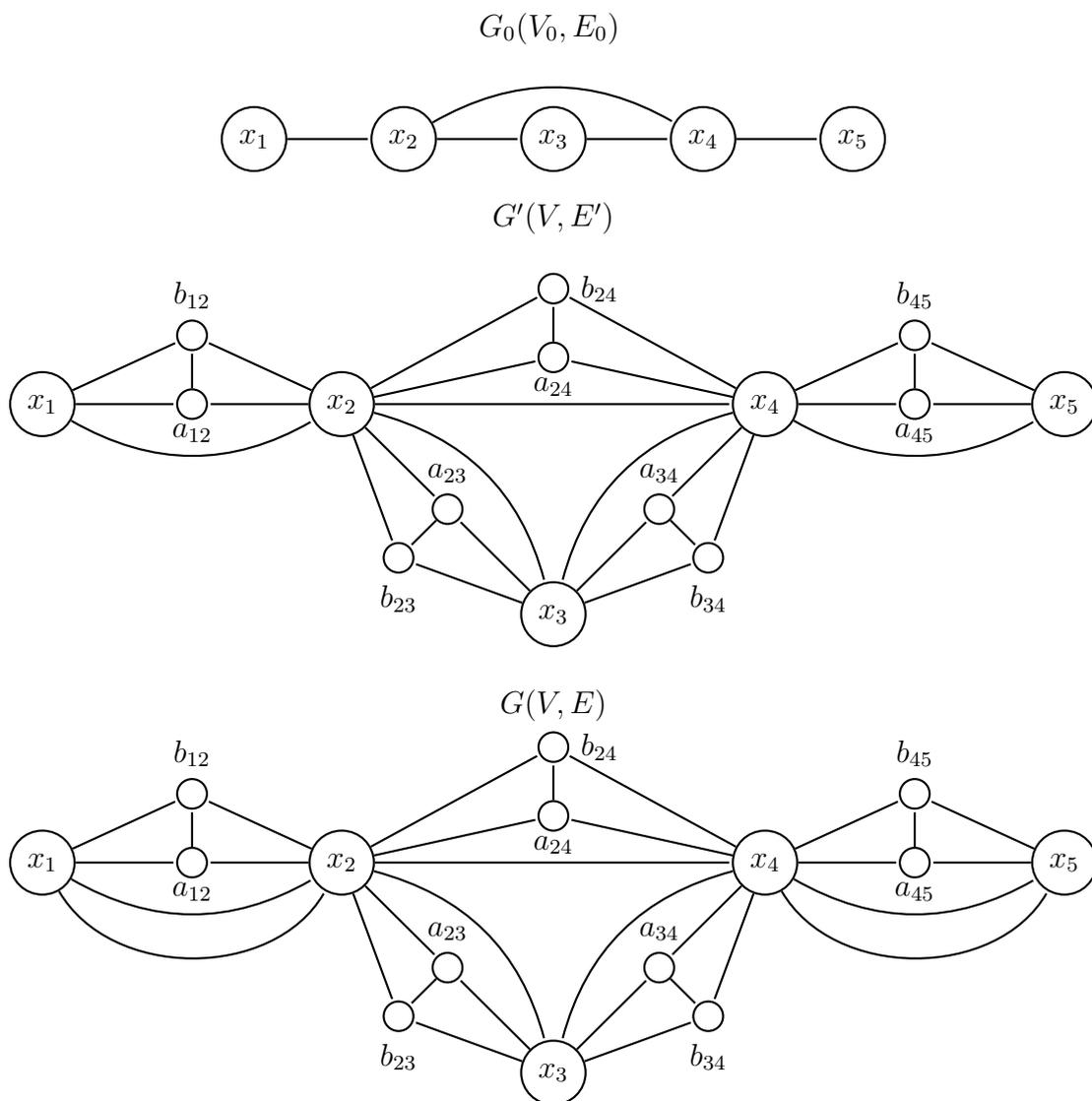


Figura 3. Exemplo de uma redução.

Pela redução, uma aresta $\{x_i, x_j\} \in E_0$ corresponde ao conjunto de vértices $\{a_{i,j}, b_{i,j}\}$ e ao de arestas $\{\{x_i, a_{i,j}\}, \{x_i, b_{i,j}\}, \{x_j, a_{i,j}\}, \{x_j, b_{i,j}\}, \{a_{i,j}, b_{i,j}\}\}$, como ilustrado na figura 2.

Vamos mostrar que caso a solução não contenha nem x_i e nem x_j , então podemos transformar essa solução para uma solução de mesmo tamanho que contém x_i ou x_j .

Caso nem x_i , nem x_j estejam na solução, então para atender os vértices $\{a_{i,j}, b_{i,j}\}$, será necessário escolher um servidor do conjunto $\{a_{i,j}, b_{i,j}\}$, pois todos os demais vértices além de x_i e x_j terão apenas dois caminhos vértice-disjuntos com eles. No entanto, esse servidor pode ser substituído por qualquer um dos vértices x_i ou x_j , pois $\kappa_2(v) = 3$ e $\kappa(x_i, v) = \kappa(x_j, v) = 3$, sendo $v \in \{a_{i,j}, b_{i,j}\}$.

Desse modo, para toda aresta $\{x_i, x_j\} \in E_0$, o conjunto S contém ao menos um dos vértices x_i ou x_j . Portanto, S é uma cobertura por vértices de G_0 . \square

5. Formulações dos Problemas de Programação Linear Inteira

Podemos definir o problema proposto usando a formulação de programação linear inteira da seguinte forma. Considere um grafo $G = (V, E)$. Sejam dados $S \subset V$, o conjunto de locais (vértices) possíveis para posicionar servidores, e $C \subset V$ o conjunto de vértices com clientes, tais que $S \cup C = V$. Sejam dados os valores $\kappa_2(v)$, para todo $v \in V$, e $\kappa(j, i)$, o número máximo de caminhos vértice disjuntos entre o cliente em j e o servidor em i , para todos $j \in C$ e $i \in S$. Sejam as variáveis de decisão definidas como:

$$x_{ij} = \begin{cases} 1 & \text{se o cliente em } j \text{ está conectado ao servidor em } i \\ 0 & \text{caso contrário} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{se há servidor em } i \\ 0 & \text{caso contrário} \end{cases}$$

O problema consiste em encontrar o menor conjunto de servidores tal que todos os clientes estejam conectados a um desses servidores com $\kappa_2(v)$ caminhos vértice disjuntos, o que pode ser formulado como:

$$\text{Min} \sum_{i \in S} y_i \quad (5)$$

$$\kappa(j, i)x_{ij} = \kappa_2(j)x_{ij}, \forall i \in S, j \in C \quad (6)$$

As demais restrições do problema são:

$$y_i \geq x_{ij}, \forall i \in S, j \in C \quad (7) \quad \sum_{i \in S} x_{ij} = 1, \forall j \in C \quad (8)$$

$$y_i \in \{0, 1\} \quad (9) \quad x_{ij} \in \{0, 1\} \quad (10)$$

A restrição 7 assegura que há um servidor em i se o cliente em j está conectado ao servidor em i . A restrição 8 assegura que o cliente em j está conectado a apenas um servidor. As restrições 9 e 10 especificam o domínio das variáveis de decisão y_i e x_{ij} .

5.1. Seleção do Conjunto de Servidores Considerando Distâncias

Instâncias do problema proposto poderão ter mais de uma solução ótima. De acordo com o contexto, é importante obter uma solução que seja otimizada de acordo com um segundo critério. Por exemplo, pode-se considerar como segundo critério a soma das distâncias entre os servidores e os clientes. Ou seja, quer-se obter o conjunto de servidores de tamanho p , o número mínimo de servidores encontrado anteriormente, e o conjunto de clientes conectados a esses servidores de modo que a soma das menores distâncias entre os servidores e os clientes seja a menor possível e o número de caminhos vértices disjuntos entre clientes e servidores seja igual a $\kappa_2(v)$ para todo $v \in C$. Podemos formular esse problema usando a programação linear inteira da seguinte forma.

Seja a entrada as mesmas variáveis do problema anterior, mais a variável d_{ij} que é a menor distância de acordo com alguma métrica entre um servidor posicionado no vértice i e um cliente posicionado no vértice j ; e a variável p que é o número mínimo de servidores obtido pela execução do problema anterior.

O problema consiste em encontrar um conjunto de servidores de tamanho p , em que todos os clientes estejam conectados a um desses servidores, de modo que a distância total D entre os servidores e clientes seja a menor possível, o que pode ser formulado como:

$$\text{Min } D = \sum_{i \in S, j \in C} d_{ij} x_{ij} \quad (11) \quad \sum_{i \in S} y_i = p \quad (12)$$

$$\kappa(j, i) x_{ij} = \kappa_2(j) x_{ij}, \quad \forall i \in S, j \in C \quad (13)$$

As demais restrições do problema são as mesmas equações 7 a 10 apresentadas anteriormente.

Para verificar se compensa usar essa otimização, podemos comparar a soma mínima das distâncias com a maior soma de distâncias possíveis sob as mesmas restrições, o que pode ser formulado como:

$$\text{Max } D = \sum_{i \in S, j \in C} d_{ij} x_{ij} \quad (14) \quad \sum_{i \in S} y_i = p \quad (15)$$

$$\kappa(j, i) x_{ij} = \kappa_2(j) x_{ij}, \quad \forall i \in S, j \in C \quad (16)$$

As demais restrições do problema são novamente as equações 7 a 10.

5.2. Comparação com a p -Mediana

As somas das distâncias entre os clientes e seus servidores do problema de máxima conectividade ainda podem ser comparadas com a menor soma de distâncias relaxando a

restrição 6 de conectividade, que consiste no problema da p -mediana, podendo ser formulado como [Reese 2006]:

$$\text{Max } D = \sum_{i \in S, j \in C} d_{ij} x_{ij} \quad (17) \quad \sum_{i \in S} y_i = p \quad (18)$$

As demais restrições do problema são as equações 7 a 10.

Também é relevante avaliar o problema da p -mediana com relação à conectividade, em relação a quão próxima uma solução para o problema da p -mediana está para as condições de conectividade máxima em que o número de caminhos vértices disjuntos entre clientes e servidores é igual a $\kappa_2(v)$. Para tal, após encontrarmos a soma mínima D , podemos calcular:

$$\text{Min } \textit{somaDif} = \sum_{i \in S, j \in C} \kappa_2(j) x_{ij} - \kappa(j, i) x_{ij} \quad (19)$$

$$\sum_{i \in S, j \in C} d_{ij} x_{ij} = D \quad (20) \quad \sum_{i \in S} y_i = p \quad (21)$$

As demais restrições do problema são as equações 7 a 10.

O termo *somaDif* na equação 19 é a soma das diferenças de $\kappa(c) - \kappa(s, c)$. Esse problema encontra a menor soma de diferenças. Se essa soma for igual a zero, então essa solução do problema da p -mediana também é solução para o problema de máxima conectividade. Valores grandes de *somaDif* mostram que ao otimizar apenas a distância, perde-se conectividade entre clientes e servidores, ainda que considerada a solução com melhor conectividade dentre aquelas com a menor soma de distâncias.

6. Resultados Experimentais

Para a análise experimental, foi executado um algoritmo de programação linear inteira em cada uma das formulações descritas na seção 5. O pacote “GNU Linear Programming Kit (GLPK) Simplex Optimizer”, versão 4.61 foi utilizado. Os dados obtidos experimentalmente são apresentados da tabela 2 e a tabela 1 descreve as abreviaturas que nomeiam as colunas da tabela 2. Foram feitos experimentos com grafos com 60 até 200 vértices, conexos, e com mais de $|V| - 1$ arestas obtidos no site `topology-zoo.org` que são topologias de diversas redes. Uma outra característica desses grafos é que são esparsos. Na tabela 2 observa-se que o maior $\kappa_2(v)$ desses grafos é 6, valor que ocorre em vértices dos grafos Uninett e Cogent.

Para a análise, primeiramente, obteve-se o número mínimo de servidores $S \in V$, tal que para todo $v \in V$, existe $s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$. Esses valores se encontram na tabela 2 sob a coluna de título p . Observa-se que para esses grafos o número mínimo de servidores é na média 6% do número total de vértices.

Em seguida encontrou-se a solução de maior conectividade para a qual a soma das distâncias entres os clientes e seus servidores seja a maior possível. Essa soma está sob a

Tabela 1. Notação

$ V $	Número de vértices do grafo.
$ A $	Número de arestas do grafo.
$\max \kappa_2(v)$	Maior $\kappa_2(v)$ de um vértice do grafo.
p	Número mínimo de servidores para o problema de máxima conectividade.
$\frac{p}{ V }$	Média do número mínimo de servidores com relação ao número de vértices.
$\maxC \max (\sum \text{dist})$	Maior soma de distâncias possível para o problema de máxima conectividade.
$\maxC \min (\sum \text{dist})$	Menor soma de distâncias possível para o problema de máxima conectividade.
$p\text{-med} \sum \text{dist}$	Soma das distâncias para o problema da p -mediana.
taxa max	Razão entre a maior soma de distâncias com a máxima conectividade e a soma do problema da p -mediana
taxa min	Razão entre a menor soma de distâncias com a máxima conectividade e a soma do problema da p -mediana
$\sum \text{dif}$	Menor soma das diferenças entre o número de caminhos vértice-disjuntos dos clientes da p -mediana e o κ_2 desses clientes.
num clie com dif	Número de clientes da p -mediana que não têm a conectividade máxima.
$\max \text{dif}(c)$	A maior diferença entre o número de caminhos vértice-disjuntos de um cliente da p -mediana e o κ_2 desse cliente.
$\kappa_2(c)$ com max dif	O maior κ_2 de um cliente com a maior diferença.
dif médio	A razão entre a soma das diferenças e o número de clientes que têm alguma diferença.

expressão “MaxC max($\sum \text{dist}$)” na tabela 2. A menor soma possível está sob a expressão “MaxC min($\sum \text{dist}$)”.

Para critério de comparação, calculou-se a menor soma possível sem as restrições de conectividade, isto é, foram calculadas as soluções para o problema da p -mediana. Os valores da p -mediana encontram-se sob a expressão $p\text{-med} \sum \text{dist}$.

As taxas na tabela 2 correspondem às razões entre as somas de distâncias obtidas pela problema de máxima conectividade e a soma ótima obtida pelo problema da p -mediana, isto é:

$$\text{taxa max} = \frac{\text{MaxC max}(\sum \text{dis})}{p\text{-med} \sum \text{dist}} \quad \text{e} \quad \text{taxa min} = \frac{\text{MaxC min}(\sum \text{dis})}{p\text{-med} \sum \text{dist}}.$$

Comparando-se a maior taxa com a menor taxa, observa-se que compensa minimizar a soma das distâncias para o problema de máxima conectividade, visto que há soluções cujas somas de distâncias são até 6 vezes maiores do que a solução otimizada para a máxima conectividade, como ocorre com o grafo Us Carrier 2008.

Observando a menor taxa, verifica-se que nesses grafos a soma de distâncias da solução para o problema máxima conectividade com a menor soma de distâncias é até 66% maior que a soma obtida pelo problema da p -mediana. Já a média é de 20% maior. De acordo com o contexto, essas diferenças podem ser significativamente grandes.

Após a comparação das somas das distâncias entre o problema de máxima conectividade e a p -mediana, também foram feitas as comparações referentes à conectividade entre ambos os problemas. Como pode haver mais de uma solução possível para o problema da p -mediana, usou-se para essas comparações a solução com a maior conectividade que de acordo com a fórmula 19 minimiza *somaDif*.

Observando o número de clientes da p -mediana que contêm alguma diferença (“dif”) entre $\kappa_2(c)$ e $\kappa(s, c)$, na tabela 2 estão sob a expressão “num clie com dif”,

Tabela 2. Resultados experimentais para o problema de máxima conectividade e suas comparações

Nome	$ V $	$ A $	$\max \kappa_2(v)$	p	$\frac{p}{ V }$	MaxC max (\sum dist)	MaxC min (\sum dist)	p -med \sum dist	taxa max	taxa min	\sum dif	num clie com dif	max dif (c)	$\kappa_2(c)$ com max dif	dif médio
GARR 2011	60	87	5	2	0.03	162	113	110	1.47	1.03	7	5	2	5	0.31
ASN-AM 2008	65	79	4	2	0.03	167	114	113	1.48	1.01	2	1	2	4	0.50
Internode 2010	66	78	4	3	0.05	217	110	109	1.99	1.01	1	1	1	4	0.25
Missouri 2010	67	83	4	5	0.07	454	137	128	3.55	1.07	5	5	1	4	0.28
GlobeNet 2009	67	113	5	8	0.12	309	114	92	3.36	1.24	13	10	3	5	0.34
ESnet 2007	68	92	4	3	0.04	274	125	119	2.30	1.05	7	6	2	4	0.36
LATNET 2007	69	74	3	1	0.01	144	144	144	1	1	0	0	0	0	0
Uninett 2011	69	98	6	5	0.07	306	129	117	2.62	1.10	9	6	3	6	0.31
Columbus 2010	70	85	4	4	0.06	570	202	159	3.58	1.27	4	4	1	4	0.31
Intellifiber 2010	73	97	4	9	0.12	377	116	100	3.77	1.16	10	10	1	4	0.33
Sinet 2010	74	76	3	2	0.03	267	137	136	1.96	1.01	1	1	1	3	0.33
SWITCH 2010	74	92	4	9	0.12	533	106	87	6.13	1.22	2	2	1	3	0.33
Bestel 2006	84	101	3	3	0.04	702	476	297	2.36	1.60	18	17	2	3	0.48
Viatel 2008	88	92	3	3	0.03	1401	566	440	3.18	1.29	4	4	1	3	0.33
Viatel 2011	92	96	3	3	0.03	1487	620	479	3.10	1.29	3	3	1	3	0.33
Interoute 2010	110	158	4	7	0.06	899	256	212	4.24	1.21	18	16	2	4	0.37
ITC Deltacom 2010	113	183	5	8	0.07	987	238	192	5.14	1.24	6	6	1	4	0.31
ION 2007	125	150	3	5	0.04	1458	407	367	3.97	1.11	8	8	1	3	0.40
PERN 2010	127	129	3	2	0.02	500	366	221	2.26	1.66	7	5	2	3	0.50
TATA 2010	145	194	5	9	0.06	1693	366	288	5.88	1.27	13	13	1	5	0.32
GTS CE 2010	149	193	4	12	0.08	1638	325	248	6.60	1.31	14	14	1	4	0.31
Colt Telecom 2010	153	191	3	11	0.07	1409	266	222	6.35	1.20	18	17	2	3	0.39
US Carrier 2008	158	189	3	13	0.08	1937	371	292	6.63	1.27	16	15	2	3	0.43
Cogent 2010	197	245	6	13	0.07	2201	464	393	5.60	1.18	23	22	2	6	0.38

constata-se que podem compor até 20% dos vértices do grafo, como ocorre com o grafo Bestel 2008. Se analisarmos os vértices com os maiores “difs” e compararmos com os $\kappa_2(c)$ desses vértices, verificamos que esses vértices têm de 25% (1/4) até 60% (3/5) menos caminhos vértice disjuntos com o servidor do que teriam sob as condições de máxima conectividade (excetuando o caso em que não há diferenças como para Latnet 2007).

Podemos também verificar a média das diferenças, sob a expressão “dif médio” na tabela 2, que corresponde à fórmula:

$$\text{dif médio} = \frac{\sum_{c \in C} (\kappa_2(c) - \kappa(s, c)) / \kappa_2(c)}{\text{num clie com dif}}$$

Por essas medidas observa-se que nesses grafos, e no problema da p -mediana, há uma diferença média de 25% a 50%, e na média de todos os grafos 36%, na diferença relativa entre a conectividade máxima e o número de caminhos vértice disjuntos entre cliente e servidores, dentre aqueles vértices que apresentam alguma diferença com relação à máxima conectividade.

7. Conclusão

Este trabalho apresentou o problema de posicionar o número mínimo de recursos em uma rede de modo a maximizar o número de caminhos vértice-disjuntos entre os consumidores e o recurso. Denominamos esse problema de alocação de recursos com a máxima conectividade. Provou-se que o problema de encontrar o número mínimo de recursos sob essas condições é NP-difícil. Uma solução exata para o problema foi implementada usando programação linear inteira. Ela foi testada em grafos arbitrários conexos com 60 a 200 vértices.

Os resultados experimentais mostram que para esses grafos o número mínimo de servidores necessários é relativamente pequeno, em média 6% do número de vértices do grafo. Mostrou-se também que compensa otimizar o problema de máxima conectividade com relação à soma das distâncias, devido a diferença entre as somas das soluções possíveis. Por exemplo, no caso do grafo Us Carrier 2008, a soma das distâncias da pior solução para a melhor foi 6 vezes maior.

Esses resultados também foram comparados com o problema da p -mediana, caso em que é minimizada a soma das distâncias entre os clientes e seus servidores. Do problema da máxima conectividade para o da p -mediana, observou-se um aumento em média de 20% para a soma das distâncias. Quanto à conectividade, o problema da p -mediana apresentou clientes com uma defasagem média de 36% de caminhos vértice-disjuntos com relação à conectividade máxima.

Trabalhos futuros incluem implementar a solução proposta tanto no contexto de redes CDN como de redes SDN.

Agradecimentos

Agradecemos os comentários e sugestões dos revisores anônimos. Esse trabalho foi parcialmente financiado pelos projetos CNPQ 311451/2016-0 e 428941/2016-8.

Referências

- [Bae 1997] Bae, M. M. (1997). Resource placement in torus-based networks. *IEEE Transactions on Computers*, 46(10):1083–1092.
- [Cormen et al. 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.
- [Heller et al. 2012] Heller, B., Sherwood, R., and McKeown, N. (2012). The Controller Placement Problem. *ACM SIGCOMM Computer Communication Review*, 42(4):473.
- [Ito et al. 2002] Ito, H., Ito, M., Itatsu, Y., Nakai, K., Uehara, H., and Yokoyama, M. (2002). Source Location Problems Considering Vertex-Connectivity and Edge-Connectivity Simultaneously. *Networks*, 40(2):63–70.
- [Li et al. 1999] Li, B., Golin, M. J., Italiano, G. F., Deng, X., and Sohraby, K. (1999). On the optimal placement of web proxies in the internet. In *INFOCOM 1999*, volume 3, pages 1282–1290. IEEE.
- [Makino 2012] Makino, K. (2012). Source Location Problems with Flow Requirements. *2012 Third International Conference on Networking and Computing*, pages 404–406.
- [Muller et al. 2014] Muller, L. F., Oliveira, R. R., Luizelli, M. C., Gaspar, L. P., and Barcellos, M. P. (2014). Survivor: An Enhanced Controller Placement Strategy for Improving SDN Survivability. In *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pages 1909–1915.
- [Pires et al. 2011] Pires, K., Cohen, J., and Duarte Jr., E. P. (2011). Medidas de conectividade baseadas em cortes de vértices para redes complexas. In *12 Workshop de Testes e Tolerância a Falhas (WTF'2011), Anais do SBRC'2011*.
- [Qiu et al. 2001] Qiu, L., Padmanabhan, V. N., and Voelker, G. M. (2001). On the placement of web server replicas. In *INFOCOM 2001*, volume 3, pages 1587–1596. IEEE.
- [Rao et al. 2010] Rao, W., Chen, L., Fu, A. W.-C., and Wang, G. (2010). Optimal resource placement in structured peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(7):1011–1026.
- [Reese 2006] Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142.
- [Rochman et al. 2013] Rochman, Y., Levy, H., and Brosh, E. (2013). Resource placement and assignment in distributed network topologies. In *INFOCOM 2013*, pages 1914–1922. IEEE.
- [Rodolakis et al. 2006] Rodolakis, G., Siachalou, S., and Georgiadis, L. (2006). Replicated server placement with qos constraints. *IEEE Transactions on Parallel and Distributed Systems*, 17(10):1151–1162.
- [Sahoo et al. 2017] Sahoo, J., Salahuddin, M., Glitho, R., Elbiaze, H., and Ajib, W. (2017). A Survey on Replica Server Placement Algorithms for Content Delivery Networks. *IEEE Communications Surveys & Tutorials*, 19(2):1002–1026.
- [Younis and Akkaya 2008] Younis, M. and Akkaya, K. (2008). Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655.