

UNIVERSIDADE FEDERAL DO PARANÁ

HENRIQUE HEPP

**ALOCÇÃO DE RECURSOS COM MÁXIMA CONECTIVIDADE EM REDES
COM TOPOLOGIA ARBITRÁRIA**

CURITIBA PR

2018

HENRIQUE HEPP

**ALOCAÇÃO DE RECURSOS COM MÁXIMA CONECTIVIDADE EM REDES COM
TOPOLOGIA ARBITRÁRIA**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Elias P. Duarte Jr.

Coorientador: Prof. Dr. Jaime Cohen.

CURITIBA PR

2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

H529a

Hepp, Henrique

Alocação de recursos com máxima conectividade em redes com topologia arbitrária / Henrique Hepp. – Curitiba, 2018.

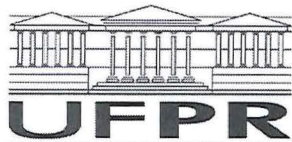
Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2018.

Orientador: Elias Procópio Duarte Junior – Coorientador: Jaime Cohen.

1. Redes de computadores. 2. Conectividade (computadores). 3. Cliente/Servidor (Computadores). I. Universidade Federal do Paraná. II. Duarte Junior, Elias Procópio. III. Cohen, Jaime. IV. Título.

CDD: 004.66

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **HENRIQUE HEPP** intitulada: **Alocação de Recursos com Máxima Conectividade em Redes com Topologia Arbitrária.**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua a aprovação no rito de defesa. A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 27 de Agosto de 2018.

ELIAS PROCÓPIO DUARTE JUNIOR
Presidente da Banca Examinadora (UFPR)

MARINA ESTHER GROSHAUS
Avaliador Externo (UBA)

RENATO JOSÉ DA SILVA CARMO
Avaliador Interno (UFPR)

JAIME COHEN
Coorientador - Avaliador Externo (UEPG)



O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

À minha família.

Agradecimentos

- A Deus
- Aos meus orientadores de mestrado Elias e Jaime.
- A todas pessoas que direta ou indiretamente contribuíram para a elaboração deste trabalho.

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 da p -mediana em que é minimizada a soma das distâncias entre os clientes e seus recursos. Resultados experimentais usando redes de *The Internet Topology Zoo* avaliam o ganho de conectividade e o aumento da soma das distâncias quando a solução proposta é aplicada.

Palavras-chave: Alocamento de Recursos, vértice-conectividade, localização de origens, p -mediana.

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 this work 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 for several arbitrary topology networks. This work presents a comparison between the connectivity based resource location problem with the classical p -median problem in which the sum of the distances between clients and their resources is minimized. Experimental results using networks from The Internet Topology Zoo show both the connectivity gains and the impact on the sum of the distances when the proposed solution is applied to real Internet-based networks.

Keywords: Resource Allocation, vertex-connectivity, source location, p -median.

Lista de Figuras

2.1	Grafo com os valores da 2-vértice-connectividade..	12
3.1	Exemplo de um grafo cuja solução para o problema da p -mediana não é uma solução para o problema AR-MaxC.	18
3.2	Redução de uma aresta $\{x_i, x_j\}$	19
3.3	Exemplo de uma redução..	20
4.1	Soma das distâncias entre clientes e seus servidores para os casos: soma máxima e mínima para o problema de máxima conectividade e o problema da p -mediana.	27
4.2	Razões entre as somas das distâncias mínimas entre clientes e servidores para o problema AR-MaxC e para o problema da p -mediana acrescentando mais servidores..	31
4.3	Número de caminhos vértice-disjuntos das soluções da p -mediana otimizada e a diferença para a máxima conectividade (para os vértices com a maior diferença)..	31

Sumário

1	Introdução	10
2	Definições e Trabalhos Relacionados	12
2.1	Grafos e Conectividade	12
2.2	Medidas de Conectividade Baseadas em Cortes de Vértices	13
2.3	Alocação de Recursos	13
2.4	p -Mediana	14
2.5	Posicionamento de Fontes de Dados	14
2.6	Localização Usando Diversidade de Caminhos	15
3	Alocação de Recursos com a Máxima Conectividade	17
3.1	Demonstração da NP-Compleitude	18
3.2	Formulações dos Problemas de Programação Linear Inteira	22
3.2.1	PLI para o problema AR-MaxC	22
3.2.2	Seleção do Conjunto de Servidores Considerando Distâncias	23
3.2.3	Comparação com a p -Mediana	24
4	Implementação e Resultados Experimentais	25
5	Conclusão	32
	Referências	33

1 Introdução

O problema do posicionamento de recursos adquire as mais diversas formas nas redes de computadores [1]. Na arquitetura cliente-servidor há várias versões para o problema como o posicionamento de servidores com o objetivo de minimizar a latência para os clientes [2] 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 [3]. Nas redes CDN (*Content-Delivery Network*) o problema é semelhante, mas consiste em posicionar réplicas de conteúdos específicos dado um padrão de tráfego [4] 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 [5]. 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 [6]. O problema assume, em geral, uma rede de topologia arbitrária, mas há variações para topologias específicas [7]. Recentemente, este problema ganhou ainda uma nova aplicação nas redes SDN (*Software Defined Networking*), para o posicionamento de controladores [8, 9].

Neste trabalho definimos uma nova versão do problema, que visa tirar partido da máxima vértice-conectividade dos nodos na rede. Para facilitar a exposição, usamos nesta dissertação 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 para cada cliente exista um servidor tal que a conectividade entre o cliente e este servidor seja igual a máxima conectividade entre o cliente e qualquer outro nodo da rede. Por máxima conectividade entre o cliente e o servidor entende-se o maior número possível de caminhos vértice-disjuntos entre eles. Desse modo, garante-se a maior disponibilidade possível de caminhos alternativos entre clientes e servidores para uma rede.

Uma das contribuições do trabalho é a prova de que o problema de encontrar o número mínimo de recursos sob essas condições é NP-difícil. Outra contribuição é a quantificação experimental em grafos reais que representam “backbones” de internet do número mínimo de servidores com a máxima conectividade e a soma de distâncias entre eles e os clientes. Considerando-se que a conectividade e a distância entre clientes e servidores são dois fatores significativos para os clientes em uma rede, essas quantias indicam o potencial uso da proposta em aplicações em redes. Como nem sempre é possível ter a maior conectividade e a menor soma de distâncias entre clientes e servidores ao mesmo tempo, procurou-se mostrar o *trade-off* entre as duas medidas. Ou seja, ao maximizar a conectividade, observou-se qual é o aumento da soma das distâncias entre clientes e servidores; e ao minimizar a soma de distâncias, observou-se qual é a perda de conectividade dos clientes.

Implementou-se uma solução exata para o problema da alocação de recursos com máxima conectividade usando programação linear inteira. Ela foi testada em grafos conexos com número de vértices entre 60 e 200. Observou-se que para esse grafos há várias soluções possíveis, possibilitando que esse problema seja ainda otimizado por outros critérios. Optou-se

por encontrar soluções com máxima conectividade que tenham a menor soma das distâncias entre os clientes e seus servidores. Esses resultados foram, então, comparados com o problema conhecido como p -mediana, caso em que é minimizada a soma das distâncias entre os clientes e os seus servidores. A solução do problema da alocação de recursos com máxima conectividade apresentou um aumento médio de 20% para a soma das distâncias em relação ao problema da p -mediana. Quando mais servidores que o mínimo necessário são adicionados, a diferença entre as duas soluções torna-se significativamente menor. Já o problema da p -mediana apresentou soluções em que clientes têm menos caminhos vértice-disjuntos com os seus servidores do que naquelas que maximizam a conectividade. Nesses clientes a defasagem média do número de caminhos vértice-disjuntos para seus servidores é de 36%.

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

2 Definições e Trabalhos Relacionados

Neste capítulo são dadas definições preliminares referentes a grafos e à vértice-conectividade que serão usadas nos próximos capítulos. Também são apresentados trabalhos relacionados referentes a problemas de alocação de recursos.

2.1 Grafos e Conectividade

Um *grafo*, ou *multigrafo* [10], $G = (V, E)$ 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 *grafo capacitado* $G = (V, E, c)$ é uma tripla na qual (V, E) é um grafo e $c : E \rightarrow \mathbb{Q}_{\geq 0}$ é uma função que define as capacidades das arestas em E . $\mathbb{Q}_{\geq 0}$ denota o conjunto dos números racionais não negativos.

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 $i \in \{1, 2, \dots, k-1\}$.

Um grafo G é dito *conexo* se existe um caminho entre todo par de vértices de G .

A *distância* em entre dois vértices v e w é o número de arestas de um caminho mínimo entre os vértices v e w .

Dados dois vértices $s, t \in V$, um conjunto de s - t -caminhos é *aresta-disjunto* se em cada par de caminhos do conjunto não houver arestas em comum.

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 maior número de caminhos vértice-disjuntos entre s e t .

A *vértice-conectividade* entre um conjunto S e um vértice t , denotada por $\kappa(S, t)$, é o maior número de caminhos entre os vértices em S e t que sejam vértice-disjuntos em $V - S - t$.

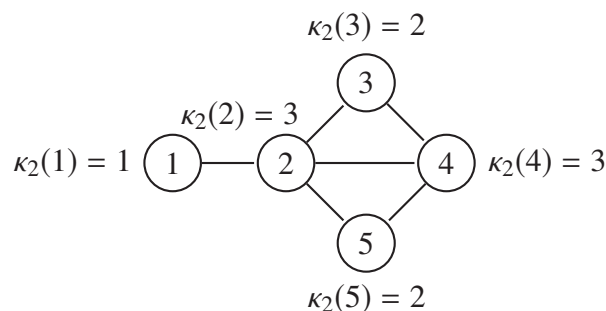


Figura 2.1: Grafo com os valores da 2-vértice-connectividade.

2.2 Medidas de Conectividade Baseadas em Cortes de Vértices

Nesta seção são definidas as medidas de conectividade baseadas em corte de vértices para um conjunto de vértices [10] e a medida de conectividade para vértices proposta por K. Pires et al. em 2011 [11, 12]:

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

Definição 2. Dado um grafo $G = (V, E)$ e $i \in \mathbb{N}$, $2 \leq i \leq |V|$. A *i -vértice-conectividade de um vértice v em G* , 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 algum outro vértice de G , ou de modo mais formal:

Definição 3. Dado um grafo $G = (V, E)$, a *2-vértice-conectividade de v* denotada por $\kappa_2(v)$ é o maior $\kappa(v, x)$ para todo $x \in V$.

A Figura 2.1 mostra um exemplo. O vértice 2 tem 3 caminhos vértice-disjuntos com o vértice 4 e como o vértice 2 não tem um número maior de caminhos vértice-disjuntos com qualquer outro vértice, $\kappa_2(2) = 3$. A 2-vértice-conectividade pode ser calculada em tempo polinomial como mostrado pelo Lema 1.

Lema 1. A 2-vértice-conectividade pode ser calculada em tempo polinomial.

Demonstração. Para calcular a 2 vértice-conectividade de um vértice v em um grafo $G = (V, E)$, calculamos a vértice-conectividade local entre v e os demais vértices em G e escolhemos o maior valor. Como a vértice-conectividade local entre 2 vértices pode ser calculada em tempo polinomial (como mostrado em [10]), a 2 vértice-conectividade pode ser calculada em tempo polinomial. \square

2.3 Alocação de Recursos

Atualmente, em redes de computadores, existem diversos recursos, como servidores de banco de dados, réplicas de servidores, máquinas virtuais, entre outros, que podem ser alocados a consumidores distribuídos pela rede, o que chamamos de alocação de recursos (*resource allocation*). É comum o posicionamento desses recursos, e sua alocação, ser feito de acordo com a formulação dos problemas de posicionamento de instalações (*facility location problems*) [13] que são problemas que objetivam encontrar as melhores localizações para instalações de modo a otimizar alguma propriedade, como, por exemplo, posicionar fábricas de modo a diminuir os custos de instalação e das entregas de produtos aos clientes [14, 15]. Um outro exemplo é o problema conhecido por Localização de Instalações com Tolerância a Falhas (*Fault-Tolerance Facility Location*) em que para cada cliente são alocados um determinado número, digamos k , de instalações, com o menor custo possível de instalação e conexão. Nesse caso, se uma das instalações falhar o cliente ainda estará conectado a $k - 1$ instalações [16].

2.4 p -Mediana

Entre os problemas de localização de instalações, a otimização de medidas relacionadas às distâncias entre instalações e consumidores é muito comum [17]. Neste trabalho também procuramos otimizar as distâncias entre recursos e consumidores e comparamos essa otimização com o problema da p -mediana, que tem como objetivo minimizar a soma das distâncias entre p instalações e seus consumidores.

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 nodos de uma rede com n nodos, os demais nodos correspondem a clientes. O objetivo é fazer o posicionamento das réplicas de forma a minimizar a soma das distâncias entre os cliente e as réplicas [18, 19].

O problema que deu origem ao problema da p -mediana geralmente é atribuído a Fermat no século XVII: dados três pontos em um plano, encontre um quarto ponto, a mediana, de modo que a soma das distâncias dele aos três pontos seja a menor possível. Esse problema foi resolvido de forma geométrica por Torricelli [20, 21]. No início do século XX, Alfred Weber estudou esse problema perguntando qual seria o melhor local para instalar uma fábrica de modo que o custo de transporte entre ela e três clientes fosse o menor possível. No lugar de usar apenas as distâncias geométricas, ele adicionou pesos a essas distâncias. Esse problema ficou conhecido como o problema de Weber [19, 22]. Hakimi na década de 1960 estudou esse problema para o caso com mais de 3 pontos em um grafo capacitado. Ele permitiu que o ponto a encontrar, a mediana, pudesse estar tanto sobre os vértices como sobre as arestas do grafo e provou que a mediana sempre será um vértice desse grafo [23]. Esse problema é chamado de 1-mediana [19]. Mais tarde S. L. Hakimi generalizou esse resultado para o problema p -mediana: nele busca-se p pontos de modo que a soma das distâncias entre esses pontos e os vértices desse grafo seja a menor possível. Ele provou que sempre existe ao menos uma solução em que esses pontos são vértices do grafo [24]. Na década de 1970, O. Kariv e S. L. Hakimi provaram que o problema da p -mediana é NP-difícil [25].

2.5 Posicionamento de Fontes de Dados

Outro problema de localização é o de posicionamento de origens, ou fontes de dados (*source location*). Nesse caso são consideradas as conectividades e/ou as quantidades de fluxo entre as instalações e os clientes. Esses problemas foram introduzidos na década de 1990 por H. Tamura et al. [26, 27] que analisaram alguns problemas de localização, entre eles o da p -mediana, levando em consideração a aresta-conectividade entre instalações e clientes.

De forma geral, o problema de localização de origens consiste em posicionar um conjunto de instalações de modo que possam atender as demandas dos consumidores por algum serviço. Caso se queira contemplar a tolerância a falhas, pode-se requerer que o conjunto de instalações tenha mais caminhos aresta e/ou vértice-disjuntos com o cliente que o absolutamente necessário para atender sua demanda. Por exemplo, se o cliente necessita de 1 caminho para o conjunto de instalações, pode-se exigir 1 caminho adicional que seja aresta e/ou vértice-disjunto, para que caso o primeiro caminho falhe o segundo possa ser usado.

Uma formulação geral desse problema em grafos 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: $\sum_{x \in S} w(x)$

Sujeito 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 [28].

Diversas variantes desse problema já foram estudadas [29]. Para este trabalho queremos citar problemas relacionados com a vértice-conectividade.

Dado um grafo $G = (V, E)$ não capacitado e uma constante inteira k , queremos:

Minimizar: $|S|$

Sob as restrições: $S \subseteq V$ e $\kappa(S, x) \geq k$, $\forall x \in V - S$

Foi provado por H. Ito et al. em 2002 [28] que esse problema é NP-difícil para $k \geq 3$. No problema analisado por H. Ito et al. garante-se que o conjunto de instalações tenha k caminhos vértice-disjuntos com o cliente. Mas poderá ocorrer que dentro desse conjunto apenas uma instalação tenha caminhos vértice-disjuntos com o cliente, e, portanto, se essa instalação falhar o cliente deixará de ser atendido. Para evitar que isso ocorra, H. Nagamochi et al. em 2001 [30] propuseram uma nova versão do problema usando na restrição a medida $\hat{\kappa}(S, x)$ no lugar de $\kappa(S, x)$. A medida $\hat{\kappa}(S, x)$ é definida como sendo o maior número de caminhos de S para x que sejam vértice-disjuntos entre eles com a exceção de x . Nessa nova versão, se uma instalação conectada com x falhar, garante-se que ainda existem $k - 1$ instalações no conjunto S conectadas ao cliente x . H. Nagamochi et al. provaram que esse problema pode ser resolvido em tempo polinomial [30]; já para o caso em que a demanda não é igual a uma constante, mas é uma função arbitrária, T. Ishii et al. provaram que o problema é NP-difícil [31].

2.6 Localização Usando Diversidade de Caminhos

Diversos trabalhos na área de redes de computadores já propuseram o uso de caminhos múltiplos entre um recurso e um consumidor para melhorar a transferência de dados ou a tolerância a falhas. Para tal, pode-se explorar a diversidade de caminhos (*path diversity*) entre esses nodos [32, 33]. A diversidade de caminhos pode ser expressa pelo conjunto de caminhos aresta e/ou vértice-disjuntos entre dois nodos [34], ou também pela seleção de caminhos que sejam o mais distintos possíveis de acordo com alguma métrica, por exemplo, selecionando caminhos que tenham o menor número de vértices e arestas em comum [35]. Nesta seção citamos alguns problemas de localização que exploram a diversidade de caminhos.

Uma área em que a diversidade de caminhos é bastante explorada é em roteamento em redes de sobreposição (*overlay network*) [36]. Uma rede de sobreposição instancia uma rede virtual sobre uma rede física usando um conjunto de nodos de sobreposição, ou também chamados de *relay*, acima do roteamento por IP. Os nodos de sobreposição fornecem os caminhos de roteamento entre nodos de origem e destino [37]. Quanto a problemas de localização, estudou-se o problema de posicionamento de nodos de sobreposição de modo a otimizar alguma propriedade como latência, ou o aumento da diversidade de caminhos [38]. Um exemplo, seria o de M. Cha et al. (2006) que propuseram o problema de posicionamento de nodos *relay* para roteamento intra-domínio [39]. Nesse contexto, existe um caminho padrão entre cada par de roteadores, sendo um a origem e o outro o destino. Para aumentar a tolerância a falhas propôs-se disponibilizar um caminho adicional, chamado de caminho de sobreposição (*overlay path*), que passa por um nodo *relay*. O problema consiste em posicionar k nodos *relay* de modo que para cada par de nós origem e destino haja um caminho de sobreposição que seja o mais distinto possível do caminho padrão. Em uma outra proposta, V. Bui et al. (2008) procuram encontrar o

menor número de nodos *relay* de modo a maximizar a diversidade de caminhos entre os nós de origem e destino em toda rede. Nesse caso, os nodos *relay* proporcionam k caminhos o mais distintos possíveis entre todos os nós de origem e destino [40].

Um trabalho que se assemelha ao problema proposto para esta dissertação foi a estratégia chamada de “survivor” implementada em redes SDN por L. F. Müller et al. em 2014 [9]. L. F. Müller et al. propuseram posicionar controladores de modo a maximizar, com relação a toda a rede, o número de caminhos vértice-disjuntos entre controladores e dispositivos de rede sob restrições de posicionamento e capacidade dos controladores. Esse artigo mostrou que explorando a diversidade de caminhos, a perda de conectividade entre controladores e dispositivos é significativamente menor.

3 Alocação de Recursos com a Máxima Conectividade

Esta seção apresenta o novo problema da alocação de recursos 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 de *problema de alocação de recursos com a máxima conectividade* ou, de forma abreviada, problema AR-MaxC.

Tendo sido definido o maior número possível de caminhos vértice-disjuntos para um vértice v como sendo sua 2-vértice-conectividade ou $\kappa_2(v)$ podemos definir o problema AR-MaxC como: dado um grafo conexo não direcionado $G = (V, E)$, queremos encontrar um conjunto mínimo de vértices $S \subset V$, de modo que para todo $v \in V$, existe $s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$. Para simplificar a notação, estendemos a definição de $\kappa_2(v)$ de forma que $\kappa(v, v) = \kappa_2(v)$, desse modo um vértice poderá ser cliente dele mesmo. Podemos também formular o problema como:

Entrada: $G = (V, E)$, um grafo conexo

Saída: $S \subset V$

Função Objetivo: $\min |S|$

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

Observa-se que em grafos desconexos, basta resolver o problema separadamente em cada componente conexo.

Como exemplo, podemos verificar a solução para o grafo apresentado na Figura 2.1. Pode-se observar que todos os vértices podem ser clientes do vértice 2, pois o número de caminhos vértice-disjuntos entre eles e o vértice 2 é igual a 2-vértice-conectividade deles. Portanto, o vértice 2 é uma solução para o problema AR-MaxC. Outra solução ótima para o o problema é o vértice 4.

Poderemos querer, dentre essas soluções, minimizar a soma de distâncias entre os clientes e seus servidores. Nesse caso, a solução é o conjunto corresponde ao vértice 2, pois a soma de distâncias entre ele e os demais vértices é igual a 4, já a soma entre o vértice 4 e os demais é igual a 5. Nesse caso, a solução com a menor soma de distâncias é idêntica à solução dada pelo problema da p -mediana, o que não ocorre em geral como podemos ver pelo exemplo dado na Figura 3.1. Nessa figura, a solução para o problema da máxima conectividade que tem a menor soma de distâncias entre clientes e servidores corresponde ao vértice 5 como único servidor. Já para o problema da p -mediana a solução é o vértice 4. Agora, se permitirmos 2 servidores na solução, as soluções para o problema otimizado do AR-MaxC e da p -mediana têm a mesma soma de distâncias para os clientes, tendo como solução os vértices 2 e 5.

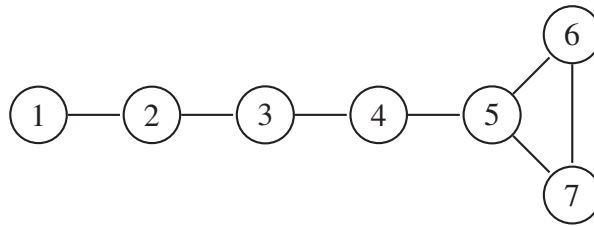


Figura 3.1: Exemplo de um grafo cuja solução para o problema da p -mediana não é uma solução para o problema AR-MaxC.

3.1 Demonstração da NP-Compleitude

Esta seção apresenta a prova de NP-completude da versão de decisão do problema AR-MaxC. O problema de decisão consiste em determinar 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 1. *A versão de decisão do problema de alocação de recursos com a 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 2 e a segunda no Lema 3.

Lema 2. *A versão de decisão da alocação de recursos com a máxima conectividade pertence à classe 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 certifica se $|S| \leq p$ 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 de tempo polinomial, veja o Lema 1, essa verificação pode ser feita em tempo polinomial. \square

Lema 3. *A versão de decisão da Alocação de Recursos com a Máxima Conectividade é NP-difícil.*

Esse lema pode ser provado mostrando que existe uma redução de tempo polinomial do problema de decisão da cobertura por vértices para o problema de decisão de alocação de recursos com máxima conectividade. O problema de decisão da cobertura por vértices é um problema NP-completo bem conhecido [41]. 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 menor ou igual p se, e somente se, o grafo reduzido tem um conjunto de servidores de tamanho menor ou igual a p que resolve o problema AR-MaxC. Isso é demonstrado pelos Lemas 8 e 9. O Lema 8 demonstra que se existe uma solução para o problema de cobertura de vértices de tamanho menor ou igual a p então existe uma solução de tamanho menor ou igual

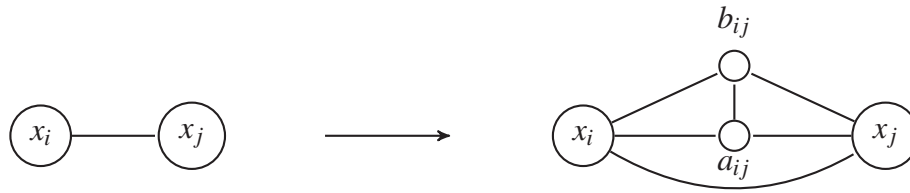


Figura 3.2: Redução de uma aresta $\{x_i, x_j\}$.

a p para o problema AR-MaxC. O Lema 9 demonstra que se existe uma solução do problema AR-MaxC de tamanho menor ou igual a p então existe uma solução de tamanho menor ou igual a p do problema de cobertura por vértices.

A versão de decisão do problema de cobertura por vértices pode ser definida 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 menor ou igual que 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 [42].

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

Saída:

sim, se $\exists V_c$, com $|V_c| \leq p$ 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 em duas partes. Primeiro, dado $G_0 = (V_0, E_0)$ constrói-se o grafo $G' = (V, E')$, e a partir dele constrói-se o grafo $G = (V, E)$. Segue o algoritmo.

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

$$V = V_0 \cup V' \quad (3.1)$$

$$V' = \bigcup_{\{x_i, x_j\} \in E_0} \{a_{ij}, b_{ij}\} \quad (3.2)$$

$$E' = \bigcup_{\{x_i, x_j\} \in E_0} \{\{x_i, x_j\}, \{x_i, a_{ij}\}, \{x_i, b_{ij}\}, \{x_j, a_{ij}\}, \{x_j, b_{ij}\}, \{a_{ij}, b_{ij}\}\} \quad (3.3)$$

A redução até aqui para cada subgrafo composto pela aresta $\{x_i, x_j\}$ pode ser observada na Figura 3.2. Continuando a redução, construímos o grafo $G = (V, E)$ da seguinte forma:

$$E = E' \cup E'' \quad (3.4)$$

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

1. Encontre o maior valor de $\kappa_2(v)$, $v \in V_0$, 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 necessário 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 . O Lema 4 justifica realizar esse passo.

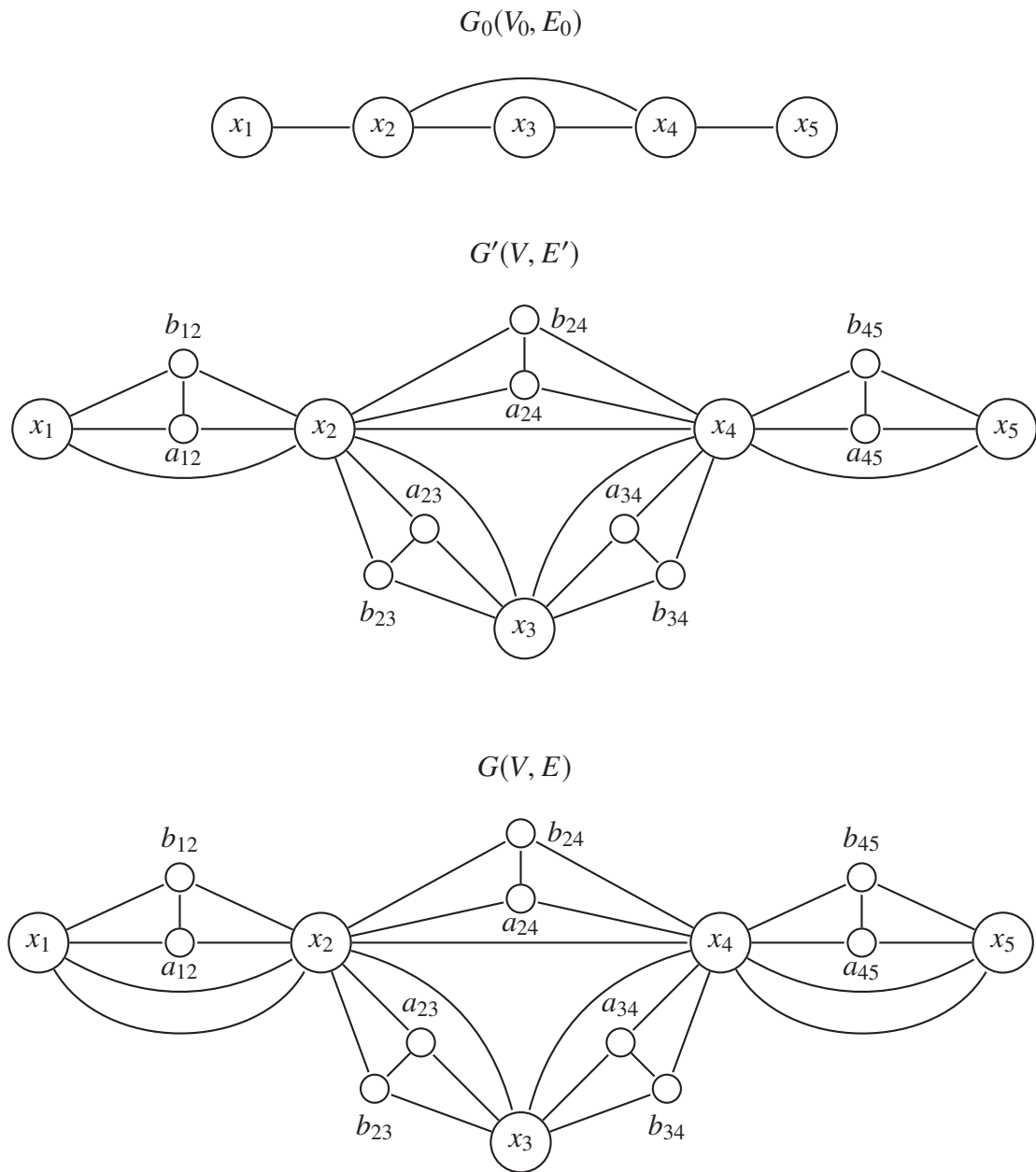


Figura 3.3: Exemplo de uma redução.

Lema 4. A adição de mais arestas $\{x_i, x_j\}$ não altera a vértice-conectividade local e a 2-vértice-conectividade dos vértices de G' com a exceção dos vértices x_i e x_j .

Demonstração. O lema segue da seguinte observação: a adição de arestas paralelas $\{x_i, x_j\}$ a uma aresta existente não altera o número de caminhos vértice-disjuntos entre qualquer par de vértices com a exceção do par $\{x_i, x_j\}$. \square

Podemos ver um exemplo de uma redução na Figura 3.3. 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 3.1 a 3.3 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 $\kappa(x_i, x_j)$ em G são iguais.

Antes de provarmos os lemas 8 e 9, mostramos os lemas 5 a 7 que serão usados na demonstração dos lemas 8 e 9. Todos os lemas se referem a um grafo G obtido pela redução.

Lema 5. *A vértice-conectividade local dos vértices x_i e x_j para com os vértices a_{ij} e b_{ij} é sempre igual a 3 para qualquer grafo G .*

Demonstração. O vértice x_i tem para com a_{ij} 3 caminhos vértice-disjuntos: o primeiro caminho sendo $\{x_i, a_{ij}\}$, o segundo passando por b_{ij} e o terceiro passando por x_j . Como a_{ij} tem o grau igual a 3 não há a possibilidade de existirem mais caminhos vértice-disjuntos para a_{ij} . Os mesmos argumentos podem ser usados de x_i para com b_{ij} e de x_j para com a_{ij} e b_{ij} . \square

Lema 6. *A 2-vértice-conectividade dos vértices a_{ij} e b_{ij} é sempre igual a 3 para qualquer grafo G .*

Demonstração. O grau de a_{ij} e b_{ij} limita a 2-vértice-conectividade de a_{ij} e b_{ij} em 3. Como a vértice-conectividade local de a_{ij} e b_{ij} com x_i ou x_j é sempre igual a 3 (lema 5), $\kappa_2(a_{ij}) = 3$ e $\kappa_2(b_{ij}) = 3$. \square

Lema 7. *Dado $x \in V \setminus \{x_i, x_j, a_{ij}, b_{ij}\}$, então $\kappa(x, a_{ij})$ e $\kappa(x, b_{ij})$ são no máximo iguais a 2.*

Demonstração. Seja $x \in V \setminus \{x_i, x_j, a_{ij}, b_{ij}\}$. Qualquer caminho entre x e a_{ij} ou b_{ij} contém x_i ou x_j . Portanto, pode haver no máximo 2 caminhos vértice-disjuntos entre x e a_{ij} ou entre x e b_{ij} . \square

Os próximos lemas demonstram que G_0 contém uma cobertura de vértices de tamanho menor ou igual a p se, e somente se, G contém uma solução para o problema AR-MaxC de tamanho menor ou igual a p .

Lema 8. *Se V_c é uma solução do problema de cobertura de vértices de tamanho menor ou igual a p para o grafo G_0 , então existe uma solução do problema de alocação de recursos com máxima conectividade para o grafo G de tamanho menor ou igual a p .*

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 é uma solução para o problema AR-MaxC, 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 $\{a_{ij}, b_{ij}\}$ obtidos na redução da aresta $\{x_i, x_j\}$. Pois, como visto nos lemas 5 e 6, os vértices a_{ij} e $b_{i,j}$ têm $\kappa_2(v) = 3$ e $\kappa(x_i, v) = \kappa(x_j, v) = 3$ para qualquer $v \in \{a_{ij}, b_{ij}\}$.
2. x_i pode ser servidor de x_j ou vice-versa. O último passo da redução, a adição das arestas E'' , garante que $\kappa_2(x_i) = \kappa_2(x_j) = \kappa(x_i, x_j)$.

\square

Lema 9. *Se S é uma solução do problema de alocação de recursos com máxima conectividade de tamanho menor ou igual a p para o grafo G , então existe uma solução do problema por cobertura de vértices para o grafo G_0 de tamanho menor ou igual a p .*

Demonstração. Seja S uma solução do problema AR-MaxC para o grafo G . Então S também será uma solução para o problema de cobertura de vértices em G_0 , se o conjunto S conter x_i ou x_j ou ambos para toda aresta $\{x_i, x_j\} \in E_0$.

Pela redução, uma aresta $\{x_i, x_j\} \in E_0$ corresponde ao conjunto de vértices $\{a_{ij}, b_{ij}\}$ e ao de arestas $\{\{x_i, a_{ij}\}, \{x_i, b_{ij}\}, \{x_j, a_{ij}\}, \{x_j, b_{ij}\}, \{a_{ij}, b_{ij}\}\}$, como ilustrado na Figura 3.2.

Vamos mostrar que caso a solução não contenha nem x_i e nem x_j , então podemos transformar essa solução em 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 $\{a_{ij}, b_{ij}\}$, será necessário escolher um servidor do conjunto $\{a_{ij}, b_{ij}\}$, pois todos os demais vértices além de x_i e x_j terão apenas dois caminhos vértice-disjuntos com eles, como visto no lema 7. 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 $v \in \{a_{ij}, b_{ij}\}$, como visto nos lemas 5 e 6.

Desse modo, o procedimento acima produz uma solução para o problema de cobertura de vértices em G_0 com o mesmo tamanho de S . \square

3.2 Formulações dos Problemas de Programação Linear Inteira

Esta seção apresenta formulações em programação linear inteira (PLI) para os problemas em estudo. Primeiro, é apresentada a formulação em PLI para o problema RA-MaxC. Em seguida, mostra-se como otimizar a solução ótima para o problema AR-MaxC considerando a soma das distâncias entre clientes e servidores para encontrar a ótima solução com a menor soma de distâncias e a com a maior soma de distâncias. Desse modo será possível mostrar experimentalmente se a otimização considerando a soma de distâncias tem potencial para melhorar as soluções do problema AR-MaxC. Também são mostradas as formulações em PLI para o problema da p -mediana e uma otimização com respeito a máxima vértice-conectividade das soluções para o problema da p -mediana.

3.2.1 PLI para o problema AR-MaxC

O problema AR-MaxC pode ser definido 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. 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:

$$\min \sum_{i \in S} y_i \tag{3.5}$$

$$\kappa(j, i)x_{ij} = \kappa_2(j)x_{ij}, \quad \forall i \in S, \quad \forall j \in C \tag{3.6}$$

As demais restrições do problema são:

$$y_i \geq x_{ij}, \forall i \in S, \forall j \in C \quad (3.7)$$

$$\sum_{i \in S} x_{ij} = 1, \forall j \in C \quad (3.8)$$

$$y_i \in \{0, 1\}, \forall i \in S \quad (3.9)$$

$$x_{ij} \in \{0, 1\}, \forall i \in S, \forall j \in C \quad (3.10)$$

A restrição 3.7 assegura que há um servidor em i se o cliente em j está conectado ao servidor em i . A restrição 3.8 assegura que o cliente em j está conectado a apenas um servidor. As restrições 3.9 e 3.10 especificam o domínio das variáveis de decisão y_i e x_{ij} .

3.2.2 Seleção do Conjunto de Servidores Considerando Distâncias

Instâncias do problema proposto poderão ter mais de uma solução ótima. De acordo com o contexto, pode ser 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 distâncias entre os servidores e os clientes seja a menor possível e o número de caminhos vértice-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 a mesma do problema anterior, mais 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:

$$\min D = \sum_{i \in S, j \in C} d_{ij} x_{ij} \quad (3.11)$$

$$\sum_{i \in S} y_i = p \quad (3.12)$$

$$\kappa(j, i) x_{ij} = \kappa_2(j) x_{ij}, \forall i \in S, \forall j \in C \quad (3.13)$$

As demais restrições do problema são as mesmas equações 3.7 a 3.10 apresentadas anteriormente.

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

$$\max D = \sum_{i \in S, j \in C} d_{ij} x_{ij} \quad (3.14)$$

$$\sum_{i \in S} y_i = p \quad (3.15)$$

$$\kappa(j, i)x_{ij} = \kappa_2(j)x_{ij}, \quad \forall i \in S, \forall j \in C \quad (3.16)$$

As demais restrições do problema são novamente as equações 3.7 a 3.10.

3.2.3 Comparação com a p -Mediana

As somas das distâncias entre os clientes e seus servidores do problema de alocação de recursos com máxima conectividade ainda podem ser comparadas com a menor soma de distâncias com p servidores removendo-se a restrição de conectividade 3.6. O que corresponde ao problema da p -mediana, podendo ser formulado como [19]:

$$\min D = \sum_{i \in S, j \in C} d_{ij}x_{ij} \quad (3.17)$$

$$\sum_{i \in S} y_i = p \quad (3.18)$$

As demais restrições do problema são as equações 3.7 a 3.10.

Também pode ser 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 dada pelo problema da p -mediana, podemos calcular:

$$\min \text{somaDif} = \sum_{i \in S, j \in C} \kappa_2(j)x_{ij} - \kappa(j, i)x_{ij} \quad (3.19)$$

$$\sum_{i \in S, j \in C} d_{ij}x_{ij} = D \quad (3.20)$$

$$\sum_{i \in S} y_i = p \quad (3.21)$$

As demais restrições do problema são as equações 3.7 a 3.10.

O termo *somaDif* na equação 3.19 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 AR-MaxC. 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.

4 Implementação e Resultados Experimentais

Neste capítulo apresentamos a avaliação experimental que consiste em comparar as soluções dos problemas apresentados na seção 3.2 em grafos que representam backbones de redes de computadores. Para esses experimentos os clientes foram posicionados em todos os vértices e servidores podem ser atribuídos para qualquer vértice do grafo. Os grafos são conexos, contém de 60 a 200 vértices e têm ao menos um ciclo. Os grafos foram obtidos do site “The Internet Topology Zoo” (topology-zoo.org), um site dedicado à coleta de topologias de redes de todo o mundo [43].

Os algoritmos foram implementados em Python 3.6 usando o pacote “GNU Linear Programming Kit (GLPK) Simplex Optimizer”, versão 4.61. Os experimentos foram executados com um Intel(R) Core(TM) i7-5500U CPU com 4 núcleos de até 2.40 GHz, e 16GiB de DDR3L RAM de 1600 executando o Linux Ubuntu 16.04 com o kernel 4.15.0-24.

Os dados obtidos experimentalmente são apresentados da Tabela 4.1 e a Tabela 4.2 descreve as abreviaturas que nomeiam as colunas da Tabela 4.1. As três primeiras colunas mostram, respectivamente, o número de vértices, o número de arestas e o $\kappa_2(v)$ máximo dos grafos. Em geral os grafos são esparsos sendo que o maior $\kappa_2(v)$ é igual a 6, valor que ocorre em vértices das redes Uninett e Cogent.

O problema de alocação de recursos com a máxima conectividade foi resolvido e o número de servidores resultante encontra-se na Tabela 4.1 sob a coluna de título p . Esses são os números mínimos de servidores $S \in V$, tal que para todo $v \in V$, existe $s \in S$ tal que $\kappa(s, v) = \kappa_2(v)$. Observa-se que para esses grafos o número mínimo de servidores com relação aos número de vértices no grafo é de 2% e o número máximo é de 12% — sendo a média para todas as instâncias igual a 6%. O tempo em segundos para calcular o número de servidores encontra-se na Tabela 4.1 sob o título “t(s)”. Observa-se que todas as soluções podem ser calculadas em menos de 65 segundos.

Entres as soluções para o problema AR-MaxC que maximizam a conectividade dos clientes com os servidores espera-se que sejam encontradas soluções que não sejam ótimas com relação a outros critérios como o da soma de distâncias entre clientes e servidores, um critério comum para encontrar recursos em redes. O trabalho experimental reportado em seguida quantifica as soluções com relação ao *trade-off* entre a conectividade e a soma das distâncias.

As soluções do problema RA-MaxC foram otimizadas para encontrar a menor e maior soma possível das distâncias entres os clientes e seus servidores. Esses valores encontram-se na Tabela 4.1 sob as expressões “AR-MaxC min(\sum dist)” e “AR-MaxC max(\sum 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 na Tabela 4.1 sob a expressão p -med \sum dist.

Os resultados para a soma das distâncias do problema da p -mediana e para a menor e maior soma de distâncias para AR-MaxC estão ilustrados no gráfico de barras na Figura 4.1.

Pode-se observar que em geral a maior soma de distâncias (barras vermelhas) são muito maiores que a menor soma de distâncias (barras azuis). A menor soma de distâncias das soluções para problema AR-MaxC são mais similares com a soma ótima das distâncias dada pelo problema da p -mediana (barras amarelas). *Esses resultados mostram que após otimizar as soluções, é possível encontrar conjuntos de servidores que maximizam a conectividade dos clientes e obtêm uma soma pequena de distâncias entre clientes e servidores na maioria das redes analisadas.*

As taxas na Tabela 4.1 sob os títulos *taxa max* e *taxa min* correspondem às razões entre as somas de distâncias obtidas pela problema AR-MaxC e a soma ótima obtida pelo problema da p -mediana, isto é:

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

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

Comparando-se a maior taxa com a menor taxa, observa-se que ao otimizar as soluções do problema AR-MaxC com relação à soma das distâncias há um bom potencial para minimizar as distâncias. Em particular, observe que para vários grafos a diferença entre *taxa max* e *taxa min* é grande e que *taxa min* está próxima a 1 em várias das redes analisadas. Por exemplo, os seguintes pares (*taxa max*, *taxa min*) são encontrados na Tabela 4.1: (3.55, 1.07), (3.77, 1.16), (6.13, 1.22), (5.14, 1.24), (6.35, 1.20). Em média, *taxa min* é 1.20, o que mostra que para maximizar a vértice-conectividade, nas redes analisadas, a menor soma de distâncias entre clientes e servidores aumenta em média 20%; no pior caso, o aumento é de 66%. Em outros grafos, a otimização consegue somas bastante próximas ao limite inferior dado pelo problema da p -mediana.

Um outro experimento foi feito para ver se o número de servidores for maior que o mínimo necessário para atender a máxima conectividade dos clientes, então o valor da soma otimizada das distâncias com máxima conectividade será mais próximo do problema da p -mediana. Para tal, adicionou-se até 6 servidores ao número mínimo de servidores encontrado nas soluções do problema AR-MaxC e calculou-se as *taxa min*, sempre usando o mesmo número de servidores no problema da p -mediana. Os valores calculados para todos os grafos estão na Tabela 4.3. Pode-se ver que a média de todas as taxas decrescem a medida que mais servidores são adicionados. Com 1 servidor a mais, a média é reduzida de 1.20 para 1.12 e com mais 6 servidores para 1.04. O grafo na Figura 4.2 mostra as taxas para os grafos que apresentavam taxas acima de 1.10 com o número mínimo de servidores. Podemos ver nessa figura como a adição de servidores impacta as taxas entre a soma das distâncias das soluções para o problema AR-MaxC e o limite inferior dado pelo problema da p -mediana. Para todos esses grafos, as taxas decresceram para menos que 1.15 com a adição de 3 servidores e para menos que 1.10 com 6 servidores a mais.

O próximo experimento consiste em comparar a conectividade das soluções do problema AR-MaxC com as soluções dadas pelo problema da p -mediana. Em geral, soluções para o problema da p -mediana também não são únicas e foram, portanto, otimizadas para maximizar a vértice-conectividade de acordo com a formula 3.19 que minimiza *sumDif* entre todas as soluções com a soma ótima das distâncias.

Com respeito ao problema da p -mediana, investigamos os clientes para os quais a conectividade não é máxima, ou seja, $\text{diff} = \kappa_2(c) - \kappa(s, c) > 0$. O resultado se encontra na Tabela 4.1 sob o título *num clie com dif*. O maior valor corresponde a 20% dos vértices do grafo (rede Bestel).

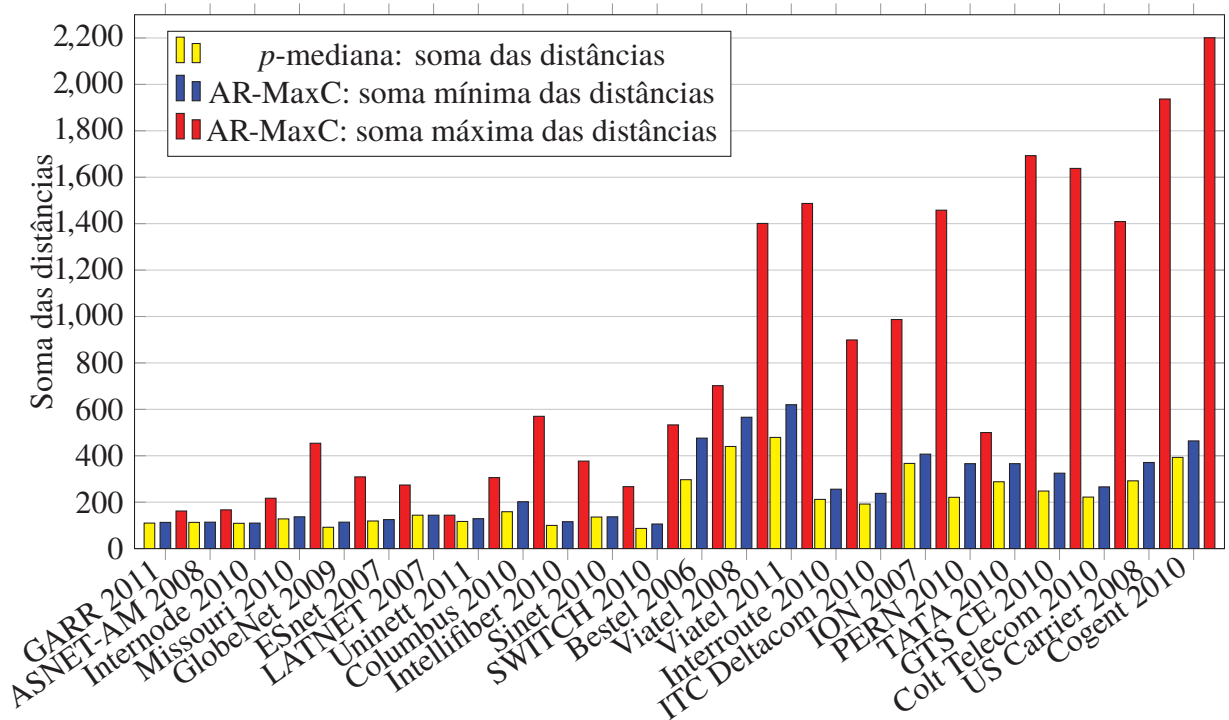


Figura 4.1: Soma das distâncias entre clientes e seus servidores para os casos: soma máxima e mínima para o problema de máxima conectividade e o problema da p -mediana.

O gráfico de barras empilhadas na Figura 4.3 mostra os vértices com os maiores $diff$. A parte azul das barras representam o número de caminhos vértice-disjuntos entre clientes e servidores. Como mostrado na coluna $max\ dif(c)$ da Tabela 4.1, a parte vermelha representa o número de caminhos vértice-disjuntos que o cliente não tem com o servidor, mas poderia ter se a restrição da máxima conectividade fosse usada. Como mostrada na coluna $\kappa_2(c)$ com $max\ dif$ da Tabela 4.1, a soma das partes azul e vermelha de uma barra corresponde ao $\kappa_2(c)$ do cliente. Podemos ver que para esses vértices os grafos têm de 25% a 60% menos caminhos vértice-disjuntos entre clientes e servidores que o valor ótimo, $\kappa_2(v)$.

Podemos também verificar a média das diferenças para os vértices com $diff > 0$, sob a expressão “dif médio” na Tabela 4.1, que corresponde à fórmula:

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

As soluções do problema da p -mediana mostram uma diferença média nos vértices com $diff > 0$ de 25% a 50%, e na média de todos os grafos de 36%.

Tabela 4.1: Resultados experimentais para o problema AR-MaxC e suas comparações.

Nome	$ V $	$ A $	$\kappa_2(v)$	p	$\frac{p}{ V }$	$t(s)$	AR-MaxC max (\sum dist)	AR-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	1.2	162	113	110	1.47	1.03	7	5	2	5	0.31
ASNET-AM 2008	65	79	4	2	0.03	1.3	167	114	113	1.48	1.01	2	1	2	4	0.50
Internode 2010	66	78	4	3	0.05	1.6	217	110	109	1.99	1.01	1	1	1	4	0.25
Missouri 2010	67	83	4	5	0.07	1.5	454	137	128	3.55	1.07	5	5	1	4	0.28
GlobeNet 2009	67	113	5	8	0.12	1.2	309	114	92	3.36	1.24	13	10	3	5	0.34
ESnet 2007	68	92	4	3	0.04	1.6	274	125	119	2.30	1.05	7	6	2	4	0.36
LATNET 2007	69	74	3	1	0.01	1.9	144	144	144	1	1	0	0	0	0	0
Uninett 2011	69	98	6	5	0.07	1.6	306	129	117	2.62	1.10	9	6	3	6	0.31
Columbus 2010	70	85	4	4	0.06	1.7	570	202	159	3.58	1.27	4	4	1	4	0.31
Intellifiber 2010	73	97	4	9	0.12	1.8	377	116	100	3.77	1.16	10	10	1	4	0.33
Sinet 2010	74	76	3	2	0.03	2.2	267	137	136	1.96	1.01	1	1	1	3	0.33
SWITCH 2010	74	92	4	9	0.12	2.1	533	106	87	6.13	1.22	2	2	1	3	0.33
Bestel 2006	84	101	3	3	0.04	3.4	702	476	297	2.36	1.60	18	17	2	3	0.48
Viatel 2008	88	92	3	3	0.03	4.2	1401	566	440	3.18	1.29	4	4	1	3	0.33
Viatel 2011	92	96	3	3	0.03	4.1	1487	620	479	3.10	1.29	3	3	1	3	0.33
Interroute 2010	110	158	4	7	0.06	7.7	899	256	212	4.24	1.21	18	16	2	4	0.37
ITC Deltacom 2010	113	183	5	8	0.07	7.7	987	238	192	5.14	1.24	6	6	1	4	0.31
ION 2007	125	150	3	5	0.04	13.2	1458	407	367	3.97	1.11	8	8	1	3	0.40
PERN 2010	127	129	3	2	0.02	14.8	500	366	221	2.26	1.66	7	5	2	3	0.50
TATA 2010	145	194	5	9	0.06	20.5	1693	366	288	5.88	1.27	13	13	1	5	0.32
GTS CE 2010	149	193	4	12	0.08	24.6	1638	325	248	6.60	1.31	14	14	1	4	0.31
Colt Telecom 2010	153	191	3	11	0.07	23.5	1409	266	222	6.35	1.20	18	17	2	3	0.39
US Carrier 2008	158	189	3	13	0.08	27.2	1937	371	292	6.63	1.27	16	15	2	3	0.43
Cogent 2010	197	245	6	13	0.07	64.7	2201	464	393	5.60	1.18	23	22	2	6	0.38

Tabela 4.2: Notação da tabela 4.1.

$ 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.
t (s)	Tempo em segundos para calcular p
AR-maxC max (\sum dist)	Maior soma de distâncias possível para o problema de máxima conectividade.
AR-maxC min (\sum dist)	Menor soma de distâncias possível para o problema de máxima conectividade.
p -med \sum 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 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 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.

Tabela 4.3: Razão entre a menor soma de distâncias com a máxima conectividade e a soma de distâncias do problema da p -mediana para acréscimos do número mínimo de servidores, p , de 1 até 6.

	p	$p + 1$	$p + 2$	$p + 3$	$p + 4$	$p + 5$	$p + 6$
GARR 2011	1.03	1.05	1.02	1.01	1.01	1.01	1.02
ASNET-AM 2008	1.01	1.01	1.01	1.01	1.01	1.01	1
Internode 2010	1.01	1.02	1.02	1.02	1.03	1.03	1.03
Missouri 2010	1.07	1.08	1.06	1.07	1.07	1.05	1.05
GlobeNet 2009	1.24	1.10	1.09	1.06	1.07	1.07	1.08
ESnet 2007	1.05	1.04	1.04	1.03	1.04	1.04	1.04
LATNET 2007	1	1	1	1	1	1	1
Uninett 2011	1.10	1.09	1.09	1.10	1.08	1.09	1.08
Columbus 2010	1.27	1.09	1.07	1.07	1.07	1.04	1.02
Intellifiber 2010	1.16	1.15	1.13	1.10	1.08	1.07	1.06
Sinet 2010	1.01	1.01	1.01	1.01	1.01	1	1
SWITCH 2010	1.22	1.02	1.03	1.03	1.03	1.03	1.05
Bestel 2006	1.60	1.17	1.13	1.08	1.03	1.02	1.02
Viatel 2008	1.29	1.18	1.11	1.04	1.04	1.04	1.03
Viatel 2011	1.29	1.23	1.10	1.03	1.04	1.06	1.05
Interoute 2010	1.21	1.16	1.13	1.10	1.10	1.10	1.08
ITC 2010	1.24	1.18	1.12	1.05	1.03	1.01	1.01
ION 2007	1.11	1.08	1.05	1.05	1.05	1.03	1.04
PERN 2010	1.66	1.36	1.16	1.10	1.12	1.11	1.10
TATA 2010	1.27	1.18	1.17	1.14	1.12	1.11	1.08
GTS CE 2010	1.31	1.18	1.13	1.09	1.08	1.07	1.07
Colt 2010	1.20	1.10	1.10	1.10	1.09	1.08	1.06
US Carrier 2008	1.27	1.19	1.12	1.10	1.08	1.06	1.06
Cogent 2010	1.18	1.11	1.08	1.06	1.04	1.03	1.02
média	1.20	1.12	1.08	1.06	1.05	1.05	1.04

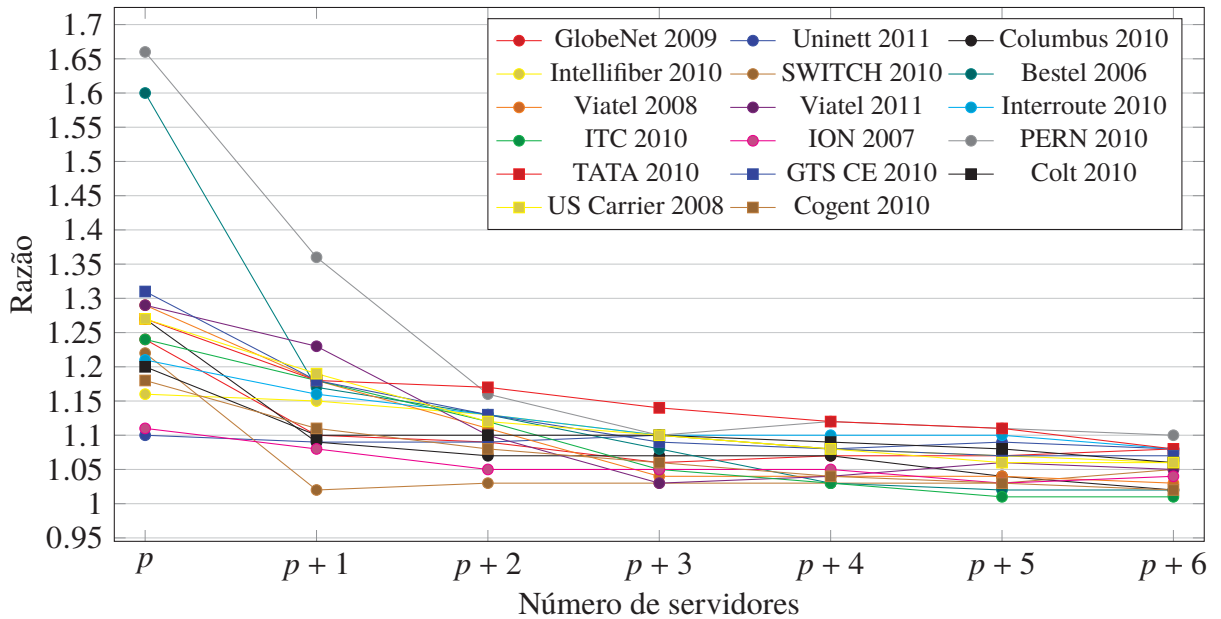


Figura 4.2: Razões entre as somas das distâncias mínimas entre clientes e servidores para o problema AR-MaxC e para o problema da p -mediana acrescentando mais servidores.

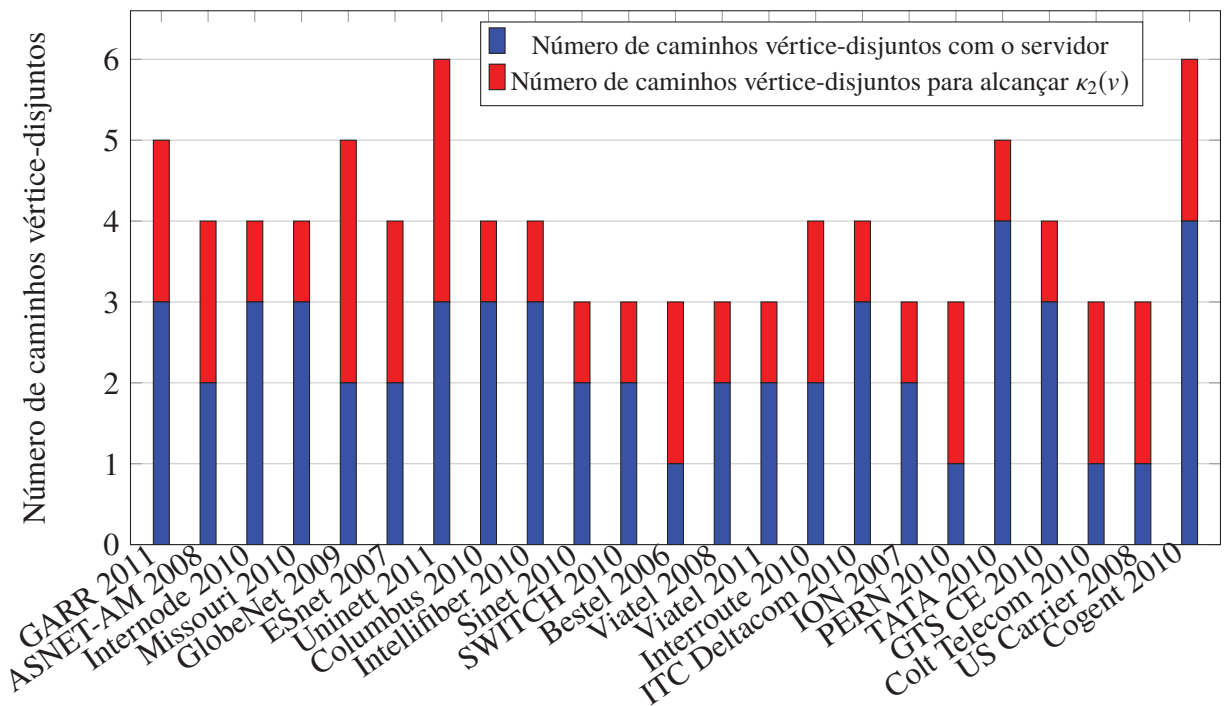


Figura 4.3: Número de caminhos vértice-disjuntos das soluções da p -mediana otimizada e a diferença para a máxima conectividade (para os vértices com a maior diferença).

5 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 a versão de decisão do 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 conexos com 60 a 200 vértices que representam backbones de redes de computadores.

Os resultados experimentais mostram que para esses grafos o número mínimo de servidores necessários é relativamente pequeno, em média 6% do número de nodos da rede. 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 ótimas. Por exemplo, no caso do grafo Us Carrier 2008, a soma das distâncias da pior solução para a melhor é 5 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. Quando mais servidores foram adicionados ao número mínimo, os resultados mostraram um rápido declínio da diferença entre as distâncias e o limite inferior dado pelo problema da p -mediana. 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. Em geral, os resultados mostraram que é possível encontrar conjuntos de servidores que maximizam a conectividade e obtêm uma soma pequena de distâncias entre clientes e servidores na maioria das redes estudadas.

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

Referências

- [1] Yuval Rochman, Hanoch Levy, and Eli Brosh. Resource placement and assignment in distributed network topologies. In *INFOCOM, 2013 Proceedings IEEE*, pages 1914–1922. IEEE, 2013.
- [2] Georgios Rodolakis, Stavroula Siachalou, and Leonidas Georgiadis. Replicated server placement with qos constraints. *IEEE Transactions on Parallel and Distributed Systems*, 17(10):1151–1162, 2006.
- [3] Bo Li, Mordecai J Golin, Giuseppe F Italiano, Xin Deng, and Kazem Sohraby. On the optimal placement of web proxies in the internet. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1282–1290. IEEE, 1999.
- [4] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker. On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1587–1596. IEEE, 2001.
- [5] Weixiong Rao, Lei Chen, Ada Wai-Chee Fu, and Guoren Wang. Optimal resource placement in structured peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(7):1011–1026, 2010.
- [6] Mohamed Younis and Kemal Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.
- [7] Myung M. Bae. Resource placement in torus-based networks. *IEEE Transactions on Computers*, 46(10):1083–1092, 1997.
- [8] Brandon Heller, Rob Sherwood, and Nick McKeown. The Controller Placement Problem. *ACM SIGCOMM Computer Communication Review*, 42(4):473, 2012.
- [9] Lucas F. Muller, Rodrigo R. Oliveira, Marcelo C. Luizelli, Luciano P. Gaspary, and Marinho P. Barcellos. Survivor: An Enhanced Controller Placement Strategy for Improving SDN Survivability. In *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pages 1909–1915, 2014.
- [10] Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press, 2008.
- [11] Karine Pires. Connectivity measures based on vertex cuts for complex networks. Master's thesis, Department of Informatics - Federal University of Paraná, UFPR, 2011.
- [12] Karine Pires, Jaime Cohen, and Elias P. Duarte Jr. Medidas de conectividade baseadas em cortes de vertices para redes complexas. In *12 Workshop de Testes e Tolerância a Falhas (WTF'2011), Anais do SBRC'2011*, 2011.

- [13] Kewen Liao. *Approximation algorithms for resource allocation optimization*. PhD thesis, School of Computer Science, The University of Adelaide., 2014.
- [14] Karen Aardal, Jaroslaw B. Byrka, and Mohammad Mahdian. *Facility Location*, pages 717–724. Springer, 2016.
- [15] Charles S. ReVelle and Horst A. Eiselt. Location analysis: A synthesis and survey. *European journal of operational research*, 165(1):1–19, 2005.
- [16] Jain Kamal and Vijay V. Vazirani. An approximation algorithm for the fault tolerant metric facility location problem. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 177–182. Springer, 2000.
- [17] Martine Labbé, Dominique Peeters, and Jacques-François Thisse. Location on networks. *Handbooks in operations research and management science*, 8:551–624, 1995.
- [18] Jagruti Sahoo, Mohammad Salahuddin, Roch Glitho, Halima Elbiaze, and Wessam Ajib. A Survey on Replica Server Placement Algorithms for Content Delivery Networks. *IEEE Communications Surveys & Tutorials*, 19(2):1002–1026, 2017.
- [19] Josh Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006.
- [20] Jakob Krarup and Steven Vajda. On Torricelli’s Geometrical Solution to a Problem of Fermat. *IMA Journal of Management Mathematics*, 8(3):215–224, 1997.
- [21] Zvi Drezner and Horst W. Hamacher. *Facility Location: Applications and Theory*. 2002.
- [22] Alfred Weber. *Ueber den Standort der Industrien*. J. C. B. Mohr, 1909.
- [23] S. Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.*, 12(3):450–459, June 1964.
- [24] S. Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Oper. Res.*, 13(3):462–475, June 1965.
- [25] Oded Kariv and S. Louis Hakimi. An Algorithmic Approach to Network Location Problems. II: The p-Medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.
- [26] Hiroshi Tamura, Masakazu Sengoku, and Shoji Shinoda. Location problems on undirected flow networks. *IEICE TRANSACTIONS (1976-1990)*, E73-E(12):1989–1993, 1990.
- [27] Hiroshi Tamura, Masakazu Sengoku, Shoji Shinoda, and Takeo Abe. Some Covering Problems in Location Theory on Flow Networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E75-A(6):678–684, 1992.
- [28] Hiro Ito, Motoyasu Ito, Yuichiro Itatsu, Kazuhiro Nakai, Hideyuki Uehara, and Mitsuo Yokoyama. Source Location Problems Considering Vertex-Connectivity and Edge-Connectivity Simultaneously. *Networks*, 40(2):63–70, 2002.
- [29] Kazuhisa Makino. Source Location Problems with Flow Requirements. *2012 Third International Conference on Networking and Computing*, pages 404–406, 2012.

- [30] Hiroshi Nagamochi, Toshimasa Ishii, and Hiro Ito. Minimum cost source location problem with vertex-connectivity requirements in digraphs. *Information Processing Letters*, 80(6):287–293, 2001.
- [31] Toshimasa Ishii, Hitoshi Fujita, and Hiroshi Nagamochi. The source location problem with local 3-vertex-connectivity requirements. *Discrete Applied Mathematics*, 155(18):2523–2538, 2007.
- [32] John G. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Visual Communications and Image Processing 2001*, volume 4310, pages 392–410. International Society for Optics and Photonics, 2000.
- [33] Junghee Han, David Watson, and Farnam Jahanian. An experimental study of internet path diversity. *IEEE Transactions on Dependable and Secure Computing*, 3(4):273–288, 2006.
- [34] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M Voelker. Characterizing and measuring path diversity of internet topologies. In *ACM SIGMETRICS Performance Evaluation Review*, volume 31, pages 304–305. ACM, 2003.
- [35] Justin P. Rohrer, Abdul Jabbar, and James P.G. Sterbenz. Path diversification: A multipath resilience mechanism. In *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on*, pages 343–351. IEEE, 2009.
- [36] Is-Haka M. Mkwawa, Demetres D. Kouvatsos, and Adrian Popescu. Overlay networks and graph theoretic concepts. In *Network performance engineering*, pages 1039–1054. Springer, 2011.
- [37] Junghee Han, David Watson, and Farnam Jahanian. Topology aware overlay networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2554–2565. IEEE, 2005.
- [38] Sabyasachi Roy, Himabindu Pucha, Zheng Zhang, Y Charlie Hu, and Lili Qiu. Overlay node placement: Analysis, algorithms and impact on applications. In *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*, pages 53–53. IEEE, 2007.
- [39] Meeyoung Cha, Sue Moon, Chong-Dae Park, and Aman Shaikh. Placing relay nodes for intra-domain path diversity. In *INFOCOM 2006: 25th IEEE International Conference on Computer Communications*. IEEE, 2006.
- [40] Vinh Bui, Weiping Zhu, and Lam Thu Bui. Optimal relay placement for maximizing path diversity in multipath overlay networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6. IEEE, 2008.
- [41] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 1972.
- [42] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [43] Simon Knight, Hung X Nguyen, Nick Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9):1765–1775, october 2011.