

# 1

## Apresentação do Problema

O objetivo deste capítulo é apresentar uma revisão bibliográfica do problema de síntese de redes a  $k$ -caminhos e de algumas de suas variantes.

Na Seção 1.1, o problema de síntese de redes a  $k$ -caminhos é apresentado. Na Seção 1.2, algumas de suas variantes e extensões são analisadas. Na Seção 1.3, são apresentados os objetivos e a organização desta tese.

### 1.1

#### O Problema de Síntese de Redes a $k$ -caminhos

Seja  $G = (V, E)$  um grafo não-orientado, onde  $V$  é o conjunto de nós e  $E$  é o conjunto de arestas. Um  $k$ -caminho entre os nós  $s$  e  $t \in V$  é um caminho de  $s$  a  $t$  formado por, no máximo,  $k$  arestas. Dados uma função de custos não-negativos  $w : E \rightarrow R_+$  associados às arestas de  $G$  e um conjunto de pares origem-destino,  $D$ , o problema de síntese de redes com  $k$ -caminhos ( $k$ PNDP) consiste em encontrar um subconjunto de arestas  $E' \subseteq E$  de custo mínimo que contenha um  $k$ -caminho entre as extremidades de cada par origem-destino pertencente a  $D$ .

Aplicações deste problema podem ser encontradas no projeto de redes de comunicação, em que caminhos com poucas arestas são desejáveis para garantir alta confiabilidade e pequenos atrasos.

Dahl e Johannessen [25] provaram que a versão de decisão de 2PNDP ( $k$ PNDP com  $k = 2$ ) é um problema NP-completo. Eles propuseram duas formulações por programação inteira e uma heurística gulosa, baseada na solução da relaxação linear do dual de uma das formulações. Também descreveram um algoritmo exato de planos de corte e apresentaram resultados computacionais para instâncias geradas aleatoriamente que continham, no máximo, 120 nós, 7140 arestas e 60 pares origem-destino.

### 1.1.1

#### Formulações de 2PNDP

Sejam  $s$  e  $t$  dois nós distintos de  $G = (V, E)$ . Define-se uma 2-estrela como sendo um subconjunto de  $E$  com a seguinte característica:

$$T(S_1, S_2) = \{(s, t)\} \cup \{(s, u) : u \in S_1\} \cup \{(v, t) : v \in S_2\}$$

onde  $S_1 \cup S_2 = V \setminus \{s, t\}$  e  $S_1 \cap S_2 = \emptyset$ .  $\Gamma_{st}$  é o conjunto de todas as 2-estrelas existentes entre  $s$  e  $t$ . A Figura 1.1 ilustra uma 2-estrela.

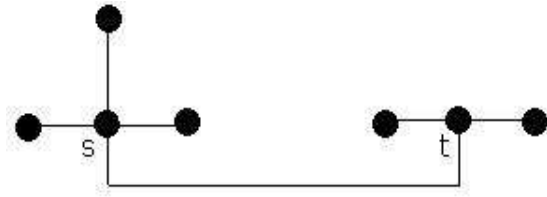


Figura 1.1: Exemplo de 2-estrela.

**Lema 1.1** *Seja  $(s, t)$  um par origem-destino de  $D$  formado por dois nós distintos de  $G$ . Seja  $F \subseteq E$  uma solução do 2PNDP e  $T$  uma 2-estrela para o par  $(s, t)$ ,  $T \in \Gamma_{st}$ .  $F$  contém um 2-caminho entre  $s$  e  $t$  se e somente se  $F \cap T \neq \emptyset$ ,  $\forall T \in \Gamma_{st}$  [25, 68].*

2PNDP pode ser formulado como um problema de programação linear inteira, usando-se o conceito de 2-estrela (modelo 2-estrela):

$$\text{minimizar } \sum_{e \in E} c_e x_e \quad (1.1)$$

sujeito a

$$\sum_{e \in T} x_e \geq 1, \forall T \in \Gamma \quad (1.2)$$

$$x_e \in \{0, 1\}, \forall e \in E \quad (1.3)$$

onde  $x_e = 1$  indica que a aresta  $e$  pertence a um 2-caminho,  $x_e = 0$  caso contrário.  $T$  é uma 2-estrela e  $\Gamma$  representa o conjunto de todas as 2-estrelas.

As restrições (1.2) indicam que, pelo menos, uma aresta de cada 2-estrela é selecionada. As restrições (1.3) garantem a integralidade da solução.

O número de variáveis desta formulação é  $O(n^2)$ , onde  $n = |V|$ . Quando o grafo  $G$  é completo, há  $2^{n-2}$  2-estrelas para cada par origem-destino. Logo, o número de restrições (1.2) é  $p2^{n-2}$ , onde  $p = |D|$ . O número de variáveis inteiras é  $n(n-1)/2$ . Portanto, o número de restrições do modelo 2-estrela cresce exponencialmente com  $n$ .

Outra formulação com um número menor de restrições foi apresentada em [25]. Seja  $P_d$  o conjunto de todos os 2-caminhos para o par origem-destino

$d \in D$ . Define-se a variável binária  $z_P = 1$  se  $P \in P_d$ ;  $z_P = 0$  caso contrário. Então, 2PNDP também pode ser formulado como:

$$\begin{aligned} & \text{minimizar } \sum_{e \in E} c_e x_e \\ & \text{sujeito a} \end{aligned}$$

$$\sum_{P \in P_d} z_P = 1, \forall d \in D \quad (1.4)$$

$$z_P \leq x_e, \forall e \in P, \forall P \in P_d, \forall d \in D \quad (1.5)$$

$$z_P \geq 0, \forall P \in P_d, \forall d \in D \quad (1.6)$$

$$x_e, z_P \in \{0, 1\}, \forall e \in P, \forall P \in P_d, \forall d \in D \quad (1.7)$$

A restrição (1.4) estabelece que um 2-caminho é selecionado para cada par origem-destino  $d \in D$ . Este modelo tem  $p(n-1)$  variáveis adicionais em relação ao modelo anterior. O número de restrições (1.4) é  $p$ . O número de restrições (1.5) é  $p(2n-3)$  porque, exceto para o 2-caminho que une diretamente um par origem-destino, todos os outros 2-caminhos são formados por duas arestas. O número de restrições (1.6) é  $p(n-1)$  porque, para cada par origem-destino, há  $(n-1)$  2-caminhos. Portanto, este modelo possui, somente,  $O(n^2)$  restrições.

### 1.1.2 Algoritmos

A Figura 1.2 ilustra o pseudo-código do algoritmo guloso de Dahl e Johannessen [25]. Este método resolve, aproximadamente, o dual da relaxação linear do modelo 2-estrela, e gera uma solução primal viável. Ele é uma extensão do algoritmo de Goemans e Williamson [43], como descrito a seguir.

Dado um par  $(s, t) \in D$ , o algoritmo mantém dois conjuntos de nós,  $S_s$  e  $S_t$ , com as seguintes características:

$$\begin{aligned} s & \in S_s, s \notin S_t \\ t & \in S_t, t \notin S_s \end{aligned}$$

Inicialmente,  $S_s = \{s\}$ ,  $S_t = \{t\}$ , onde  $(s, t) \in D$  (linhas 1 e 2).  $\bar{c}_e$  é o custo reduzido da aresta  $e \in E$ . No início do algoritmo,  $\bar{c} = c$  (linha 2). Este método gera duas 2-estrelas (linha 4) como sendo:

$$T_s = (S_s, V - S_s) \text{ e } T_t = (V - S_t, S_t)$$

O algoritmo calcula o número  $m_e$  de 2-estrelas que contêm a aresta  $e$  (linhas 5 a 7) e computa  $\epsilon^* = \max\{\epsilon : \epsilon \cdot m_e \leq \bar{c}_e, \forall e \in E\}$  (linha 8).

Após o cálculo de  $\epsilon^*$ , o algoritmo atualiza o custo reduzido de cada aresta para  $\bar{c}_e - \epsilon^* \cdot m_e$  (linhas 9 e 10). Se o custo reduzido de uma aresta

tornar-se nulo (linha 11), esta aresta pertencerá à solução gulosa (linha 12) e as suas extremidades deverão fazer parte dos conjuntos  $S_s$  e  $S_t$ , uma vez verificadas suas regras de construção. Se as extremidades da aresta não pertencem ao conjunto  $S_t$  e ao conjunto  $S_s$ , faça um nó pertencer ao conjunto  $S_s$  e o outro pertencer ao conjunto  $S_t$  (linhas 13 a 15). Se uma de suas extremidades pertence a  $S_t$ , faça a outra extremidade pertencer a  $S_s$  (linhas 16 a 18). Se uma de suas extremidades pertence a  $S_s$ , faça a outra extremidade pertencer a  $S_t$  (linhas 19 a 23).

Todos os pares origem-destino que tenham um 2-caminho na solução gulosa devem ser removidos do conjunto de demandas (linha 24). Se este conjunto estiver vazio, o algoritmo pára (linha 3). Senão, uma nova iteração do laço das linhas 4 a 25 é realizada.

Antes de retornar a solução gulosa construída (linha 27), o algoritmo remove desta solução todas as arestas que não são importantes para a manutenção da sua viabilidade (linha 26).

O laço das linhas 3 a 25 é executado  $p$  vezes, onde  $p = |D|$ . Os laços das linhas 5 a 7 e 9 a 23 são executados  $m$  vezes, onde  $m = |E|$ . Considerando-se que os demais passos do algoritmo são  $O(1)$ , a complexidade do algoritmo da Figura 1.2 é  $O(pm)$ .

Dahl e Johannessen [25] também desenvolveram um algoritmo de planos de corte para 2PNDP baseado no modelo caminho. Em cada iteração, o algoritmo resolve o problema de programação linear relaxado e adiciona alguns planos de corte ao problema, gerados por procedimentos polinomiais de separação. Se a solução do problema relaxado viola pelo menos uma das novas restrições impostas por estes planos de corte, uma nova iteração é realizada. Senão, o algoritmo pára.

### 1.1.3

#### Resultados Computacionais

A fim de testar os algoritmos guloso e de planos de corte, Dahl e Johannessen geraram duas classes distintas de instâncias: instâncias com pares origem-destino disjuntos e instâncias com pares origem-destino aleatórios. O grafo  $G$  é sempre completo e os custos das arestas foram gerados aleatoriamente de uma distribuição uniforme no intervalo  $[1,100]$ . O número de pares origem-destino é  $|V|/2$ . Os pares são escolhidos aleatoriamente do conjunto de arestas  $E$ . Estas instâncias contém, no máximo, 120 nós, 7140 arestas e 60 pares origem-destino.

```

procedimento GULOSO_DJ;
1  Seja  $(s, t)$  um par origem-destino de  $D$ ;
2   $S_s \leftarrow \{s\}$ ;  $S_t \leftarrow \{t\}$ ;  $\bar{c} \leftarrow c$ ;  $x \leftarrow \emptyset$ ;
3  enquanto  $|D| \neq 0$  faça
4      Gerar duas 2-estrelas  $T_s$  e  $T_t$ ;
5      para  $\forall e \in E$  faça
6          Calcular o numero  $m_e$  de 2-estrelas que contenham  $e$ ;
7      fim-para;
8       $\epsilon^* = \max\{\epsilon : \epsilon \cdot m_e \leq \bar{c}_e, \forall e \in E\}$ 
9      para  $\forall e \in E$  faça
10          $\bar{c}_e \leftarrow \bar{c}_e - \epsilon^* \cdot m_e$ ;
11         se  $\bar{c}_e = 0$  então
12              $x \leftarrow x \cup \{e\}$ ;
13             Sejam  $u$  e  $v$  as extremidades de  $e$ ;
14             se  $(u, v \notin S_s)$  e  $(u, v \notin S_t)$  então
15                  $S_s \leftarrow S_s \cup \{u\}$ ;  $S_t \leftarrow S_t \cup \{v\}$ ;
16             senão
17                 se  $v \in S_t$  então  $S_s \leftarrow S_s \cup \{u\}$ ;
18                 se  $u \in S_t$  então  $S_s \leftarrow S_s \cup \{v\}$ ;
19                 se  $v \in S_s$  então  $S_t \leftarrow S_t \cup \{u\}$ ;
20                 se  $u \in S_s$  então  $S_t \leftarrow S_t \cup \{v\}$ ;
21             fim-se;
22         fim-se;
23     fim-para;
24     Remover de  $D$  os pares origem-destino que tenham um 2-caminho
        em  $x$ ;
25 fim-enquanto;
26 Remover de  $x$  todas as arestas que não alterem a sua viabilidade;
27 retornar  $x$ ;
fim GULOSO_DJ;

```

Figura 1.2: Pseudo-código do algoritmo guloso [25].

Para o algoritmo de planos de corte, foi observado que quase todos os problemas teste foram resolvidos até a otimalidade. Observou-se que o algoritmo guloso é extremamente rápido, mas a qualidade da solução encontrada é inferior, quando comparada à solução obtida pelo algoritmo de planos de corte. Para as instâncias com pares disjuntos, observou-se um erro de 5% a 10% em relação ao valor da solução ótima, enquanto que, para as instâncias com pares aleatórios, este erro foi de 15% a 20%.

## 1.2

### Variantes e Extensões

Nesta seção são discutidas três variantes do problema de síntese de redes a  $k$ -caminhos, incluindo o problema de síntese de redes a  $k$ -caminhos

com restrições de fluxo, o problema da árvore geradora mínima com  $k$ -caminhos e o problema da árvore de Steiner com  $k$ -caminhos.

### 1.2.1

#### Problema de Síntese de Redes a $k$ -caminhos com Restrições de Fluxo

Dada uma função de capacidade  $\mu : E \rightarrow R_+$ , associada às arestas de  $G$ , o problema de síntese de redes a  $k$ -caminhos com restrições de fluxos ( $k$ PNDP-RF) consiste em encontrar um subconjunto de arestas  $E' \subseteq E$  com custo mínimo e as capacidades não-violadas, que contenha um  $k$ -caminho entre as extremidades de cada par origem-destino pertencente a  $D$ .

Balakrishnan e Altinkemer [11] formularam o  $k$ PNDP-RF como um problema de programação inteira. Eles propuseram uma maneira simples de modelar este problema usando uma formulação de multifluxo e derivaram uma relaxação lagrangeana cujo limite dual lagrangeano dominava o limite dado pela relaxação linear. Os resultados numéricos mostraram que este método gerava ótimos locais para instâncias que continham, no máximo, 30 nós e 420 arestas.

Este problema também foi investigado por LeBlanc et al [56]. Eles analisaram extensões do modelo de roteamento clássico, impondo restrições nos tamanhos dos caminhos entre pares origem-destino e nas capacidades das arestas. Seus modelos tinham como objetivo identificar quais caminhos poderiam ser usados entre os pares origem-destino, a fim de minimizar o tempo de espera de pacotes na rede. Eles usaram um algoritmo de desvio de fluxo para resolver o  $k$ PNDP-RF, apresentando resultados computacionais para instâncias que continham, no máximo, 106 nós e 11130 pares origem-destino.

### 1.2.2

#### Problema da Árvore Geradora Mínima com $k$ -caminhos

Seja  $G = (V, E)$  um grafo não-orientado, onde  $V = \{0, \dots, n\}$  é o conjunto de nós e  $E$  é o conjunto de arestas. Dado um nó raiz 0 e um conjunto  $X = V - \{0\}$ , o problema da árvore geradora mínima com  $k$ -caminhos ( $k$ MSTP) consiste em encontrar uma árvore geradora do conjunto  $\{0\} \cup X$  com custo mínimo, que contenha um  $k$ -caminho entre qualquer nó de  $X$  e a raiz 0.

Gouveia [45] mostrou que  $k$ MSTP é um problema NP-difícil e propôs modelos baseados em extensões das restrições de Miller, Tucker e Zem-

lim [64]. Ele apresentou alguns esquemas de cálculos de limites inferiores para este problema, baseados em relaxação lagrangeana e em otimização por subgradientes. Seus experimentos relataram resultados numéricos para instâncias com, no máximo, 40 nós.

Em [46], Gouveia propôs formulações multifluxos para este problema. Ele derivou uma relaxação lagrangeana cujo limite dual lagrangeano dominava o limite dado pela relaxação linear das formulações de [45]. Seus experimentos solucionavam as mesmas instâncias tratadas em [45] e indicavam que estes limites inferiores eram significativamente melhores que aqueles conhecidos.

O caso particular do problema da árvore geradora mínima com 2-caminhos (2MSTP) foi primeiramente estudado por Dahl [24]. Ele estudou este problema do ponto de vista poliedral e provou que o 2MSTP é NP-difícil. Além disto, apresentou uma descrição completa de um politopo inteiro para classes particulares de grafos.

### 1.2.3

#### Problema da Árvore de Steiner com $k$ -caminhos

Seja  $G = (V, E)$  um grafo não-orientado, onde  $V = \{0, \dots, n\}$  é o conjunto de nós e  $E$  é o conjunto de arestas. Dado um nó raiz 0 e um conjunto de nós terminais  $X \subseteq \{1, 2, \dots, n\}$ , o problema da árvore de Steiner com  $k$ -caminhos ( $k$ STP) consiste em encontrar uma árvore geradora do conjunto  $\{0\} \cup X$  com custo mínimo, possivelmente contendo nós não-terminais em  $V - X$ , que contenha um  $k$ -caminho entre qualquer nó de  $X$  e a raiz 0.

$k$ STP é NP-difícil porque sua versão não-restrita, o problema de Steiner em grafos, é NP-difícil [52]. Gouveia [47] discutiu alternativas para reforçar o limite dado pela programação linear do modelo de multifluxos para o  $k$ STP, tentando aplicar o estudo que foi desenvolvido em [46] para o  $k$ MSTP. Ele derivou uma relaxação lagrangeana da formulação multifluxos que melhorou o limite dado pela programação linear. Ele apresentou uma nova formulação cujo limite dado pela programação linear é igual ao limite dual lagrangeano, usando a técnica de redefinição de variáveis de Martin [58]. Além disto, Gouveia descreveu um algoritmo exato baseado na relaxação de sua nova formulação. Seus experimentos computacionais resolvem instâncias que contém, no máximo, 101 nós e 350 arestas.

Este problema também foi investigado por Voss [101]. Ele discutiu algumas variações de um algoritmo de busca tabu e apresentou limites

inferiores baseados em extensões e generalizações das restrições de Miller, Tucker e Zemlin [64].

Para construir soluções iniciais viáveis para o  $k$ STP, Voss desenvolveu uma heurística gulosa simples, usando alguns princípios aplicados em algoritmos conhecidos para o problema de árvore geradora mínima (como, por exemplo, o algoritmo de Prim [77]), juntamente com a computação de caminhos mais curtos. Voss propôs um algoritmo de busca local que procura substituir todos os caminhos chave [59, 99] da solução gulosa. A fim de superar a otimalidade local, desenvolveu uma heurística de busca tabu [38], que possui um gerenciamento dinâmico da lista tabu [26, 38]. Seus experimentos computacionais resolvem instâncias de, no máximo, 500 nós e 2500 arestas.

Em [104], o  $k$ STP foi usado para modelar projetos de redes centralizadas com restrições na qualidade de serviço. O nó raiz desempenhava o papel de um *site*, os nós terminais representavam os usuários e os nós não-terminais eram os pontos de chaveamento na rede. Limitações no número de caminhos entre o nó raiz e quaisquer nós terminais garantem a qualidade do serviço, respeitando restrições de desempenho relacionadas à eficácia (isto é, à fração de tempo que o *site* pode se comunicar com um usuário, quando todos os caminhos de roteamento estejam funcionando), à confiabilidade (isto é, a probabilidade de que uma sessão entre um usuário e o *site* não seja interrompida, devido a uma falha de conexão) e ao atraso da rede (isto é, o tempo médio de uma mensagem ser roteada do *site* a um usuário e vice-versa).

Woolston e Albin [104] descreveram uma heurística de duas fases para projeto de redes, baseada no algoritmo de Frank et al. [37]. A primeira fase é responsável por construir uma solução gulosa viável, que pode ser uma árvore geradora mínima ou uma estrela, de acordo com os seus parâmetros de entrada. Vizinhanças desta solução são investigadas até que um ótimo local seja encontrado pela última fase. A solução gulosa é construída pela inserção de caminhos mais curtos ligando a raiz aos nós terminais, desde que não violem as restrições do número de caminhos. O algoritmo de construção pára quando todos os nós terminais estiverem na solução. O algoritmo de busca local examina as variações no projeto inicial da rede. Este procedimento seleciona um nó terminal e inclui a aresta adjacente a este nó que tenha o menor custo e que não esteja presente na solução atual, causando a formação de um ciclo. O algoritmo procura excluir as outras arestas, tentando diminuir o custo da solução e remover o ciclo, sempre testando a viabilidade das modificações efetuadas. A busca local



pára quando todos os nós da rede já tiverem sido visitados. Os experimentos realizados com problemas com, no máximo, 25 nós, indicam que restrições no número de caminhos melhoram consideravelmente a qualidade de serviço em redes centralizadas.

### 1.3

#### **Objetivos e Organização da Tese**

O objetivo desta tese é desenvolver novas heurísticas para 2PNDP, analisar variantes e combinações de GRASP e reconexão por caminhos e comparar os resultados obtidos pelos algoritmos desenvolvidos com os descritos na literatura.

No Capítulo 2 a metaheurística GRASP é discutida. Além de explicar cada etapa do método, este capítulo descreve melhoramentos propostos para o algoritmo básico, tais como os mecanismos alternativos usados na fase construtiva, a reconexão por caminhos e os algoritmos híbridos. Neste capítulo é proposta uma heurística GRASP para 2PNDP e são discutidas todas suas fases.

No Capítulo 3 são analisados os resultados computacionais da heurística GRASP combinada com reconexão por caminhos para 2PNDP. Este capítulo descreve as instâncias geradas para realizar os diversos testes. São comparadas diferentes versões de GRASP e reconexão por caminhos, a fim de verificar como uma componente de memória age sobre as soluções encontradas pela heurística GRASP. A melhor destas versões é comparada com o algoritmo guloso da literatura. Também considera-se a adição de uma componente de memória ao procedimento de busca local.

No Capítulo 4 apresentam-se estratégias paralelas de implementação da heurística GRASP seqüencial combinada com reconexão por caminhos. Descreve-se cada uma das variantes paralelas desenvolvidas neste trabalho. Além disto, estas variantes são comparadas, a fim de verificar-se qual o papel que a troca de informações entre os processadores participantes do ambiente paralelo desempenha no encontro das melhores soluções e nos tempos de processamento.

No Capítulo 5 realiza-se uma análise das contribuições deste trabalho e propõem-se extensões e pesquisas futuras.