MAC 5701 - Tópicos em Ciência da Computação
Relatório de Estudos
Problemas de Conexão em Grafos

Aluno: *Fabrício Vertamatti* (favert@ime.usp.br)
Orientador: *Carlos Eduardo Ferreira* (cef@ime.usp.br)

Breve resumo: No decorrer do curso estudei diversos problemas de conexão em grafos. Os artigos estudados tratam de problemas clássicos de conexidade em grafos: problema de caminhos disjuntos, problema da árvore de Steiner e o problema de fluxo multicomodity. Nesta monografia apresento uma categorização dos problemas estudados bem como uma inter-relação entre eles, por exemplo através do fato de muitos se assemelharem com casos particulares de outros. Nosso interesse por estes problemas surgiu ao estudarmos os problemas de entrega e conexão ponto-a-ponto nos artigos [5] e [6]. Nesta monografia estudamos outros problemas interessantes de conexidade, buscando que os métodos aplicados nestes problemas possam se aplicar aos problemas de conexão ponto-a-ponto.

Introdução: O problema de Steiner é base para uma série de problemas de conexão na teoria dos grafos. Ele pode ser formulado da seguinte forma:

Problema 0.1 (Problema da Árvore de Steiner Mínima) *Seja $G = (V, E)$ um grafo conexo e $c : E \rightarrow \mathbb{Q}_+$ custos positivos nas arestas de G. Seja $V_1 \subseteq V$ um conjunto de vértices terminais. O problema é encontrar um subgrafo conexo de G que contenha todos os vértices terminais. Como a função de custo é positiva o grafo será acíclico.*

O problema da árvore de Steiner pertence à classe de problemas $\mathcal{NP}$-difícil, o que implica na dificuldade de resolver todos os problemas dele derivados, como a maioria dos que foram objetos deste estudo. Veremos alguns problemas que são casos particulares do Problema da Árvore de Steiner [4] na seção 1, por exemplo [2] e[3]. Na seção 2, veremos dois problemas relacionados ao problema de Fluxo mínimo de Ford & Fulkerson [7],[1]. Por fim, na seção 3, apresento os problemas de conexão e entrega ponto a ponto e a sua relação com os demais problemas apresentados.

Sumário

1 Problemas relacionados ao problema da Árvore de Steiner

1.1 Problema de Steiner em grafos [4]

Problema 1.1 *Seja $G = (V, E)$ um grafo conexo com pesos nas arestas. Seja $V_1 \subseteq V$ e $G(V_1)$=$\{g_i \subseteq G | g_i$ é conexo com respeito a $V_1\}$. Encontre, dentre todos os subgrafos em $G(V_1)$, o subgrafo com menor peso; tal subgrafo é denotado por $g^*(V_1)$ e é chamado Subgrafo de Steiner de G com respeito a V_1. Caso $g^*(V_1)$ é um grafo acíclico, $g^*(V_1)$ é chamado Árvore de Steiner de G com respeito a V_1.*

Este é o Problema de Steiner em Grafos. Foi atacado com sucesso pela primeira vez por Melzak e mais tarde generalizado por Gilbert e Pollack. Primeiramente vamos estudar dois casos especiais do problema. Feito isso, discutiremos a relação entre o problema de Steiner e alguns problemas de cobertura em grafos. Para encerrar apresentaremos um algoritmo para o problema dada uma certa topologia. Na última seção uma versão para grafos orientados do problema é apresentada.

Casos especiais do problema de Steiner em Grafos
No primeiro caso vamos estudar o problema quando a cardinalidade de V_1, denotada por $|V_1|$ é igual a 2. Se todas as arestas do grafo $G = (V, E)$ possuem pesos não-negativos, o problema 1 se reduz o problema de encontrar um caminho mínimo em um grafo, que foi resolvido por Ford, Moore, Bellman, Floyd e mais efetivamente por Dantzig. Este último forneceu um algoritmo $O(log_2(n-1)\sqrt{2})$, onde $|V| = n$. Caso algumas arestas de $G = (V, E)$ possuem pesos negativos, procedemos da seguinte forma: Seja b_k uma aresta de G e $b_k = (v_i, v_j)$. Seja B_k o conjunto de todas as arestas de G incidentes em ambos v_i e v_j. Estas arestas são chamadas de arestas em *paralelo* com (v_i, v_j). *Reduzir* uma aresta em $G = (V, E)$ significa encontrar um grafo $G(b_k) = [V - \{v_i\}, E - B_k] = [V(b_k), E(b_k)]$ a partir de G tal que, se $b_t \in E - B_k$ incide em $v_r \neq v_i$ em G, então b_t é incidente em v_r em $G(b_k)$; e se b_t

incide em v_i em G, então b_t incide em v_j em $G(b_k)$. Seja $G = (V, E)$ um grafo com peso nas arestas e $g \subseteq G$. Por $G_g(V, E)$ denotamos o grafo obtido de G setando todas as arestas de g para 0 em G_g e não alterando as arestas remanescentes em G_g. A seguinte asserção fornece uma solução para o problema 1 quando $|V_1| = 2$.

Asserção 1.1 *Seja $G = (V, E)$ um grafo conexo com pesos nas arestas e $V_1 = (v_r, v_s)$. Seja $g \subseteq G$ o subgrafo de G contendo todas as arestas de pesos negativos em G. Seja $G_g = (V, E)$. Se $p_g(v_r, v_s)$ é o menor caminho em G_g entre v_r e v_s, e $p(v_r, v_s)$ é o correspondente subgrafo em G, então o subgrafo de Steiner $g^*(V_1) = g \cup p(v_r, v_s)$. Vale ressaltar que o subgrafo $g^*(V_1)$, no geral, não é um caminho de v_r a v_s em G, mas contém tal caminho. O problema de encontrar um menor caminho em um grafo é um importante problema em aberto.*

O segundo caso que vamos considerar ocorre quando $V_1 = V$. Se todas as arestas do grafo $G = (V, E)$ possuem pesos não-negativos, o problema 1 se reduz o problema de encontrar uma árvore geradora mínima. Este problema foi primeiramente considerado por Kruskal e mais eficientemente resolvido por Prim. Uma versão do algoritmo de Prim e uma prova de sua eficácia será apresentada mais adiante. A seguinte asserção fornece uma solução para o problema 1 quando $|V_1| = V$.

Asserção 1.2 *Seja $G = (V, E)$ um grafo conexo com pesos nas arestas. Seja $g \subseteq G$ um subgrafo contendo todas as arestas de pesos negativos em G. Seja $G_g = (V, E)$. Se t_g é a árvore geradora mínima de G_g e t a sua correspondente subgrafo em G, então $g^*(V) = g \cup t$. Vale ressaltar que o subgrafo $g^*(V)$ da asserção 2 no geral, não é necessariamente uma árvore geradora de G. Uma versão do algoritmo de Prim fornece uma árvore geradora mínima de um grafo conexo com pesos nas arestas G, mesmo que algumas arestas de G possuam pesos negativos.*

Problemas de cobertura e o Problema de Steiner

Aqui vamos definir diversos problemas de cobertura em grafos e mostrar sua relação com o problema de Steiner (problema 1).

(2- Problema da Cobertura): Seja $G = [X, Y; E]$ um grafo bipartido. Encontrar um subconjunto de cardinalidade mínima $X_0 \subseteq X$ tal que para todo $y \in Y$, y é adjacente a um vértice em X_0. Tal conjunto é chamado *mínima cobertura* de Y em G.

(3- Cobertura de Vértices): Seja $G = [V, E]$. Encontrar um subconjunto mínimo $V_0 \subseteq V$ tal que para todo $v \in V - V_0$, v é adjacente a um vértice em V_0. O conjunto V_0 é chamado *cobertura mínima de vértices* de G. Um problema parecido neste caso é o da Cobertura de Arestas, o qual é um pouco mais simples devido a sua relação com o problema de emparelhamentos em grafos.

(4- Conjunto Independente Máximo): Seja $G = [V, E]$. Encontrar um subconjunto de cardinalidade máxima $V_0 \subseteq V$ tal que nenhum par de vértices distintos em V_0 são adjacentes. Este conjunto é chamado *conjunto independente máximo* de G.

(5- Clique Máximo): Seja $G = [V, E]$. Encontrar um subconjunto de cardinalidade máxima $V_0 \subseteq V$ tal que se v_i e $v_j \in V_0, (v_i \neq v_j)$, então $(v_i, v_j) \in E$. O conjunto V_0 é chamado *clique máximo* de G.

Sabe-se que se podemos resolver o problema 2, então podemos resolver os problemas 3, 4 e 5; e que a solução do problema 4 nos dá uma solução para o problema 5 e vice-versa. Além disso, um resultado importante é o fato de que, resolvendo o problema 1, podemos resolver o problema 2 e assim o 3, 4 e 5 também.

Problema de Steiner para grafos orientados Existe uma maneira simples de definir o problema para grafos orientados:

Seja $G = [V, E]$ um grafo orientado e assuma que $Y \subseteq V$. Definimos um conjunto de subgrafos $G(Y)$ de G como segue: $G(Y) = \{g \subset G |$ se um caminho de y_i para y_j existe em G, então o caminho também existe em g para todo $y_i, y_j \in Y\}$. Encontrar o subgrafo em $G(Y)$ de menor peso.

Mais uma vez, no caso $|Y| = 2$ o problema se reduz ao problema de encontrar um caminho mínimo em um grafo orientado, já resolvido caso G seja acíclico. Caso $Y = V$, o problema também se reduz a um problema solucionável caso G não tenha circuitos.

Existe um algoritmo de aproximação para o problema de Steiner com razão $\frac{5}{3}$ como pode ser visto em [8]. Este artigo também foi objeto de estudo para esta monografia. Este algoritmo é baseado no problema de árvores geradoras mínimas em hipergrafos.

1.2 Problema da árvore retilínea de Steiner [2]

Uma árvore de Steiner retilínea de um conjunto A de pontos no plano é uma árvore que conecta A usando linhas verticais e horizontais de menor tamanho possível. A seguir será mostrado que o problema para encontrar tal árvore é NP-completo, daí a ênfase na literatura sobre o problema em algoritmos para casos especiais e heurísticas.

Problema 1.2 *Seja A um conjunto finito de pontos no plano orientado. Uma árvore retilínea de Steiner (RST - Rectilinear Steiner Tree) de A é uma árovore composta unicamente por segmentos de linha horizontais e verticais, que interconcectam os pontos de A. Uma RST ótima de A é uma na qual as linhas usadas possuem o menor tamanho possível. Vale lembrar que ao contrário do que ocorre com uma árvore geradora, em uma RST pode existir 3 ou mais linhas se encontrando num ponto que não pertence a A, chamado de* Ponto de Steiner. *De fato, frequentemente uma RST de A contém pelo menos um ponto de Steiner. O "problema da árvore retilínea de Steiner (RST)" é então, dado A, encontrar uma RST ótima de A.*

Existe uma forte evidência da impossibilidade encontrar um algoritmo eficiente para o problema, que pode ser provada mostrando que ele pertence à classe NP-completo.

A principal relação do problema RST com a classe NP-completa por enquanto é o problema de Steiner em grafos. Imagina-se que a restrição geométrica do RST poderia torná-lo um problema melhor tratável, mas neste artigo mostraremos que isso não é verdade.

Para provar que RST pertence à classe NP-completo, vamos provar inicialmente uma sequência de resultados auxiliares.

Dado um grafo $G = (V, E)$, uma cobertura de vértices de G é qualquer subconjunto $V^* \subseteq V$ que contém pelo menos uma extremidade de cada aresta de G. Começamos com o seguinte problema que pertence à classe NP-completo:

Cobertura de vértices de grafos planares. Dado um grafo planar $G = (V, E)$ e um número inteiro k, existe uma cobertura de vértices V^* de G satisfazendo $|V^*| \leq k$?

Vamos transformar a *Cobertura de vértices de grafos planares* no seguinte problema um pouco mais restrito:

Cobertura de vértices de grafos planares com grau máximo 3. Dado um grafo planar $G = (V, E)$ com nenhum vértice com grau maior que 3, e um inteiro k, existe uma cobertura de vértices V^* de G satisfazendo $|V^*| \leq k$?

A seguir transformamos este problema em outro restrito problema de cobertura de vértices:

Cobertura conexa de vértices em grafos planares com grau máximo 4. Dado um grafo planar $G = (V, E)$ com nenhum vértice com grau maior que 4, e um inteiro k, existe uma cobertura de vértices V^* de G satisfazendo $|V^*| \leq k$ e tal que o subgrafo de G induzido por V^* é conexo?

Este último problema é transformado no RST, enunciado como segue:

Árvore retilínea de Steiner. Dado um conjunto finito A de coordenadas inteiras de pontos no plano, e um inteiro l, existe uma RST de A com tamanho total menor ou igual a l?

Lema 1.1 *Cobertura de vértices de grafos planares com grau máximo 3 é NP-completo.*

Prova. Dado um grafo planar e um inteiro k, construímos um grafo planar G' com nenhum vértice com grau maior que 3 e um inteiro k' tal que G' tem uma cobertura de vértices de tamanho k' se e somente se, G possui uma cobertura de tamanho k. Seja $G = (V, E)$ onde $V = \{v_1, v_2, ..., v_n\}$. A construção começa com uma representação fixa $G = G_0$. Para cada $i, i = 1, 2, ..., n$ construimos uma representação planar do grafo G_i a partir do grafo G_{i-1} como segue:

(i) Seja $\{v_i, w_1\}, \{v_i, w_2\}, ..., \{v_i, w_p\}$ as arestas saindo de v_i na ordem em que ocorrem em torno de v_i na representação planar de G_{i-1}.

(ii) Substitua v_i por um ciclo consistindo dos novos vértices $u_i(j), v_i(j), 1 \leq j \leq n$ e as novas arestas $\{(u_i(j), v_i(j)\}, 1 \leq j \leq n, \{v_i(j), u_i(j+1)\}, 1 \leq j \leq n-1$, e $\{v_i(n), u_i(1)\}$.

(iii) Substitua cada aresta $\{v_i, w_j\}$ pela aresta $\{v_i(j), w_j)\}$, inclua um novo vértice z_i e adicione a aresta $\{u_i(1), z_i\}$. Finalmente defina $G' = G_n$ e

$k' = n^2 + k$. Agora suponha que V^* é uma cobertura de vértice de G com $|V^*| \leq k$. Então existe uma cobertura de vértices V_1^* de G' tal que $|V_1^*| \leq k'$, tal que

$V_1^* = \{v_i(j) : v_i \in V^*, 1 \leq j \leq n\} \cup \{u_i(1) : v_i \in V^*\} \cup \{u_(j) : v_i \ni V^*, 1 \leq j \leq n\}$.

Da mesma forma, suponha V_1^* é uma cobertura de G' que satisfaz $|V_1^*| \leq k'$. Uma vez que somente os vértices de G' que cobrem arestas correspondentes a arestas de G são os $v_i(j)$, temos que o conjunto

$V^* = \{v_i$: para algum j,$1 \leq j \leq n, v_i(j) \in V_1^*\}$

forma uma cobertura de G. Devemos mostrar que $|V^*| \leq k$. Defina, para $1 \leq i \leq n, S_i = V_1^* \cap \{u_i(j), v{+}i(j) : 1 \leq j \leq n\}$. Para cobrir todos as $2n$ arestas no ciclo para v_i devemos ter $|S_i| \geq n$. Como $k' = n^2 + k$, isso implica que no máximo k valores de i podem satisfazer $|S_i| > n$. Como $u_i(1) \in S_i$, o único conjunto de exatamente n vértices que cobre todas as $2n$ arestas no ciclo para v_i é $\{u_i(j) : 1 \leq j \leq n\}$. Logo se existe um j para o qual $v_i(j) \in S_i$, precisamos ter $|S_i| > n$. Como isso ocorre para no máximo k valores de i, temos $|V^*| \leq k$ e V^* é a cobertura desejada de G. Como G' pode ser construído em tempo polinomial no tamanho de G, o problema restrito é NP-completo. c.q.d.

Lema 1.2 *Cobertura conexa de vértices em grafos planares com grau máximo 4 é NP-completo.*

Prova. Dado o grafo planar G com nenhum vértice de grau maior que 3 e um inteiro k, construimos um grafo planar G' com nenhum vértice de grau maior que 4 e um inteiro k' tal que G possui uma cobertura de tamanho k se e somente se G' possui uma cobertura de tamanho k'.

Seja $G = (V, E)$ onde $V = \{v_1, ..., v_n\}$. A construção é feita da seguinte forma:

(i) . Substitua cada aresta $\{v_i, v_j\}$ de E por três arestas:

$\{v_i, x_i(j)\}$, $\{x_i(j), x_j(i)\}$, $\{x_j(i), v_j\}$ onde $x_i(j)$ e $x_j(i)$ são novos vértices.

(ii) Considere cada região R da representação planar do grafo resultante do passo (i). Seja W o conjunto de todos aqueles vértices na vizinhança de R com grau menor que 4. Para cada w em W, introduza dois novos vértices w', w'' no interior de R e adicione as arestas $\{w, w'\}, \{w', w''\}$.

(iii) Para cada região, una todos os w' em cada região em um ciclo tal que o grafo continue planar. Seja $G' = (V', E')$ o grafo resultante. Seja r o número total de vértices w' introduzidos no passo (ii). Então $|V'| = n + 2|E| + 2r$. Defina $k' = k + |E| + r$.

O grafo G' e o inteiro k' podem claramente ser construídos em tempo polinomial. Para completar a prova, devemos mostrar que G' possui a cobertura se G possui.

Primeiro suponha que $V*$ é uma cobertura de G com $|V^*| \leq k$. Defina o conjunto $V_1^* \subseteq V'$ como sendo:

$V_1^* = V^* \cup \{\text{todos os vértices } w'\} \cup \{x_i(j) \in V' : v_i \in V^* oui < j\}$.

Claramente as arestas $\{w, w'\}, \{w', w''\}$ são cobertas. Agora considere uma aresta $\{v_i, v_j\} \in E$. Pela construção temos

$\{v_i, v_j\} \cap V_1^* = \{v_i, v_j\} \cap V^* \neq \emptyset$.

Se $\{v_i, v_j\} \cap V^* = \{v_j\}$, então v_j e $x_i(j)$ cobrem as arestas $\{v_i, x_i(j)\}$, $\{x_i(j), x_j(i)\}$, $\{x_j(i), v_j\}$. Se $\{v_i, v_j\} \cap V^* = \{v_i, v_j\}$ e $i < j$, então v_j, v_i e $x_i(j)$ cobrem tais arestas e V_1^* é uma cobertura de G'. Além disso, como exatamente um dos pares $\{x_i(j), x_j(i)\} \in V_1^*$ para cada $\{v_i, v_j\} \in E$, temos $|V_1^*| \leq k + r + |E| = k'$. Agora suponha que V_1^*, $|V_1^*| \leq k$ é uma cobertura de G'. Como todo w' pertence à cobertura, w'' não pertence a mesma. Podemos assumir também que um de cada par $\{x_i(j), x_j(i)\}$ pertence a V_1^*. Com essas afirmações, temos:

$|V_1^* \cap V| \leq k' - r - |E| = k$.

Falta mostrar que $V^* = V_1^* \cap V$ forma a cobertura desejada de G. Considere qualquer aresta $\{v_i, v_j\} \in E$. SPG suponha que $x_j(i)$ é um único membro de $\{x_i(j), x_j(i)\}$ que pertence a V_1^*. Então para cobrir a aresta $\{x_i(j), v_i\}$ temos de ter $v_i \in V_1^*$ e consequentemente $v_j \in V_1^* \cap V$. Logo $V_1^* \cap V$ contém pelo menos uma extremidade de cada aresta em E. c.q.d.

Teorema 1. RST é NP-completo. *Prova.* Dado um grafo planar G com nenhum vértice com grau maior que 4 e um inteiro k, construímos um conjunto A de pontos no plano orientado e um inteiro l tal que G tem uma cobertura de tamanho k se e somente se existe uma RST para A de tamanho l ou menos.

Seja $G = (V, E)$ onde $V = \{v_1, ..., v_n\}$ e $E = \{e_1, ..., e_m\}$. Considere uma grade de quadrados impostas no plano orientado consistindo de todas os segmentos de linhas da forma $[(6in^2, 6jn^2), (6(i-1)n^2, 6jn^2)]$ ou $[(6in^2, 6jn^2), (6(1)n^2, 6(j-1)n^2)]$, com i, j inteiros. A construção começa obtendo uma representação planar de G que usa somente segmentos de linha verticais ou horizontais. Escolhidos na grade acima. Cada vértice de G é mapeado a um ponto da forma $(6in^2, 6jn^2)$ para alguns i, j inteiros. Para a construção vamos usar o fato de nenhum vértice ter grau maior que 4 pois nenhum ponto da grade possui mais de 4 linhas incidentes.

Para cada $v_i \in V$, seja p_i o ponto correspondente na grade. Considere cada p_i e delete-o juntamente da porção de segmentos que chegam em p_i que distam até 2 unidade de p_i. Seja L o tamanho total dos segmentos de linhas remanescentes. Agora substitua cada segmento de linha pelo conjunto de todos os pontos daquele segmento que possuem coordenadas inteiras. Estes pontos formam o conjunto A e definimos $l = L + 2m + 2(k-1)$. Suponha que G tem uma cobertura conexa de tamanho menor que k. A correspondente RST de A contém todos os segmentos de linha unindo pares de pontos de A que estão exatamente distantes de uma unidade. Cada componente conectada da estrutura corresponde a uma aretsa de G e chamamos essa compontente para $e_s \in E$ de e_s-componente. Além disso, para cada $v_i \in V^*$ selecionamos alguns dos segmentos de linhas incidentes em p_i que foram deletados. Para isso primeiramente escolhemos uma árvore geradora do subgrafo de G induzido por V^*. Para cada aresta $e_s = \{v_i, vj\} \in E$ que não está na árvore, selecionamos uma extremidade v_i que pertence a $e_s \cap V^*$ e selecionamos o segmento de linha de tamanho 2 unindo p_I a e_s-componente. Pra cada aresta $e_s = \{v_i, v_j\}$ que está na árvore, selecionamos ambos segmentos de linha de tamanho 2, unindo e_s-componente a p_i e a p_j. Como V^* é uma cobertura conexa de G, a coleção resultante de segmentos formam uma RST para A. Para cada aresta na árvore adicionamos 2 segmentos e para cada uma que não está na árvore adicionamos 1. Logo o tamanho da RST é $L + 2m + 2(k-1)$ pois $|V^*| = k$.

Agora precisamos provar o contrário. Se existe uma RST de A com tamanho l ou menos, então existe uma cobertura conexa para G com tamanho k. Para isso, precisamos considerar uma RST como é construída na primeira parte da prova. Seja T uma RST construída desta forma, ótima para A, e que maximiza o número de segmentos de uma unidade com ambos pontos pertencendo a A. Os únicos segmentos dessa forma possíveis são aqueles deletados que podemos chamar de *segmento de aresta*. Assumimos que T possui todos os segmentos de arestas (é fácil ver por uma prova por contradição que tal T existe). Usando a terminologia anterior, podemos dizer que T é composto de "componentes de arestas" para cada aresta de G, com alguns segmentos de linhas adicionais unindo estas componentes de arestas. Estes segmentos adicionais chamaremos de *segmentos suplementares*. Vamos mostrar agora que esses segmentos suplementares são oriundos de um conjunto bem restrito. Qualquer segmento suplementar em T precisa ser parte de algum caminho composto inteiramente de segmentos suplementares que unem duas componentes de arestas. Para cada p_i definimos a *região ativa* de p_i como sendo o conjunto de pontos atingíveis por p_i por uma caminho retilíneo de tamanho menor que $3n^2$. Considere algum segmento suplementar que não está contido em nenhuma região ativa. Por nossa construção de A e a definiçnao de região ativa, qualquer caminho contendo o segmento que une duas componentes de arestas precisam conter pelo menos $3n^2 > 2m + 2(k_1)$ segmentos suplementares. Mas sabemos que T não contém esse número de segmentos suplementares, e assim deduzimos que todos os segmentos suplementares estão em regiões ativas. Segmentos suplementares dentro de uma determinada região ativa só podem servir para unir duas componentes de arestas representando arestas que encontram-se em um vértive v_i em G. Considerando todos os casos podemos ver que todas essas conexões na região ativa de p_i podem ser feitas com mínimo tamanho de segmentos de linha possível simplesmente unindo cada componente de aresta envolvida na conexão com o ponto p_i com um segmento de linha de tamanho 2. Assim podemos assumir que T consiste de todos as componentes de arestas, somados com vários segmentos de linha de tamanho 2 unindo certas componentes de arestas a certos pontos p_i, a qual é feita da mesma forma que a RST construida anteriormente.

Agora precisamos mostrar que podemos determinar uma cobertura conexa de G a partir dessa RST. Para isso assumimos que

$V^* = \{v_i \in V :$ alguma componente de aresta está unida a $p_i\}$

forma a nossa cobertura. Uma vez que toda componente de aresta está conectada a p_i para algum $v_i \in V^*$ e desde que uma componente de aresta pode somente estar unida a p_i se a aresta correspondente de G tem v_i como extremidade, V^* é portanto uma cobertura. Como $|T| < l$, pode existir no máximo $k-1$ componentes de arestas unidas a dois pontos p_i. Se deletarmos de T estas componentes que estão unidas a apenas um ponto p_i, a estrutura que resulta é conexa e possui no máximo $k-1$ componentes de arestas. Assim, deletando em G as correspondentes arestas, o subgrafo resultante terá no máximo $k-1$ arestas, será conexo e conterá exatamente os vértices de V^*, e, portanto, será uma cobertura conexa com $|V^*| \leq k$.c.q.d.

Concluindo, vimos então que o problema da árvore retilínea de Steiner é NP-completo.

1.3 Empacotamento de árvores de Steiner [3]

Nesta seção é apresentado um algoritmo de planos de corte para o problema de empacotamento de árvores de Steiner. Este algoritmo é usado para resolver problemas de roteamento switchbox de design de circuitos VLSI. Além disso, uma breve discussão da aplicação do algoritmo é apresentada, bem como uma heurística primal baseada num problema linear e detalhes de sua implementação. Este problema parece ser o que possui mais aplicação prática existente atualmente dentre os problemas estudados.

O problema de empacotamento de árvores de Steiner, pode ser definido como segue:

Problema 1.3 *Dado um grafo não direcionado $G = (V, E)$ com capacidades $c_e \in \mathbb{N}$ para todo $e \in E$ e uma coleção de conjuntos $\mathcal{N} = \{T_1, \ldots, T_N\}, N \in \mathbb{N}$, encontrar árvores de Steiner S_k para $T_k, k = 1, \ldots, N$, tais que cada aresta $e \in E$ está contida em no máximo c_e dos conjuntos de arestas $S_i, \ldots, S_N$. Toda coleção de árvores de Steiner*

$S_1, \ldots, S_N$ *com essa propriedade é chamado empacotamento de árvores de Steiner. Se as arestas possuem pesos, o problema passa a ser encontrar um empacotamento de custo mínimo.*

A principal motivação de estudar este problema vem do design de circuitos eletrônicos. A primeira fase (*design lógico*) consiste em especificar quais operações básicas são combinadas com unidades lógicas (chamadas *células*) e como estas células precisam se conectar através de trilhas. Os pontos onde as células são conectadas pelas trilhas são chamados *terminais* e o conjunto de terminais a serem conectados, são chamados de *rede*. A segunda fase (*design físico*) recebe os dados da fase anterior e consiste em distribuir as células numa área dada e conectar (roteamento) as redes via trilhas. O problema é usualmente decomposto em *problema da distribuição* e *problema do roteamento*. O que o artigo se interessa é o segundo, que pode ser enunciado da seguinte forma:

> Dada uma área (tipicamente um retângulo ocupado pelas células) e uma lista de redes, o problema de roteamento consiste em conectar os terminais de cada rede através de trilhas de modo que certas restrições técnicas são satisfeitas e uma determinada função objetivo é minimizada.

Um possível valor para a função objetivo pode ser, por exemplo, o tamanho total das trilhas ou a área resultante. O roteamento tem lugar nas chamadas *camadas*. Cada camada é dividida em *faixas* onde as trilhas correm. As faixas e as *vias*, pontos onde as trilhas mudam de camadas, precisam respeitar certos limites de distância.

O problema de roteamento é muito complexo para ser solucionado em um só passo. Na prática divide-se o problema em dois subproblemas. Num primeiro momento é determinado como as trilhas "se dispões" pelas células (*roteamento global*). Feito isso, as trilhas se distribuem às camadas e faixas de acordo com o que foi definido no fase de roteamento global (*roteamento detalhado*).

Um exemplo que pode ser mencionado é uma variante do problema de roteamento detalhado, chamado *problema de roteamento switchbox*. Aqui um grafo grade retangular completo é dado e seus conjuntos de terminais se localizam todos nos quarto lados da grade.

O problema de roteamento detalhado pode ser dividido em alguns modelos:

- **Modelo multicamada.** Dada um grafo tipo grade com k camadas (k grades sobrepostas com seus respectivos nós conectados por linhas perpendiculares), as redes devem ser conectadas de modo disjunto nos nós.
- **Modelo Manhattan.** Dado (subgrago de) uma grade retangular completa, as redes precisam ser interconectadas de modo disjunto nas arestas com a restrição de que se as redes se encontram em algum ponto, elas não podem "dobrar" neste ponto. Isso garante que o roteamento resultante pode ser obtido em duas camadas.
- **Modelo knock-knee.** Este modelo difere-se do anterior pois as dobras (*knock-knees*) são permitidas. Geralmente o tamanho total das trilhas deste roteamento é menor do que o obtido no modelo Manhattan.

Uma nova notação do problema que é usada no decorrer desta seção é esta:

Seja $G = (V, E)$ um grafo e $T \subseteq V$ um conjunto de nós de G. Um conjunto de arestas S é chamado de árvore de Steiner para T se o subgrafo $(V(S), S)$ contém caminhos de s para t para todos os pares de nós $s, t \in T, s \neq t$.

Problema 1.4 *(Problema de empacotamento de árvores de Steiner com pesos)*
Instância: Um grafo $G = (V, E)$ com capacidades inteiras positivas $c_e \in \mathbb{N}$ e pesos não negativos $w_e \in \mathbb{R}_+, e \in E$. Uma coleção de conjuntos de nós $\mathcal{N} = \{T_1, \ldots, T_N\}, N \geq 1$, com $T_k \subseteq V$ para todo $k = 1, \ldots, N$.
Problema: Encontrar conjuntos de arestas $S_1, \ldots, S_N \subseteq E$ tais que:
(i) S_k é uma árvore de Steiner em G para T_k para todo $k = 1, \ldots, N$,
(ii) $\sum_{k=1}^{N} |S_k \cap \{e\}| \leq c_e$ para todo $e \in E$,
(iii) $\sum_{k=1}^{N} \sum_{e \in S_k} w_e$ é minimo

Se o requerimento (iii) é omitido, chama-se o problema de empacotamento de árvores de Steiner. A coleção $\mathcal{N}$ é chamada de lista de rede. Qualquer elemento $T_k \in \mathcal{N}$ é chamado *conjunto de terminais* e os nós $t \in T_k$ são chamados *terminais*. Ao invés de conjunto terminal T_k, será dito simplesmente rede k. Uma instância do problema será referida por $(G, \mathcal{N}, c, w)$ ou $(G, \mathcal{N}, c)$, dependendo da existência de pesos nas arestas. $\mathbb{R}^{\mathcal{N} \times E}$ denota o espaço vetorial de dimensão $N.|E|$, $\mathbb{R}^E \times \ldots \times \mathbb{R}^E$, onde os componentes de cada vetor $x \in \mathbb{R}^{\mathcal{N} \times E}$ são indexados por x_e^k para $k \in (1, \ldots, N), e \in E$. O poliedro de empacotamento de árvores de Steiner $\mathrm{STP}(G, \mathcal{N}, c) \subset \mathbb{R}^{\mathcal{N} \times E}$ é o fecho convexo de todos os vetores de incidência dos empacotamentos de árvores de Steiner.

O problema de empacotamento de árvores de Steiner, bem como sua variante com pesos são respectivamente $\mathcal{NP}$-completo e $\mathcal{NP}$-difícil. O problema permanece difícil mesmo no caso de problemas de roteamento switchbox usando modelo knock-knee.

Problema de separação para diversas classes de inequações

Nesta seção é tratado brevemente algumas idéias de como separar classes de inequações apresentadas em um artigo sobre o mesmo problema de empacotamento, mas com uma investigação poliédrica. Neste resumo vou apenas citar alguns conceitos principais sobre o problema de separação.

O *problema de separação* para uma dada classe de inequações pode ser enunciado da seguinte forma:

Dada uma instância $(G, \mathcal{N}, c)$ do problema de empacotamento de Steiner, um vetor $y \in \mathbb{R}^{\mathcal{N} \times E}, y \geq 0$, e uma classe de inequações válidas para $\mathrm{STP}(G, \mathcal{N}, c)$; decidir se y satisfaz as inequações da classe dada e, caso negativo, encontrar uma inequação da classe violada por y.

Separação das inequações de partição de Steiner

Seja G um grafo e $T \subseteq V$ um conjunto de terminais. Chamamos uma partição $V_i, \ldots, V_p, p \geq 2$ de V uma *partição de Steiner com respeito a* T se $V_i \cap T \neq \emptyset$ para $i = 1, \ldots, p$. A inequação

$x(\delta(V_1, \ldots, V_p)) \geq p - 1$

é chamada de *inequação de partição de Steiner* e é válida para o poliedro $\mathrm{STP}(G, \mathcal{N}, \mathbb{1})$. Apesar do problema de separação correspondente ser

$\mathcal{NP}$-completo, existem casos especiais em que uma solução polinomial pode ser obtida. Um desses casos é quando o grafo G é planar e o conjunto de vértices terminais T está situado na borda externa de G. Este é o caso prático que inclui o problema de roteamento switchbox. O algoritmo para resolver este problema funciona da seguinte forma: assuma que G é 2-conexo. Logo o conjunto que engloba a face externa de G é um ciclo. Numere os elementos do conjunto terminal T de forma horária na face externa no ciclo. Agora considere o grafo dual $G^* = (V^*, E)$ de G e subdivida o nó representando a face externa em z nós $d_1, \ldots, d_z$ tais que toda aresta pertencente a um caminho de t_i para t_{i+1} em G na face externa seja agora incidente em d_{i+1} para $i = 1, \ldots, z$. Seja $G_D = (V_D, E)$ o grafo resultante e seja $D = \{d_1, \ldots, d_z\}$. Isto nos leva sob algumas considerações, que cada conjunto $S = \delta(V_1, \ldots, V_i)$, induzido por uma partição de Steiner $V_1, \ldots, V_i$ tem uma correspondência um pra um com uma árvore de Steiner, minimal nas arestas, em G_D com respeito a algum subconsjunto $J \subseteq D$. Esta equivalência proporciona o fato que o problema de separar as classes de inequações de partições de Steiner se reduz ao problema de encontrar um subconjunto J de D e uma árvore de Steiner em G_D minimal quanto as arestas com respeito a J. Dado J pode-se obter uma árvore ótima de Steiner em G_D com respeito a J utilizando programação dinâmica. Logo o ponto crucial é descobrir o conjunto $J \subseteq D$.

A seguir estuda-se a separação em inequações de partição de Steiner e inequações de ciclos alternantes; e o problema de encontrar um corte crítico, onde corte crítico pode ser definido dessa forma:

Um corte induzido por um conjunto W de nós é crítico se $s(W) := c(\delta(W)) - |S(W)| \geq 1$, onde $S(W) := \{k \in \{1, \ldots, N\} | T_k \cap W \neq \emptyset, T_k \cap (V \setminus W) \neq \emptyset\}$.

Baseado nas definições, o pode-se definir um algoritmo para encontrar cortes mínimos.

Implementação de algoritmo de planos de corte

Nesta seção é discutido características de um algoritmo de planos de corte para problema de empacotamento de árvores de Steiner (com ou sem pesos). Mais adiante, é introduzida uma heurística primal para o problema de roteamento switchbox, e alguns detalhes de implementação indispensáveis para solucionar tal problema são discutidos.

Heurística primal
A idéia da heurística é fazer uso da informação dada na solução atual da fase de plano de corte. Um algoritmo sequencial desenvolvido, considera cada conjunto de terminais como um componente isolado. Interativamente conecta-se dois componentes de uma rede de acordo com uma sequência determinada a priori. Esta sequência é determinada pela solução y atual de um programa linear. Logo em seguida o algoritmo é apresentado.

Detalhes de implementação
Neste tópico é abordado alguns "truques" que são introduzidos no algoritmo de planos de corte. Algumas idéias apresentadas são indispensáveis para a solução de problemas práticos. Uma delas consiste em "perturbar" o programa linear. Outro trata de gerar uma classe particular de corte de Steiner e inequações de partição de Steiner após resolver o programa linear inicial, para o qual há razões heurísticas para acreditar que formam um conjunto "bom" de cortes planos iniciais.

Feitas estas considerações, o algorimo principal para o problema de roteamento switchbox é apresentado (vide artigo).

2 Problemas de fluxo Multicomodity

2.1 Problema de fluxo multicomodity em grafos planares [7]

Nesta seção apresento um problema baseado no problema de fluxo em grafos definido por Ford e Fulkerson. A formulação do problema é a seguinte:

Problema 2.1 *Seja G um grafo e (s_i, t_i) $(1 \leq i \leq k)$ pares de vértices; cada aresta possui uma capacidade dada por um número real (≥ 0). Suponha também que $q_i \geq 0 (1 \leq i \leq k)$ são números reais que representam a demanda. Encontrar um fluxo para cada i entre s_i e t_i de valor q_i, tal que o fluxo total em cada aresta não excede sua capacidade.*

Este problema foi resolvido por Ford e Fulkerson quando $k = 1$ e por Hu quando $k = 2$. Neste artigo é apresentada uma solução para o caso geral quando G é planar e pode ser desenhado tal que $s_1, \dots, s_k, t_1, \dots, t_k$ estão todos na borda da face infinita.

Seja $G = (V, E)$ um grafo (não-direcionado e finito).

Seja $(s_1, t_1), \dots, (s_k, t_k)$ pares não necessariamente distintos de vértices de G. Suponha que cada aresta $e \in E$ possui uma capacidade (valor real) $w(e) \geq 0$ e que q_i $(1 \leq i \leq k)$ são valores reais que representam as demandas. Quando a seguinte sentença é verdadeira?

Teorema 2.1 *Para $1 \leq i \leq k$ existe um fluxo F_i de s_i para t_i de valor q_i tal que para cada aresta $e \in E$,*

$$\sum_i |F_i(e)| \leq w(e).$$

Este problema é chamado *multicommodity flow.*

Seja um conjunto $X \subseteq V$, seja $\partial(X) \subseteq E$ o conjunto de arestas com uma extremidade em X e a outra em $V - X$. Seja $D(X) \subseteq \{1, \dots k\}$ sendo

$$\{i : 1 \leq i \leq k, \{s_i, t_i\} \cap X \neq \emptyset \neq \{s_i, t_i\} \cap (V - X)\}.$$

Está claro que se 3.1 é verdadeiro, então a seguinte condição de conectividade vale.

Teorema 2.2 *Para cada $X \subseteq V$,*

$$\sum_{e \in \partial(X)} w(e) \geq \sum_{i \in D(X)} q_i$$

Ford e Fulkerson provaram que 3.2 é necessário e suficiente para 3.1 para $k = 1$; e Hu provou o mesmo para $k = 2$. O principal teorema do artigo segue abaixo:

Teorema 2.3 *Se G é planar e pode ser desenhado no plano de tal forma que $s_1, \ldots s_k, t_i, \ldots t_k$ estão todos na borda da região infinita, então 3.1 é equivalente a 3.2.*

Para provar o teorema, é suficiente lidar com o caso em que w e q são inteiros. Dessa forma pode-se provar um importante resultado; que sob hipótese 2.3, se 3.2 vale e w e q são inteiros, então não somente 3.1 vale, como também se w e q são pares, então o fluxo F_i em 3.1 é inteiro.

Além disso, pode-se provar este último resultado sob a fraca hipótese que w e q são inteiros e para cada $X \subseteq V$,

$$\sum_{e\in\partial(X)} w(e) - \sum_{i\in D(X)} q_i$$

$$(\geq 0)$$

é par. E para provar isto basta considerar o caso em que w e q são iguais a 1, adicionando e removendo apropriadamente arestas paralelas em G.

Para a prova de 2.3 é preciso provar o seguinte:

Teorema 2.4 *Suponha que $G = (V, E)$ é planar e pode ser desenhado no plano de tal forma que $s_1, \ldots s_k, t_i, \ldots t_k$ estão todos na borda da região infinita. E suponha que para cada $X \subseteq V$,$|\partial(X)| - |D(X)|$ é par e não negativo. Então existem caminhos disjuntos nas arestas $P_1, \ldots P_k$ de G, tal que P_i tem extremidades $s_i, t_i (1 \leq i \leq k)$.*

Um problema em aberto é achar uma generalização de 2.4.

Prova do Teorema

Para a prova, o artigo enuncia os seguintes lemas:

Teorema 2.5 *Suponha que $s_1, \dots s_k, t_i, \dots t_k$ são vértices do grafo $G = (V, E)$. Então: (i)$|\partial(X)| - |D(X)|$ é par para todo $X \subseteq V$ se e somente se é ímpar para todo $X \subseteq V$ com $|X| = 1$.*
(ii) se G é conexo, $|\partial(X)| - |D(X)| \geq 0$ para todo $X \subseteq V$ se e somente se é ≥ 0 para todo $X \subseteq V$ com $G|X$ e $G|(V - X)$ ambos conexos. [$G|X$ denota o resultado de deletar todos os vértices de $(V - X)$ e suas incidentes arestas.

Teorema 2.6 *Se $X, X' \subseteq V$, então*

$$|\partial(X \cap X')| + |\partial(X \cup X')| = |\partial(X)| + |\partial(X')| - 2|\partial(X - X') \cap \partial(X' - X)|.$$

A prova do teorema principal vem a seguir, usando indução em $|E|$, assumindo que G é conexo e mais ainda, 2-conexo. Assim a borda da região infinita consiste de um circuito C. Dizemos que $X \subseteq V$ é crítico se $|\partial(X)| = |D(X)|$ e $\partial(X)$ contém apenas duas arestas do circuito C.

Com esta propriedade, segue a prova baseada em 2.4, 2.5 e 2.6; e nos dois seguintes lemas:

Lema 2.1 *Se $\emptyset \subset X \subset V$ e $|\partial(X)| = |D(X)|$, e ambos $G|X$ e $G|(V - X)$ são conexos, então X é crítico.*

Lema 2.2 *Para alguma aresta $e \in E(C)$ com extremidades a, b e algum $j(1 \leq j \leq k)$, não existe $X \subseteq V$ crítico com $s_j, t_j \in X$ e com $X \cup \{a, b\} \neq \emptyset$.*

Suponha agora que, no problema anterior, todas as demandas e capacidades são iguais a 1. Suponha ainda que o grafo em questão seja uma grade. Este caso particular é representado pelo problema de caminhos disjuntos em grades retilíneas [1], que veremos no tópico seguinte.

2.2 Caminhos Disjuntos em Grades Retilíneas [1]

Problema 2.2 *Dada um grafo não orientado tipo grade, encontrar k caminhos disjuntos entre k pares de vértices previamente especificados.*

O artigo que foi estudado sobre o problema, fornece uma boa caracterização e um bom algoritmo para o caso especial do problema Integral Multicommodity Flow quando o grafo é definido como uma grade retangular. Este problema está diretamente relacionado à questão da construção de placas de circuitos.

Um dos tópicos centrais na teoria dos grafos consiste na descoberta de caminhos disjuntos respeitando diversas restrições. Aqui será apresentado o seguinte problema neste campo: Dado um grafo não orientado, encontrar k caminhos disjuntos entre k pares de vértices especificados de antemão. Este problema, também chamado de *Problema dos caminhos disjuntos* é uma especialização do problema Multicommodity flow, que pertence à classe NP-completo mesmo se $k = 2$. Existe uma boa caracterização para o problema dos caminhos disjuntos com $k = 2$ fornecida por Seymour, mas a questão está aberta para $k \geq 3$. No entanto, o seguinte caso relacionado com a impressão de placas de circuito, pode ser bem caracterizado. A prova fornece um bom algoritmo.

Em uma grade retangular, é dado um retângulo T (limitado por linhas traçadas) e k pares de pontos traçados distintos na borda (terminais). T define um subgrafo finito G com $n.m$ vértices obtidos quando as n linhas verticais e as m linhas horizontais se intersectam em T. O propósito é fornecer uma resposta neste caso especial.

No primeiro tópico adiante, é considerado um caso ainda mais específico quando um membro de cada par está na borda superior de T e outro está na borda inferior. Este caso também é conhecido como caso bipartido. O caso não bipartido, onde os terminais podem estar em qualquer lugar da borda de T, será analisado no tópico seguinte.

Por uma *coluna* de T define-se a região entre duas linhas traçadas consecutivas. Uma *linha* defini-se da mesma forma. O *congestionamento do caminho* (ou *congestionamento*) de uma coluna c é o número de pares terminais separados por c. Chamamos um ponto traçado de *exposto* se ele não é um terminal. Uma condição óbvia para a solução é o *critério de coluna*: o congestionamento de qualquer coluna é no máximo m, o número de linhas.

Caso bipartido
Infelizmente o critério de coluna não é suficiente no geral. Mas há uma pequena restrição que torna o critério da coluna suficiente, a saber:

Teorema 2.7 *Se pelo menos um ponto de canto de T não é um terminal, o critério de coluna é necessário e suficiente para a existência de caminhos disjuntos entre os correpondentes terminais.*

A prova da suficiência é apresentada no artigo. A prova é feita por indução, e sugere um algoritmo e sua prova de corretude (neste resumo vou apenas apresentar a idéia principal). Suponha que os quatro cantos de T sejam $A = (1, m)$, $B = (n, m)$, $C = (n, 1)$ e $D = (1, 1)$. Para cada par de terminais, seja $U(i)$ e $L(i)$ as x-cooredenadas do membro superior e inferior do par respectivamente (vamos adotar que a x-coordenada de um ponto P é representada por mesmo termo P).

Assumimos que m é igual ao máximo congestinamento Δ e fazemos indução em m. A base com $m = 1$ é trivial. Assuma que o ponto A (canto superior esquerdo) é exposto. O Algoritmo cnsiste de m fases. Em cada uma, o congestionamento e o número de linhas horizontais é diminuído em uma unidade. Apenas a primeira fase é analisada. Dela resulta uma sequência $U'(i)(i = 1, 2, ..., k)$ de pontos que dizem onde o caminho i deve ir na linha superior. Deve-se construir a sequência de forma que a hipótese de indução valha para o retângulo T' constituído de uma linha horizontal a menos, ou seja, o congestionamento máximo deve ser $m-1$ e deve existir um canto exposto na primeira linha. O procedimento possui a característica de alternar os cantos superiores entre exposto e não exposto. Um caminho é chamado *caminho-esquerdo* (*caminho-direito*) se $U(i) > L(i)$ $(U(i) < L(i))$. Sejam X e Y pontos do segmento AB com $X < Y$. O passo básico do algoritmo é o seguinte: "left-pushing" em XY. Assuma que X é exposto. Um *left-pushing elementar* informa a um caminho-esquerdo onde ir no segmento XY, decidindo se existe um caminho-esquerdo i para o qual $X < U(i) \leq Y$ e, caso afirmativo, escolha um, digamos j, para o qual $L(j)$ é o menor possível (se tal caminho não existe, o left-pushing elementar é chamado de *vanishing*). Defina $U'(j) = X$ se $L(i) \leq X$ e $U'(j)$=max (Z:$X \leq Z \leq L(j)$,(Z, m) é exposto) se $L(j) > X$. Um *right-pusinhg elementar* é definido analogamente. Um *left-pushing* em XY informa a diversos caminhos-esquerdos onde ir no segmento XY. Primeiro aplique um left-pushing elementar em XY. Assuma que o caminho j foi movido para a esquerda. Então o ponto $X' = U(j)$ na linha superior se tornou exposto. Aplique agora um left-pushing elementar em $X'Y$ e repita este procedimento até que o atual left-pushing seja vanishing. A primeira fase do algoritmo consiste em duas partes. Primeiro, aplique left-pushing em todo segmento AB. Na segunda parte, considere cada segmento maximal XY de AB que não foi coberto por nenhum caminho na primeira parte e defina Y'=max$(Z : X \leq Z < Y, (Z, m)$ é exposto). Para cada XY' aplique right-pushing em XY'. A descrição do algoritmo está então completa. Definimos $U'(j) = U(j)$ para cada caminho j que não tenha aparecido no left-pushing ou right-pushing na primeira fase. A prova de corretude do algoritmo será omitida conforme dito anteriormente.

Pode-se observar que uma linha vertical (exceto as bordas) pode ser retirada caso nenhum terminal esteja sobre ela, uma vez que o critério da coluna continua válido. Dessa forma, podemos assumir que $m \leq k$ e $n \leq 2k$. Portanto a complexidade do algoritmo depende apenas de k e não de m ou n.

O teorema 2.7 não fornece uma resposta para o caso onde nenhum dos pontos dos cantos são expostos. o teorema 2.8 a seguir resolve este problema. Para incorporar o teorema 2.7, é conveniente considerar os pontos (0,m), ($n+1$,m) como pontos *expostos fictícios.*

Teorema 2.8 *(a) Se não há pontos apropriados na linha superior, o problema possui solução se, e somente se, cada caminho é trivial ($L(i) = U(i)$).*

(b) Assuma que existe pelo menos um ponto exposto apropriado.

b1. Se $m < \Delta$, não há solução.

b2. Se $m > \Delta$, sempre existe solução.

b3. Se $m = \Delta$, existe solução se e somente se existem dois pontos expostos (fictícios ou não) na linha superior tal que o congestionamento de qualquer colunas entre eles seja menor que Δ.

Observe que o Teorema 2.7 está incluído, Se $m = \Delta$ e A é exposto, a "coluna"entre 0 e A tem congestionamento 0.

Prova: (a) trivial. Já vimos b1. Para b2, seja (X, m) um ponto exposto. aplique right-pushing em AX. Então temos um novo problema no retângulo de $m-1$ linhas horizontais. Mas agora o canto superior esquerdo se tornou exposto e $m-1 \geq \Delta$, e assim o Teorema 2.7 se aplica. A prova de b3 não é muito complicada. A necessidade pode ser provada por contradição. A prova da suficiência é feita aplicando operações de pushing e o Teorema 2.7.

Caso não-bipartido

A partir de agora assumimos que os terminais estão posicionados em qualquer lugar da borda de T. Além disso, os cantos podem conter terminais. Para resolver este problema precisamos na noção de conjuntos ímpares. Para isto, una cada par de terminais por uma nova aresta. Um subconjunto X de vértices de G é chamado *ímpar* se no grafo extendido, o número de arestas (novas ou velhas) saindo de X é ímpar. Um simples argumento de paridade mostra que nenhuma solução utiliza cada aresta deixando X. Chame um corte de G *saturado* se seu congestionamento é igual ao número de arestas nele. Vamos observar que colunas e linhas definem cortes de determinado tipo. Seja $r_1, r_2, ..., r_t$ o conjunto de linhas saturadas ($t \geq 0$) e seja c uma coluna qualquer. Isto define $t+1$ conjuntos disjuntos T_i no lado esquerdo de c, separados verticalmente pelos elementos r_i.

O número de conjutos ímpares de T é chamado *congestionamento de paridade* de c. O *congestionamento revisado* ou *r-congestionamento* de uma coluna c é a soma do congestionamento de paridade com o congestionamento de caminho de c.

Critério de coluna revisado. O r-congestionamento de qualquer coluna é no máximo m. Do mesmo modo definimos o *critério de linha revisado.*

Teorema 2.9 *Seja um retângulo em uma grade retilínea e k pares de terminais na borda, dados. Existe k caminhos disjuntos-nas-arestas entre os correpondentes terminais se e somente se os cirtérios de coluna e linha revisados valem.*

Este teorema envolve o Teorema 2.8, mas sua prova fornece um algoritmo mais eficiente que o descrito na Seção 2. Mais uma vez vou apenas dar a idéia central da prova neste resumo.

Prova: *Necessidade (do critério de coluna revisado).* Se T_i é ímpar, pelo menos uma aresta e saindo de T_i não será usada na solução. A aresta e está na coluna c, pois de outra forma e estaria em uma linha saturada, o que é impossível. Portanto o número m de arestas em c deve ser pelo menos o número de conjuntos ímpares T_i somado ao congestionamento de c, isto é, ao menos o r-congestionamento de c.

Suficiência. O teorema é trivial para $m = 1$, ou $n = 1$ então podemos supor que $m, n > 1$. Primeiro provamos o teorema quando não há conjuntos ímpares. Neste caso o r-congestionamento e o congestionamento são os mesmos. Ao invés de retângulos, vamos usar a idéia de n-retângulos. Por *retângulo-perto* ou *n-retângulo*, queremos dizer a região R limitada pelos segmentos AY, YX, XB, BC, CD, DA, onde $A = (1, m-1)$, $Y = (v, m-1)$, $X = (v, m)$, $B = (n, m)$, $C = (n, 1)$, $D = (1, 1)$, com $n > v \geq 1$e $m > 2$. O critério de linha (coluna) agora está levemente modificado: o congestionamento de qualquer linha (coluna) r deve ser no máximo o número de arestas em r. Os cantos convexos A, X, B, C e D podem ser atribuídos a dois ou nenhum terminais. O canto Y é atribuído a nenhum terminal e qualquer outro ponto da borda a exatamente um terminal. Dizemos que um n-retângulo é satisfatório se estas suposições e o critério de linha ou coluna vale.

Lema 2.3 *Em n-retângulos satisfatórios existem caminhos disjuntos-nas-arestas entre os correspondentes terminais.*

A prova do lema é feita por indução em R usando uma operação "cutting off" em X que substitui R por R' definido por $X' = (v+1, m)$, $Y' = (v+1, m-1)$, $A' = A$, $B' = B$, $C' = C$, $D' = D$ se $v < n-1$, e por $A' = A$, $B' = (m-1, n)$, $C' = C$, $D' = D$ se $v = n-1$. A prova é dividida em vários casos possíveis para provar que R' é satisfatório e que uma solução para R' implica em uma solução para R.

Com a prova do lema, segue-se a prova que o critério de coluna revisado implica a afirmação e dessa forma prova-se o Teorema 2.9

A prova do Teorema 2.9 envolve um algoritmo. Na primeira parte temos de parear os vértices ímpares, o que é fácil. A segunda parte consiste da aplicação de $m.n$ operações de "cutting off". Logo a complexidade é proporcional a $m.n$ comparando com o algoritmo do caso bipartido, o qual possui complexidade dependendo do número de caminhos a serem construídos.

Consequência. Se não existe nenhuma coluna ou linha saturada, o problema sempre tem solução.

Grafos planares
Um resultado relacionado de Okamura e Seymor consiste do problema dos caminhos disjuntos em grafos planares arbitrários com alguma propriedade fixa. Novamente os terminais ficam na borda. O *critério do corte* requer que o congestionamento de qualquer corte seja no máximo a cardinalidade do corte.

Teorema 2.10 *Se não existem conjuntos ímpares, o critério do corte é necessário e suficiente para a existência de caminhos disjuntos nas arestas entre os terminais correspondentes.*

O Lema usado no caso não bipartido pode ser facilmente derivado deste teorema e o algoritmo é simples para este caso. Seria bem vinda uma boa generalização dos Teoremas 2.9 e 2.10.

3 Problema de conexão e entrega ponto-a-ponto

Dois problemas mais recentes são os problemas de conexão e entrega ponto-a-ponto [5],[6].

Seja $G = (V, E)$ um grafo com origens $s_1, \ldots, s_p \in V$, destinos $t_1, \ldots, t_p \in V$ e uma função custo c_{uv} não-negativa para cada arco $u \to v \in E$. Cada origem tem a oferta de um item e cada destino a demanda de um item.O *problema de entrega ponto-a-ponto* é o problema de transportar p itens de origens para destinos de forma a minimizar o custo total de transporte.

Agora suponha este mesmo problema com valores de K "grandes" em um grafo G acíclico direcionado. Isto resulta num problema parecido, conhecido como problema de conexão ponto-a-ponto:

Dado um grafo $G = (V, E)$ orientado com origens $s_1, \ldots, s_p \in V$ e destinos $t_i, \ldots, t_p \in V$, e um custo não negativo $c_{i,j}$ para cada $i \to j \in E$, encontrar um subconjunto de arcos $E' \subseteq E$ em G tal que cada par origem-destino é conectado por um caminho que usa somente arcos de E' e que $\sum_{i \to j \in E'} c_{i,j}$ é minimizado.

Observe que caso $s_i = s, i = 1, \ldots, p$, então o problema de conexão se torna equivalente ao problema de árvore de Steiner que exige que os terminais $t_1, \ldots, t_p$ sejam conectados a s por uma árvore de Steiner, problema este $\mathcal{NP}$-completo como sabemos.

Esses dois problemas são o ponto principal desta monografia e serão apresentados a seguir.

3.1 Problema de entrega ponto-a-ponto [5]

Nesta seção é considerada a complexidade computacional de problemas de entrega ponto-a-ponto. Estes problemas envolvem transportar um item de uma dentre p origens para p destinos. Os destinos podem ser pareados às origens (caso de destino fixo) ou senão, um item saindo de uma origem pode ir para qualquer destino (caso não fixo). Os grafos podem ser direcionados ou não-direcionados. No máximo K itens podem compartilhar um caminhão por vez em um arco, e os custos são lineares no número de caminhões utilizados. É considerado também o problema de conexão ponto-a-ponto, que consiste em encontrar um subconjunto mínimo das arestas que conectam origens e destinos. É demostrado que todas as variantes do problema são $\mathcal{NP}$-difíceis para $K \geq 2$, mas também que existem algoritimos eficientes em alguns casos se p é fixo, ou se o grafo em questão é uma grade com origens em um lado e destinos em outro.

um grafo de entrega ponto-a-ponto é um grafo contendo vértices terminais e centros de distribuição. O tráfego origina-se e é destinado a *vértices terminais* (consistindo de vértices *origens* e *destinos*, enquanto os *centros de distruibuição* são vértices para transporte intermediário. Nosso objetivo é usar caminhões para carregar um item de determinada mercadoria de um vértice origem, e fazê-lo alcançar cada vértice de destino, de modo que o custo total de transporte seja minimizado, sujeito à capacidade dos caminhões. Este problema tem aplicações práticas em roteamento, entrega de pacotes, entrega de mails e transporte ferroviário. A seguir pretendo demostrar a complexidade e algoritmos ótimos para vários casos no problema de entrega ponto-a-ponto onde o único custo é linear no número de caminhões usado em cada arco, e onde todos os caminhões possuem a mesma capacidade.

Primeiramente vou apresentar as definições de diferentes variantes do problema de entrega ponto-a-ponto. Estas variantes valem também para o problema de conexão ponto-a-ponto que veremos a seguir. Mais adiante, as provas da $\mathcal{NP}$-dificuldade do problema são apresentadas e por fim são apresentados algoritmos em tempo polinomial para casos dos problemas de entrega onde o número de par origem-destino é fixo.

Definições e Notação

Problema 3.1 *Seja $G = (V, E)$ um grafo com origens $s_i, \ldots, s_p \in V$, destinos $t_1, \ldots, t_p \in V$, e um custo não negativo c_{uv} para cada arco $u \to v \in E$ (ou $uv \in E$). Cada origem tem uma oferta de um item e cada destino possui a demanda de um item. O* problema de entrega ponto-a-ponto *é o problema de transportar os p itens das origens para os destinos de forma a minimizar oo custo total de transporte, definido como segue: se um item precisa atravessar o arco uv, quando ele chega ao vértice u ele pode esperar (sem nenhum custo) por outros itens que querem atravessar o mesmo arco chegarem no vértice u, e então compartilhar um caminhão através do arco uv com eles. No entanto, cada caminhão possui uma capacidade de K itens. O custo de se despachar um caminhão no arco uv é c_{uv}, e o custo total de transporte no arco depende apenas do número de caminhões viajando através dele. Isto é, se n_{uv} caminhões são usados no arco uv, então o custo total de transporte é $\sum_{u \to v \in E} c_{uv} n_{uv}$, e no máximo $K.n_{uv}$ itens podem ser transportados através de uv.*

Neste artigo assume-se que todos os caminhões possuem a mesma capacidade e exatamente um item é enviado de cada origem para cada destino.

Existem diversas variantes deste problema: o grafo G pode ser direcionado ou não-direcionado, um item chegando num destino t_i pode ser exigido que tenha partido de s_i (caso *destino fixo*) ou pode ser oriundo de qualquer origem (*caso não fixo*).

Para diferenciar as variantes, foi desenvolvida uma notação composta de 3 componentes separados por barras. O primeiro define se o problema é de entrega, com capacidades de cada caminhão $= K$ (D_K), ou de conexão (C). O segundo representa se o caso é fixo (f) ou não fixo (n) em relação à correspondência das origens e destinos. O terceiro pode ser d, u, g ou g_t, onde d significa que o grafo é direcionado, u grafo não direcionado, g uma grade e g_t grade com origens num das faces e destinos na face oposta.

$\mathcal{NP}$-complexidade

Nesta seção são apresentadas provas de que as quatro variantes do problema de entrega ponto-a-ponto em grafos genéricos com $K \geq 2$ são fortemente $\mathcal{NP}$-difíceis, assim como os quatro problemas de conexão ponto-a-ponto também o são. As provas são todas baseadas em reduções de problemas $\mathcal{NP}$-completos já existentes. Não vou me aprofundar nas provas, apenas darei uma breve introdução sobre onde se baseiam.

Teorema 3.1 *$D_2/f/d$ é fortemente $\mathcal{NP}$-difícil.*

A prova é feita por redução do problema 3SAT (3-satisfatibilidade) para a versão de decisão de problema $D_2/f/d$ que pergunta se existe um roteamento de custo total M.

Corolário 3.1.1 *$D_2/f/u$ é fortemente $\mathcal{NP}$-difícil.*

A prova é feita apenas alterando a prova do Teorema 3.1 substituindo arestas dirigidas em G por arestas não-dirigidas de acordo com determinado critério.

Teorema 3.2 *$D_2/n/d$ é fortemente $\mathcal{NP}$-difícil.*

Mais uma vez a prova é feita baseada numa redução do 3SAT. Desta vez é usado uma versão esparsa do 3SAT chamada *Sparse 3SAT* e reduz-se ela ao problema de entrega ponto-a-ponto.

Corolário 3.2.1 *$D_2/n/u$ é fortemente $\mathcal{NP}$-difícil.*

Modifica-se o Teorema 3.2 substituindo arcos dirigidos por não dirigidos e alterando custos de determinadas arestas. A prova é similar ao Corolário 3.1.1

Teorema 3.3 *$D_K/{}^f_n/d$ é fortemente $\mathcal{NP}$-difícil para qualquer $K \geq 3$ fixo.*

A prova é feita através da transformação do X3C (cobertura exata por conjuntos de três elementos) num caso especial de ambas fixa e não-fixa variantes do problema de entrega.

Corolário 3.3.1 *$D_K/{}^f_n/u$ é fortemente $\mathcal{NP}$-difícil para qualquer $K \geq 3$ fixo.*

Novamente feito de uma alteração da versão direcionada do teorema 3.3 através de substituição e mudanças de custos e de arestas.

Teorema 3.4 *$C/{}^f_n/{}^u_d$ é fortemente $\mathcal{NP}$-difícil.*

Com $K = 3$ o problema de entrega ponto-a-ponto construído no Teorema 3.3 se torna o problema de conexão ponto-a-ponto e as provas do Teorema 3.3 e do Corolário 3.3.1 permanecem válidas e desta forma ambos $C/{}^f_n/u$ e $C/{}^f_n/d$ são fortemente $\mathcal{NP}$-difícil.

Número fixado de origens e destinos
Aqui é apresentado um algoritmo para um caso especial do problema de entrega ponto-a-ponto onde p, o número de pares origens/destinos, é fixado. Para o caso não direcionado pode-se facilmente reduzi-lo para o direcionado substituindo cada par de aresta $i-j$ por dois pares $i \to j$ e $j \to i$, assim é considerado apenas o caso do grafo ser direcionado.

Seja $\bar{G} = (\bar{V}, \bar{E})$ definido por:

$\bar{V} = \{< u_1, \ldots, u_p > | u_i \in V, i = 1, \ldots p\} = V \times V \times \ldots \times V, p$ vezes

$\bar{E} = \{< u_1, \ldots, u_p >\to< v_1, \ldots, v_p > \mid$ o número de i's tal que $u_i \neq v_i$ é pelo menos um e no máximo K, e existe um arco $u \to v \in E$ tal que para todo i com $u_i \neq v_i, u_i = u$ e $v_i = v\}$

onde o tamanho do arco $< u_1, \ldots, u_p >\to< v_1, \ldots, v_p >$ é o mesmo do tamanho (ou custo) do arco $u \to v$. Seja $s_1, \ldots, s_p$ e $t_i, \ldots, t_p$ as origens e destinos em G. Em $\bar{G}$, cada vértice representa uma potencial localização de p itens oriundos de $s_1, \ldots, s_p$, ou seja, o vértice $< u_1, \ldots, u_p >$ representa o estado onde o item i está atualmente no vértice u_i em $G, i = 1 \ldots p$. Cada arco $< u_1, \ldots, u_p >\to< v_1, \ldots, v_p >$ representa um único caminhão passando através de $u \to v$ mudando seu estado atual de $< u_1, \ldots, u_p >$ para $< v_1, \ldots, v_p >$. Logo um caminho mínimo de $< s_1, \ldots, s_p >$ para $< t_1, \ldots, t_p >$ em $\bar{G}$ fornece uma solução ótima para o problema de entrega ponto-a-ponto com destinos fixados. Para o caso não fixo, adiciona-se um vértice de origem $\bar{s}_0$ em $\bar{G}$ juntamente com arestas de custo zero $\bar{s} \to< s_{\phi(1)}, s_{\phi(2)}, \ldots, s_{\phi(p)} >$ para cada permutação $(\phi(1), \phi(2), \ldots, \phi(p))$ de $(1, 2, \ldots, p)$. Este caso pode ser então resolvido encontrando-se um caminho mínimos de $\bar{s}$ para $< t_1, \ldots, t_p >$.

Teorema 3.5 *$D_K/^f_n/^u_d$ com p origens e destinos pode ser resolvido em tempo $O(|V|^p log|V| + |V|^{p-1}|E|)$ quando $p \geq 2$ é uma constante.*

Prova vide [5].

3.2 Problema de conexão ponto-a-ponto [5],[6]

Este problema tem uma relação muito próxima ao problema de entrega ponto-a-ponto com capacidades "grandes" dos caminhões, e sua aplicação principal se encontra no design de circuitos VLSI. Um caso especial importante do problema de conexão em VLSI é quando o grafo em questão é uma grade. Este problema também está relacionado com problema de árvores de Steiner e mais ainda com os problemas de fluxo discutidos na seção 2.

O problema de conexão ponto-a-ponto consiste em encontrar um subconjunto de arcos de custo total mínimo conectando um número fixo de pares origem-destino. Este artigo foca-se no caso em que os pares origem-destino são previamente relacionados. É avaliada a estrutura de uma solução para o caso de dois pares origem-destino e um eficiente algoritmo baseado no Dijkstra é fornecido com complexidade de tempo de $\mathcal{O}(mn + n^2 \log n)$. Também é apresentado um algoritmo de programação dinâmica com complexidade de tempo $\mathcal{O}(n^{11})$ para o caso com três pares origem-destino. Uma conjectura que esta última aproximação pode ser generalizada para o caso de p pares origem-destino é apresentada no final do artigo.

Problema 3.2 *Dado um grafo $G = (V, E)$ orientada com origens $s_1, \ldots, s_p \in V$ e destinos $t_i, \ldots, t_p \in V$, e um custo não negativo $c_{i,j}$ para cada $i \to j \in E$, encontrar um subconjunto de arcos $E' \subseteq E$ em G tal que cada par origem-destino é conectado por um caminho que usa somente arcos de E' e que $\sum_{i \to j \in E'} c_{i,j}$ é minimizado.*

Este problema tem aplicação em redes de transportes onde o custo total corresponde ao custo fixado de configuração do grafo.

Li conseguiu resolver o problema para o caso $p = 2$ com complexidade $\mathcal{O}(n^5)$. Natu e Fangpropuseram um aperfeiçamento para $\mathcal{O}(n^4)$. Nesta seção, primeiramente é fornecido um algoritmo-Dijkstra com complexidade melhorada para $\mathcal{O}(mn + n^2 \log n)$.

Já o problema de conexão ponto-a-ponto com $p = 3$ também recebe atenção e sua análise considera diferentes possíveis configurações de uma solução ótima. Um algoritmo de programação dinâmica com complexidade de tempo $\mathcal{O}(n^{11})$ é fornecido. A análise ajuda a entender a decomposição usada no algoritmo de programação dinâmica e indica porque é difícil generalizar a prova construtiva para $p > 3$. Mesmo assim uma conjectura de que a mesma aproximação para $p = 3$ pode ser generalizada para qualquer p com complexidade $\mathcal{O}(n^{3p+2})$. é apresentada.

Problema de conexão ponto-a-ponto com $p = 2$

Estrutura de uma solução ótima
Dados 2 pares (s_1, t_1) e (s_2, t_2), seja c_{uv} o custo de um arco entre u e v, $C(u, v)$ o custo de um caminho arbitrário de u para v, e $C^*(u, v)$

o custo do menor caminho de u para v. Denote também um caminho entre s_1 e t_1 por P_1 e entre s_2, t_2 por P_2. Definimos P como *segmento compartilhado* se P é um subcaminho maximal comum de P_1 e P_2. Dois nós u, v são ditos *uv-conectados* se existe um caminho dirigido de u para v e este é denotado por $u \rightsquigarrow v$.

Uma solução ótima não pode ter a estrutura com um subcaminho indo através de $v_1 \rightsquigarrow i \rightsquigarrow v_2$ e outro indo através de $v_1 \rightsquigarrow j \rightsquigarrow v_2$ pois a solução pode ser rearranjada de modo a usar somente um destes subcaminhos. Se uma solução ótima possui dois caminhos P_1^* e P_2^* com k segmentos compartilhados, eles precisam compartilhar estes segmentos na ordem inversa. Isto é o que chamamos de *propriedade escada*.

A propriedade da escada de uma solução ótima nos permite decompor o problema em diversos subproblemas e transformar o problema em um problema de caminho mínimo com $\mathcal{O}(n^2)$ nós, onde n é o número de nós do grafo original.

Algoritmo Dijkstra

Constrói-se um grafo de "duas dimensões" $\bar{G} = (\bar{V}, \bar{E} \equiv \bar{E}^1 \cup \bar{E}^2)$ onde:

- $\bar{V} = \{< v_1, v_2 > | v_1, v_2 \in V\} = V \times V$,
- $\bar{E}^1 = \{< v_1, v_2 > \rightarrow < v_1', v_2' > | v_2 \rightsquigarrow v_1' \rightsquigarrow v_2' \rightsquigarrow v_1$, e $< v_1, v_2 > \neq < s_2, t_1 >, < v_1, v_2 > \neq < v_1', v_2' >\}$,
- $\bar{E}^2 = \{< v_1, v_2 > \rightarrow < s_2', t_1' > | s_2 \rightsquigarrow v_1, v_2 \rightsquigarrow t_1, < v_1, v_2 > \neq < s_2, t_1 >\}$.

Considere também em $\bar{E}^1$ o custo de $< v_1, v_2 > \rightarrow < v_1', v_2' >$ sendo $C^*(v_2, v_1') + C^*(v_1', v_2') + C^*(v_2', v_1)$; e em $\bar{E}^2$, o custo de $< v_1, v_2 > \rightarrow < s_2, t_1 >$ sendo $C^*(s_2, v_1) + C^*(v_2, t_1)$.

A partir desta construção, prova-se que pode-se obter uma solução para o problema de conexão ponto-a-ponto a partir da resolução do problema de encontrar um caminho mínimo no grafo construído.

Uma vez que $\bar{G}$ possui (n^2) nós e $(2m+n^2)$ arcos, a construção da novo grafo consome tempo $\mathcal{O}(mn)$. Para melhorar a computação, pode-se fazer pré-processamento para obter $(C^*(u, v)$ em $\mathcal{O}(mn + n^2 \log n)$ usando heaps de Fibonacci para encontrar todos os caminhos mínimos. O algoritmo de caminho mínimo Dijkstra no grafo $\bar{G}$ leva $\mathcal{O}(mn +$

$n^2 \log n$) e consequentemente, a complexidade do algoritmo para $p = 2$ é $\mathcal{O}(mn + n^2 \log n)$.

Problema de conexão ponto-a-ponto com $p = 3$
Para o caso $p = 3$, a simples estrutura da seção 2 não pode ser considerada. Definições e notações anteriores permanecem válidas, exceto que algumas delas devem ser extendidas.

Seja P_i um caminho entre uma origem s_i e um destino t_i. A partir de agora, um segmento compartilhado é referido como sendo um subcaminho comum maximal entre quaisquer caminhos ligando os $p(\geq 3)$ pares origem-destino. É maximal com respeito a duas propriedades: (i) maximal no número de par origem-destino que compartilham o segmento; (ii)não está contido em nenhum subcaminho com mais arcos e compartilhado pelo mesmo par origem-destino. Em outras palavras, qualquer arco desse subcaminho maximal comum não pode ser usado por outro caminho conectando um par origem-destino. O "problema" no caso $p \geq 3$ é o número de possíveis estruturas que os caminhos compartilhados podem tomar e a necessidade de considerá-los todos para se elaborar o algoritmo de programação dinâmica. A solução é feita então a partir da decomposição do problema em subproblemas.

O artigo define então uma função extremamente complexa para o algoritmo de programação dinâmica e após a apresentação do que consite o algoritmo, mostra através de uma prova que sua complexidade de tempo é $\mathcal{O}(n^{11})$, baseado em valores de pré-processamento idênticos aos da seção 2, bem como um limite superior para o número de segmentos compartilhados.

Conjectura de solução para $p > 3$
Os autores apresentam uma conjectura de que a formulação em programação dinâmica da seção anterior pode ser extendida para $p > 3$. Isto é feito definindo a função para o algoritmo de programação dinâmica e a relação de recorrência. A conjectura segundo os autores, deve resolver o problema para $p > 3$ pares origem-destino em tempo $\mathcal{O}(n^{3p+2})$. Observa-se ainda que a dificuldade da prova não está na complexidade de tempo, mas sim no fato da grande complexidade da estrutura de uma solução ótima acarretar muita dificuldade para se generalizar em

valores > 3.

Problema de conexão ponto-a-ponto em uma grade
Nesta seção considera-se o caso especial do problema de conexão ponto-a-ponto onde o grafo dada é uma *grade*. Vamos utilizar a notação descrita no problema de entrega ponto-a-ponto.

Teorema 3.6 *$C/^f_n/g$ é fortemente $\mathcal{NP}$-difícil.*

Prova: O problema de encontrar uma árvore de Steiner mínima em uma grade é $\mathcal{NP}$-completo. Pode-se transformar o problema de árvore de Steiner mínima em grades no problema de conexão ponto-a-ponto e nisso a prova consiste.

Agora seja uma grade onde todas as origens se encontram em uma das faces e os destinos estão na face oposta. O problema $C/^f_n/g_t$ é o problema de conexão ponto-a-ponto nestas grades. Observe que estudamos este problema anteriormente na seção 2. Este problema se assemelha com o caso bipartido do problema de caminhos disjuntos em grades retilíneas e, como visto anteriormente, possui solução polinomial.

4 Conclusão

O objetivo de meu estudo neste curso foi obter uma relação entre diversos problemas de conexão em grafos, ligados principalmente ao problema de Steiner.

O problema de conexão ponto-a-ponto tem relação direta com diversos dos problemas aqui apresentados. Observando casos particulares do problema, podemos ver semelhança com os demais. O problema do empacotamento de árvores de Steiner e o problema de caminhos disjuntos em grades são dois exemplos disto. No primeiro caso, o problema de conexão ponto-a-ponto é um caso particular do empacotamento para conjuntos com cardinalidade $= 2$. Já no segundo caso, basta considerar o grafo dado para o problema de conexão ponto-a-ponto como sendo uma grade e teremos como resultado o problema de se encontrar caminhos disjuntos numa grade retilínea.

Outra relação importante diz respeito à complexidade dos problemas. O problema de conexão, é derivado do problema de entrega ponto-a-ponto que, assim como a maioria dos problemas estudados, tem complexidade relacionada ao problema de Steiner, que por sua vez é $\mathcal{NP}$-completo.

A escolha dos problemas a serem estudados, teve como raiz o problema de conexão ponto-a-ponto. A partir de sua bibliografia fui descobrindo outros problemas que caracterizavam casos particulares do ponto-a-ponto ou problemas dos quais o ponto-a-ponto se originou, culminando então com a análise deste estudo presente nesta monografia.

5 Bibliografia

Referências

[1] A. Frank. Disjoint paths in a rectilinear grid. *Combinatorica*, (2):361–371, 1982.

[2] M.R. Gray and D.S. Johnson. The rectilinear steiner tree problem is np-complete. *SIAM J. Appl. Math.*, (32):826–834, 1977.

[3] M. Grötschel, A. Martin, and R. Weismantel. Packing steiner trees: cutting plane algorithm and computational results. *Mathematical Programming*, A(72):124–145, 1996.

[4] S.L. Hakimi. Steiner's problems in graphs and its applications. *Networks*, (1):113–133, 1971.

[5] Chung-Lun Li, S. Thomas McCormick, and David Shimchi-Levi. The point-to-point delivery and connection problems: complexity and algorithms. *Discrete Applied Mathematics*, (36):267–292, 1992.

[6] Madan Natu and Shu-Chern Fang. The point-to-point connection problem - analysis and algorithms. *Discrete Applied Mathematics*, (78):207–226, 1997.

[7] H. Okamura and P. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory*, B(31):75–81, 1981.

[8] H.J. Prömel and Steger A. A new approximation algorithm for the steiner tree problem with performance 5/3. *Journal of Algorithms*, (36):89–101, 2000.

Data: 04/03/2004, segunda-feira

Orientador: Carlos Eduardo Ferreira

Professor da disciplina: Yoshiko Wakabayashi