O Problema do Multicorte Mínimo em Digrafos

Aluno: Pedro Luis Furio Raphael
Orientador: Professor Paulo Feofiloff
MAC0499 - Trabalho de Formatura Supervisionado
Palavras Chave: multicorte - digrafo - árvore - anel

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

1. Introdução

Considere $G = (V, E)$ um digrafo, isto é, um grafo dirigido, com uma função peso p que associa um numero não-negativo a cada arco de E, e considere um conjunto $S = \{(s_i, t_i) \,|\, i = 1, \ldots, K, s_i \neq t_i\}$ de pares ordenados de vértices, onde s_i é chamado de *fonte* e t_i é chamado de *sorvedouro* e o *tamanho* de S é K.
Um *multi corte* em um digrafo G é um subconjunto de arcos de E que se for retirado de G passe a não existir caminho dirigido entre s_i e t_i, para todos os pares (s_i, t_i) de S.
Desta forma, o problema do multi corte mínimo consiste em encontrar um multi corte M que tenha peso mínimo, ou seja $\sum_{e \in M} p(e)$ seja o menor possivel.
Este problema é o clássico min cut-max flow quando o $K = 1$. No entanto, ele é NP-difícil para $K \geq 3$ [4].
Existe um problema parecido a este, considerando grafos não-dirigidos. No entanto, dificilmente as estratégias e resultados obtidos para grafos podem ser adaptadas para digrafos, devido a natureza do problema [3].
Nesta monografia, analisaremos o problema para dois tipos específicos de digrafos, árvores [2] e anéis [1], e mostraremos algoritmos que resolvem este problema em tempo polinomial para estes digrafos.

2. Árvores

2.1 Introdução

Uma *árvore* $T = (V, E)$ é um grafo orientado com as seguintes propriedades:

i) Existe um único vértice em T cujo grau de entrada é 0, chamado de raiz;

ii) Todos os demais vertices de T tem grau de entrada 1;

iii) Para todo vértice $v \in T$ existe um caminho que sai da raiz e chega em v.

Definiremos a *distancia* entre um vértice e a raiz como sendo o número de arcos que temos que percorrer partindo da raiz até encontrá-lo.
Associamos a T uma função peso p e um conjunto S de pares ordenados de vértices. Vem imediatamente da definição de árvore a seguinte propriedade:

- Seja $(s_i, t_i) \in S$ um par qualquer. Existe no máximo um caminho que sai de s_i e chega em t_i.

Isto pode ser deduzido intuitivamente. Comece por t_i e ande na direção oposta ao arco que chega nele. Deste modo, uma das duas coisas será verdadeira: alcançaremos s_i e portanto este é o único caminho entre eles, ou alcançaremos a raiz e não há caminho entre eles.

2.1.1 Tamanho de S

Dada uma árvore $T = (V, E)$, qual o tamanho máximo de S? Uma primeira análise nos mostra que S pode ter no máximo n^2 elementos. Para cada vértice $v \in V$, associe $(n-1)$ pares do tipo (v, w), para cada $w \in V, w \neq v$. Deste modo, o número de elementos de S pode ser da ordem de n^2.
No entanto, muitos destes pares serão irrelevantes para o problema, dado que *não existirá caminho entre a fonte e o sorvedouro para alguns deles*, por exemplo, para um par (s_i, t_i), cuja distancia entre t_i e a raiz for menor que a distancia entre s_i e a raiz. Deste modo, precisamos de uma análise mais cuidadosa.
Primeiramente, podemos retirar de S todos os pares para os quais não há caminho entre s_i e t_i. Além disso, considere:
Definição: Dizemos que um par $(s_i, t_i) \in S$ está contido em um par $(s_j, t_j) \in S$, $j \neq i$ se p_i (o caminho entre s_i e t_i) está inteiramente contido em p_j, ou seja, todos os arcos que formam p_i estão também em p_j.
Portanto, se um par $(s_i, t_i) \in S$ está contido em outro par $(s_j, t_j) \in S$, obviamente uma arco que separa s_i de t_i também separa s_j de t_j. Desta forma, podemos retirar o par (s_j, t_j) de S sem alterar a solução do problema.
Este fato nos leva a seguinte conclusão: cada vértice $v \in V$ pode ser o sorvedouro de um único par. Caso v seja sorvedouro de mais de um par em S então, um desses pares está contido no outro, e podemos eliminá-lo sem alterar a solução. Desta forma, como existem n vértices em V, podemos limitar o tamanho de S para n pares, onde cada par tem o sorvedouro em um vértice diferente.

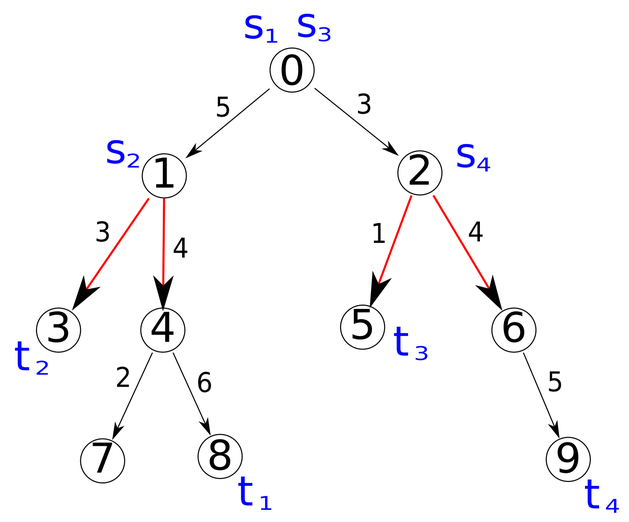

Figura 1: *Exemplo de árvore, associada com uma função peso e um conjunto S (elementos em azul). As arestas em vermelho formam um multi corte mínimo.*

2.2 Algoritmo

Segue o pseudo-código do algoritmo:
ALGORITMO MultiCorteMínimo-Árvore
Recebe: Uma árvore $T = (V, E)$, uma função p que associa um número não-negativo a cada arco de E e um conjunto de pares ordenados de vértices S.
Devolve: Um multi corte M de peso mínimo.

ALGORITMO MultiCorteMínimo-Árvore(T, u, S)

1. Ordene os pares de S em ordem decrescente de distância entre a fonte e a raiz.
2. $M \leftarrow \emptyset$
3. $u' \leftarrow u$
4. **para** $k \leftarrow 1$ **até** K **faça**
5. $\quad p_k \leftarrow$ caminho entre s_k e t_k
6. $\quad f_k \leftarrow$ valor de $\min_{e \in p_k}\{u'(e)\}$
7. $\quad$ **para cada** arco $e \in p_k$ **faça**
8. $\quad\quad u'(e) \leftarrow u'(e) - pesoMinimo$
9. $\quad\quad$ **se** $u'(e) = 0$
10. $\quad\quad\quad$ **então** $M \leftarrow M \cup \{e\}$
11. Ordene os pares de S em ordem crescente de distancia entre a fonte e a raiz.
12. **para** $k \leftarrow 1$ **até** K **faça**
13. $\quad$ **se** $f_k > 0$ **então**
14. $\quad\quad p_k \leftarrow$ caminho entre s_k e t_k
15. $\quad\quad e \leftarrow$ primeiro arco de p_k que está em M.
16. $\quad\quad$ Retira-se de M todas os outros arcos de p_k.
17. $\quad\quad M \leftarrow M \cup \{e\}$
18. **devolva** M

2.3 Análise de Complexidade

A análise de complexidade deste algoritmo é simples. Ela depende da complexidade de se descobrir um caminho entre dois vértices em uma árvore e do tamanho do conjunto S. Para encontrar o caminho entre dois vértices de uma árvore, digamos $v, w \in V$, partimos de w e percorremos ao contrário a única aresta que chega em w. Desta forma, assumindo que o caminho entre v e w existe, chegaremos a v em no máximo n passos, onde n é o número de vértices da árvore. Assim, achar um caminho entre dois vértices quaisquer em uma árvore é $O(n)$.
Continuando a análise, a linha 5 acha o caminho entre (s_k, t_k) em $O(n)$. Na linha 6, percorremos este caminho para achar o menor peso, também $O(n)$. Finalmente nas linhas 8-10, o **para** percorre todos os arcos do caminho achado na linha 5, também $O(n)$.
O **para** que começa na linha 4 e acaba na 10 roda K vezes, então, esta fase algoritmo roda em $O(Kn)$.
A segunda fase, que começa no **para** da linha 12 e vai até a 17 também roda K vezes. Na linha 14, achamos o caminho entre dois vértices em $O(n)$. Desta forma, esta fase também roda em $O(Kn)$ e o algoritmo todo roda em $O(Kn)$.

3. Áneis

3.1 Definição

Um anel $R = (V, E)$ é um digrafo com as seguintes propriedades:

i) Todos os vértices tem grau de entrada 1 e grau de saida também 1.

ii) É conexo, ou seja, partindo de um vértice qualquer consiguimos chegar em todos os outros.

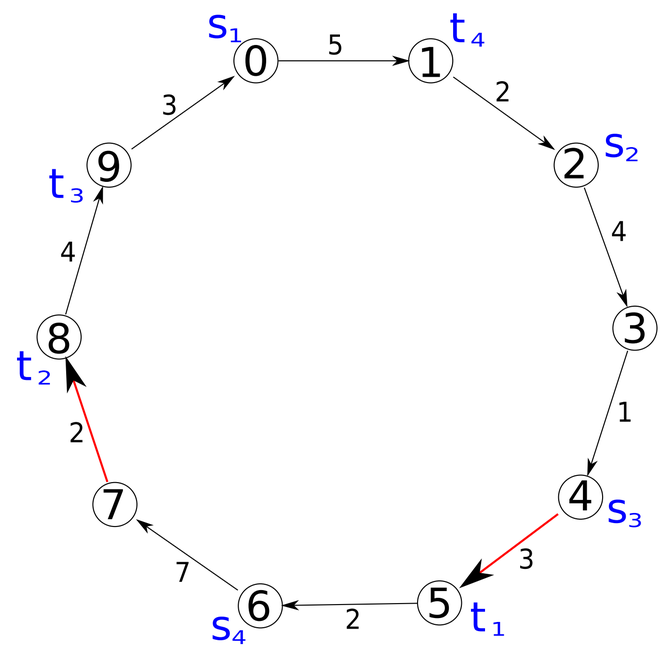

Figura 2: *Exemplo de anel, associada com uma função peso e um conjunto S (elementos em azul). As arestas em vermelho formam um multi corte mínimo.*

Ultimamente, principalmente no setor de Telecomunicações, estas estruturas tem ganhado destaque, impulsinando o estudo de algoritmos na área.

3.2 Algoritmo

Segue o pseudo-código do algoritmo:
ALGORITMO MultiCorteMínimo-Anel
Recebe: Uma anel $R = (V, E)$, uma função p que associa um número não-negativo a cada arco de E e um conjunto de pares ordenados de vértices S.
Devolve: Um multi corte M de peso mínimo.

ALGORITMO MultiCorteMínimo-Anel(R, u, S)

1. $M \leftarrow E$
2. $(s, t) \leftarrow$ elemento qualquer de S
3. **para cada** e no caminho entre (s, t) **faça**
4. $\quad R' = (V, E') \leftarrow$ novo grafo tal que $E' \leftarrow E - e$
5. $\quad S' \leftarrow S -$ { pares cujo caminho contém e }
6. $\quad M' \leftarrow$ MultiCorteMínimo-Árvore(R', u, S')
7. $\quad M' \leftarrow M' \cup \{e\}$
8. $\quad$ **se** $p(M') < p(M)$ **então**
9. $\quad\quad M \leftarrow M'$
10. **devolva** M

3.3 Análise de Complexidade

A complexidade deste algoritmo baseia-se na complexidade do algoritmo MultiCorteMínimo-Árvore, senda esta $O(Kn)$. Como um caminho em R contém não mais que n arcos, a complexidade de MultiCorteMínimo-Anel é de $O(Kn^2)$.

Referências

[1] C. Bentz, M.C. Costa, L. Létocart, and F. Roupin, *Multicuts and integral multiflows in rings*, European Journal of Operational Research **196** (2009), no. 3, 1251–1254.

[2] M.C. Costa, L. Létocart, and F. Roupin, *A greedy algorithm for multicut and integral multiflow in rooted trees*, Operations Research Letters **31** (2003), no. 1, 21–27.

[3] M.C. Costa, L. Letocart, and F. Roupin, *Minimal multicut and maximal integer multiflow: a survey*, European Journal of Operational Research **162** (2005), no. 1, 55–69.

[4] N. Garg, V.V. Vazirani, and M. Yannakakis, *Primal-dual approximation algorithms for integral flow and multicut in trees*, Algorithmica **18** (1997), no. 1, 3–20.