Online Survivable Network Design

Autor: Mário César San Felice.

Fonte: Artigo de Gupta, Krishnaswamy e Ravi.

06 de abril de 2011, IC-Unicamp.

Problema de Survivable Network Design

- Problema de construir uma rede com tolerância a falhas.
- Podemos considerar falhas nos links ou nos nós da rede.
- Vamos tratar a versão de aresta conectividade.
- Esta versão do problema generaliza o problema da Árvore de Steiner.

Problema de Survivable Network Design

- A entrada é composta por um grafo $G = (V, E)$ com custo $c(e)$ nas arestas e requisição de conectividade r_{ij} entre cada par de vértices.

- Uma solução para o problema é um subgrafo H de G no qual para todo par de vértices i e j existem r_{ij} caminhos disjuntos nas arestas conectando-os.

Versão *Online* do Problema

- Problemas *online* são problemas em que alguma parte da entrada é recebida em sequência e cada item da sequência é processado antes do seguinte.
- Na versão online que consideramos são as requisições de conectividade entre vértices que chegam em sequência.
- Para cada requisição que chega devem ser compradas arestas para garantir a conectividade exigida por ela antes que chegue a requisição seguinte.

Análise Competitiva

- Técnica de análise de algoritmos *online*.
- Apresenta como garantia uma razão de competitividade (c) que satisfaz:

$$ALG(I) \leq c \times OPT(I) + \alpha,$$

para toda entrada I.

- Trata-se de uma análise de pior caso.

Resultado Principal do Artigo

- Algoritmo *online* (ALG_{ND}) para a versão de aresta conectividade do problema com requisição de conectividade máxima k (k-ECND).
- ALG_{ND} possui razão de competitividade $\tilde{O}(k \log^2 m \log n)$.
- Utiliza imersão de grafos em árvores com alguma garantia de preservação de distâncias.
- Utiliza como subrotina um algoritmo *online* para o problema *Hitting Set*.
- A complexidade do algoritmo é $O(m^k)$.

Idéias Centrais

- Consideramos o cenário em que temos um subgrafo H de G que l-conecta um par de vértices (s_i, t_i) e queremos adicionar arestas a H para $(l+1)$-conectar este par.
- Podemos construir uma instância do problema de *Hitting Set* em que cortes correspondem a conjuntos e arestas correspondem a elementos.
- Esta idéia não funciona pois existem $M = 2^{n-1}$ cortes/conjuntos e a melhor aproximação para o problema *Hitting Set* é $O(\log M)$.

Idéias Centrais

- Como em H cada $s_i - t_i$ corte mínimo tem l arestas podemos limitar o número de cortes para $M = \binom{m}{l}$.
- No entanto, para cada corte em H podem existir diversos cortes correspondentes em G.
- Outra alternativa é atingir os cortes usando caminhos ao invés de arestas.
- No entanto, pode existir um número exponencial de caminhos.

Idéias Centrais

- Podemos encontrar uma árvore geradora aleatória T de G com pequena distorção nas distâncias.
- O teorema principal do artigo mostra que é possível aumentar a conectividade utilizando apenas circuitos fundamentais com relação a T.
- Além disso é possível relacionar o custo dos circuitos fundamentais com o custo do algoritmo offline ótimo.

Imersão em Árvores e Grafos *Backboned*

- Usando resultados de Elkin et al. [2] e Abraham et al. [3] obtemos uma árvore geradora aleatória T de G com pequena distorção esperada nas distâncias.
- Para toda aresta $e = (u, v) \notin T$ temos $E[d_T(u, v)] \leq \tilde{O}(\log n) d_G(u, v)$.
- Construímos então um grafo G_T com custos $\hat{c}$ em que:
  - $\forall e \in E_T$ temos $\hat{c}(e) = c(e)$.
  - $\forall e = (u, v) \in E_T$ temos $\hat{c}(e) = \max\{d_T(u, v), c(e)\}$.

Imersão em Árvores e Grafos *Backboned*

- Definimos um grafo *Backboned* como um grafo G que possui uma árvore geradora T tal que para toda aresta $e = (u, v) \notin E_T$ temos $c(e) \geq d_T(u, v)$. Chamamos T de árvore base de G.

- Notem que, por construção, o grafo G_T é um grafo *Backboned* com árvore base T.

Teorema (Imersão em Grafos *Backboned*)

Dado um algoritmo online *β-competitivo para o problema k-ECND em grafos* Backboned *temos um algoritmo probabilístico com competitividade esperada $\beta \times \tilde{O}(\log n)$ para o problema em grafos gerais.*

Circuitos Fundamentais e Cobertura de Cortes

- A partir de agora consideramos que nosso grafo G é um grafo *Backboned* com árvore base T.
- Dada uma aresta $e = (u, v) \notin E_T$ definimos um circuito fundamental O_e como o circuito formado pela união de e com $P_T(u, v)$.
- Como o grafo é *Backboned* temos $c(O_e) \leq 2c(e)$.

Circuitos Fundamentais e Cobertura de Cortes

- Seja H um subgrafo de G que l-conecta um par de vértices $s_i - t_i$ e contém $P_T(s_i, t_i)$.
- Definimos $\mathcal{P}_i$ como o conjunto dos l caminhos disjuntos nas arestas que conectam $s_i - t_i$ em H.
- Cada l-corte que desconecta $s_i - t_i$ em H tem uma aresta de cada caminho de $\mathcal{P}_i$.
- Definimos $viol_H(i)$ como o conjunto formado por estes l-cortes.

Circuitos Fundamentais e Cobertura de Cortes

- Dado um l-corte $Q \in viol_H(i)$, vamos rotular os vértices de G em função de H e Q.
- Rotulamos os vértices v que são ponta de arestas de Q da seguinte forma:
  - Se v é atingível por s em $H \setminus Q$ rotulamos v com S.
  - Se v é atingível por t em $H \setminus Q$ rotulamos v com T.
- Todos os demais vértices de G são rotulados com U.

Circuitos Fundamentais e Cobertura de Cortes

Teorema (Cobertura de Cortes)

Seja H um subgrafo de G que l-conecta $s_i - t_i$ e contém $P_T(s_i, t_i)$. Para qualquer l-corte $Q \in viol_H(i)$, dados os rótulos correspondentes a H e Q, existe uma aresta e que pertence a uma solução offline *ótima e cujo circuito fundamental O_e conecta algum vértice S com algum vértice T.*

- Com isso temos que $s_i - t_i$ estão conectados em $(H \cup O_e) \setminus Q$.
- Notem que o teorema vale para qualquer subgrafo H que contém o caminho $P_T(s_i, t_i)$, para o qual Q é um l-corte separando $s_i - t_i$ e para o qual valem os rótulos S, T e U.

Algoritmo *Online* usando *Hitting Set*

- Dada uma instância do problema k-ECND em grafos *Backboned* construímos uma instância $\mathcal{I}$ do problema *Hitting Set* de modo que:
  - Para cada aresta e temos um elemento $\bar{e}$ com custo $c(\bar{e}) = c(O_e) \leq 2c(e)$. Assim, o universo de elementos tem tamanho $N = m$.
  - Para cada corte rotulado Q temos um conjunto $\bar{Q}$. Assim, para cada l entre 1 e $k-1$ temos uma família $\mathcal{F}_l$ de conjuntos com tamanho $M_l = \binom{m}{l}2^l$.
  - Com isso, a família de todos os cortes rotulados $\mathcal{F} = \bigcup_{l=1}^{k} \mathcal{F}_l$ tem tamanho $M = O((2m)^k)$.
  - Um elemento $\bar{e}$ pertence a um conjunto $\bar{Q}$ se $O_e \setminus Q$ conecta um vértice S a um vértice T.

Algoritmo *Online* usando *Hitting Set*

- Quando chega um par de vértices $s_i - t_i$, o algoritmo *online* (ALG_{ND}) coloca em H as arestas faltantes do caminho $P_T(s_i, t_i)$. Assim, temos que H 1-conecta $s_i - t_i$.
- Em seguida o algoritmo realiza $k - 1$ aumentações de conectividade, de modo que no final destas aumentações H k-conecta $s_i - t_i$.
- Numa aumentação em que o subgrago H l-conecta $s_i - t_i$ ALG_{ND} envia para um algoritmo *online* para o problema *Hitting Set* (ALG_{HS}) os conjuntos que correspondem a l-cortes rotulados que separam $s_i - t_i$ em H.

Algoritmo *Online* usando *Hitting Set*

- Para cada elemento $\bar{e}$ que ALG_{HS} escolhe, ALG_{ND} adiciona as arestas de O_e a H.
- Como todo l-corte de $viol_H(i)$ recebe pelo menos uma aresta, H passa a $l+1$-conectar $s_i - t_i$.
- O algoritmo *online* para *Hitting Set* de Alon et al. [4] é $O(\log N \log M) = O(k \log^2 m)$-competitivo.

Algoritmo *Online* usando *Hitting Set*

Teorema

O algoritmo ALG_{ND} tem razão de competitividade $O(k \log^2 m)$ para o problema k-ECND em grafos Backboned.

- Considere que OPT_{ND} é uma solução *offline* ótima para o problema k-ECND.
- O custo que ALG_{ND} paga para comprar arestas da árvore base T é limitado pelo custo de OPT_{ND}.

Algoritmo *Online* usando *Hitting Set*

- Pelo teorema da Cobertura de Cortes para cada corte/conjunto que ALG_{HS} tem de atingir existe uma aresta de OPT_{ND} cujo circuito fundamental atinge o corte/conjunto.
- Portanto, os elementos correspondentes às arestas da solução OPT_{ND} constituem uma solução viável para a instância do problema *Hitting Set* resolvido por ALG_{HS}.

Algoritmo *Online* usando *Hitting Set*

- Como para todo elemento $\bar{e}$ vale $c(\bar{e}) \leq 2c(e)$, temos que $c(OPT_{HS}) \leq 2c(OPT_{ND})$.
- Como $c(ALG_{ND}) = c(ALG_{HS}) \leq O(k \log^2 m) OPT_{HS}$ temos que $c(ALG_{ND}) \leq O(k \log^2 m) OPT_{ND}$.
- Pelo teorema da Imersão em Grafos *Backboned* temos um algoritmo *online* probabilístico para grafos gerais com razão de competitividade $\tilde{O}(k \log^2 m \log n)$.

Referências

A. Gupta, R. Krishnaswamy, R. Ravi.
Online and Stochastic Survivable Network Desing.
41st STOC, 685–694, 2009.

M. Elkin, Y. Emek, D.A. Spielman, S.H. Teng.
Lower-Stretch Spanning Trees.
37th STOC, 494–503, 2005.

I. Abraham, Y. Bartal, O. Neiman.
Nearly Tight Low Stretch Spanning Trees.
49th FOCS, 781–790, 2008.

N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, J.S. Naor.
The Online Set Cover Problem.
35 STOC, 100–105, 2003.