PCC173/BCC463 - Otimização em Redes

Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

Conteúdo

Aviso

Fonte

Este material é baseado no material

- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1988). *Network flows*. PHI, Englewood Cliffs NJ.
- Top Coder. *Minimum cost flow part II: algorithms*. Disponível em `https://www.topcoder.com/community/data-science/data-science-tutorials/minimum-cost-flow-part-two-algorithms/`.

Licença

Algoritmo de Caminhos Mais Curtos Sucessivos

Princípio

Como visto, o algoritmo de Cancelamento de Ciclos resolve o problema de fluxo máximo como uma subrotina.

O algoritmo de Caminhos Mais Curtos Sucessivos busca o fluxo máximo e otimiza a função objetivo do problema de fluxo de custo mínimo ao mesmo tempo, resolvendo o assim chamado **Problema de Fluxo Máximo a Custo Mínimo**.

Este algoritmo utiliza novamente a rede transformada e o conceito de rede residual, vistos anteriormente, em que dois vértices artificiais de oferta e demanda e também arcos artificiais de custo zero ligando-os aos verdadeiros vértices de oferta e demanda são adicionados.

A capacidade dos arcos (s,i) é b_i e dos arcos $(i,\ t)$ é $-b_i$.

Algoritmo de Caminhos Mais Curtos Sucessivos

Princípio

Ao invés de determinar o fluxo máximo na rede original como no algoritmo de cancelamento de ciclos, o fluxo entre s e t é aumentado pelo caminho mais curto em relação ao custo dos arcos na rede residual.

A rede residual é então atualizada, caminhos mais curtos são determinados sucessivamente e o fluxo é atualizado até que não haja mais caminhos entre s e t.

O fluxo então é máximo e corresponde a uma solução viável do problema original de fluxo de custo mínimo – com efeito, também é a solução ótima para o problema original.

Este algoritmo não mantém a viabilidade da solução a cada instante, porém, mantém a otimalidade.

Algoritmo de Caminhos Mais Curtos Sucessivos

Entrada: Rede R

Transforme a rede R pela adição de vértices e arcos artificiais;

`// O fluxo inicial é zero`

Construa R_f a partir de R;

enquanto R_f *contiver um caminho entre* s *e* t **faça**

  Determine o caminho mais curto P entre s e t em R_f;

  $\delta \leftarrow min\{r_{ij} : (i,j) \in P\}$;

  Aumente o fluxo em P na rede R em δ unidades de fluxo;

  Atualize a rede residual R_f;

fim

Algoritmo de Caminhos Mais Curtos Sucessivos

Complexidade

O algoritmo de Caminhos Mais Curtos Sucessivos realiza no máximo $O(nB_{max})$ aumentos de fluxo, em que B_{max} é o limitante superior para oferta/demanda em qualquer vértice.

Se utilizarmos um algoritmo $O(nm)$ para determinar os caminhos mais curtos (podem haver arestas de custo negativo), teremos complexidade total $O(n^2mB_{max})$.

Para efeito de comparação, a implementação "ruim" do algoritmo de cancelamento de ciclos possui complexidade $O(nm^2C_{max}U_{max})$, em que C_{max} denota o custo máximo de um arco e U_{max} denota a capacidade máxima de um arco.

Alternativa

Os caminhos entre s e t em cada iteração do algoritmo podem ser substituídos por caminhos entre vértices com oferta e demanda não completamente atendidos.

Algoritmo de Caminhos Mais Curtos Sucessivos

Detalhe

Eventualmente, o caminho mais curto na rede residual poderá conter arcos de custo negativo, o que exige o uso de algoritmos com maior complexidade.

Entretanto, podemos realizar uma segunda transformação na rede de maneira que todos os arcos tenham custo positivo, reponderando os arcos usando o **potencial** de cada vértice, por exemplo.

Desta forma, podemos reduzir a complexidade necessária para determinar os caminhos mais curtos.

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

Potencial de um Vértice – Definição Alternativa

O **potencial** π_i de um vértice i é dado pelo custo do caminho mais curto entre s e i, calculado, por exemplo, pelo algoritmo de *Bellman-Ford*.

Custos Reduzidos dos Arcos

A cada caminho mais curto sucessivo determinado, atualizamos o custo reduzido de cada arco.

Os custos reduzidos são calculados de maneira que arcos pertencentes ao caminho mais curto entre s e qualquer vértice terão valor zero após a reponderação dos arcos.

Isto implica na não inclusão de arcos reversos (j, i) com custo negativo na rede residual, teremos $c_{ji} = 0$.

Arcos reversos (j, i) são importantes para evitar fluxos blocantes não máximos.

Procedimento para Cálculo de Custos Reduzidos

Custo_Reduzido
Entrada: Rede residual R_f, conjunto de potenciais π
para cada $(i, j) \in A_f$ **faça**
    $c_{ij} \leftarrow c_{ij} + \pi_i - \pi_j$;
    $c_{ji} \leftarrow 0$;// `arco reverso`
fim

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

Síntese

O Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais realiza uma primeira transformação na rede para garantir a existência de apenas um vértice de oferta e um vértice de demanda.

Este artifício é particularmente útil nos casos em que as capacidades dos arcos é maior do que a oferta/demanda – não haverá caminho a partir de s quando o fluxo chegar ao limite da oferta.

O algoritmo de Bellman-Ford é utilizado para determinar o conjunto de potenciais dos vértices π, necessário para reduzir os custos dos arcos.

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

Síntese

Enquanto houver caminho entre s e t na rede residual:

- Determina-se o caminho mais curto P na rede residual, de acordo com os custos dos arcos, utilizando o algoritmo de Dijkstra;
- Os potenciais são atualizados com os valores dos caminhos determinados no passo anterior;
- Os custos reduzidos dos arcos são atualizados de acordo com os potenciais atualizados;
- O fluxo nos arcos do caminho P é aumentado de acordo com a menor capacidade de um arco no caminho;
- A rede residual é atualizada de acordo com o aumento de fluxo realizado.

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

Entrada: Rede R

Transforme a rede R pela adição de vértices e arcos artificiais;

// O fluxo inicial é zero

Construa R_f a partir de R;

$\pi \leftarrow$**Bellman_Ford**(R_f);

Custo_Reduzido(π);

enquanto R_f *contiver um caminho entre s e t* **faça**

  Determine o caminho mais curto P entre s e t;

  Atualize os potenciais π;

  Custo_Reduzido(π);

  Aumente o fluxo em P;

  Atualize a rede residual R_f;

fim

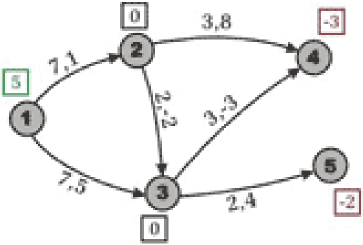
Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

Complexidade

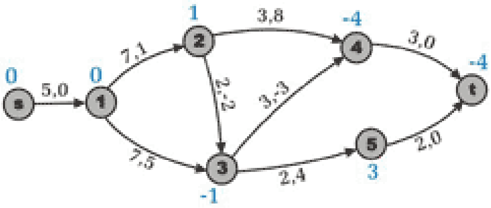
O algoritmo de Bellman-Ford, utilizado apenas uma vez para calcular os potenciais iniciais pode ser implementado em $O(nm)$.

O algoritmo de Dijkstra, utilizado para determinar os caminhos mais curtos sucessivos e atualizar os potenciais, que possui complexidade $O(mlogn)$ usando *heaps*, é executado $O(nB_{max})$ vezes, em que B_{max} é o limitante superior para oferta/demanda em qualquer vértice.

No total, temos complexidade $O(nB_{max}mlogn)$.

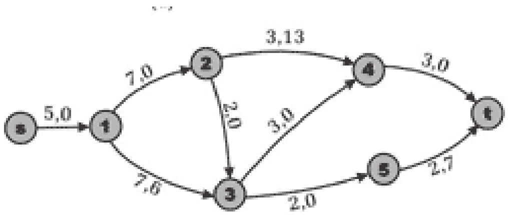


Rede original. Os rótulos indicam capacidade e custo (u_{ij}, c_{ij}).



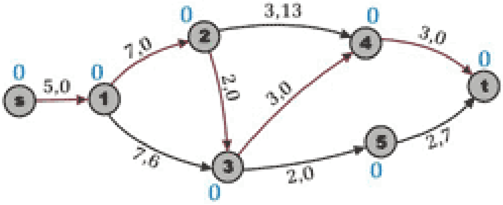
Rede transformada e com potenciais dos vértices calculados.

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais



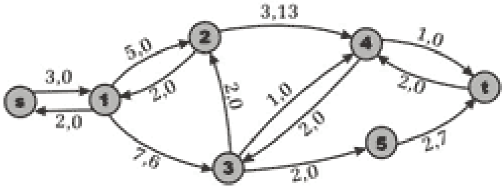
Custos reduzidos nos arcos. Os rótulos indicam capacidade e custo (u_{ij}, c_{ij}).

$$\pi = [0, 0, 1, -1, -4, 3, -4]$$

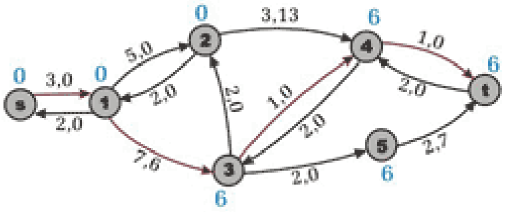


Primeiro caminho, $s - 1 - 2 - 3 - 4 - t$ com gargalo 2.
Os potenciais são atualizados.

Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais

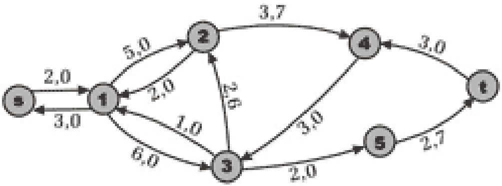


Rede residual com custos reduzidos.



Segundo caminho $s - 1 - 3 - 4 - t$, com capacidade 1.
Os potenciais são atualizados.

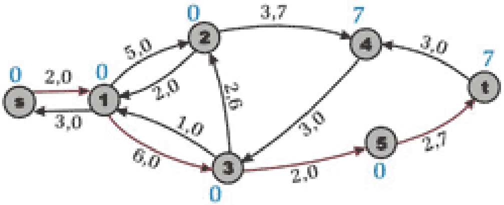
Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais



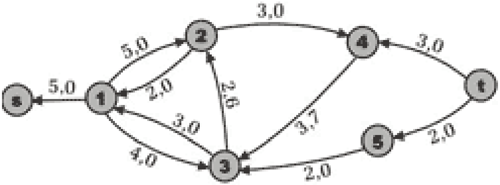
Rede residual com custos reduzidos.

Note que após a atualização dos custos reduzidos, os arcos do caminho mais curto têm custo 0, assim como os arcos reversos.

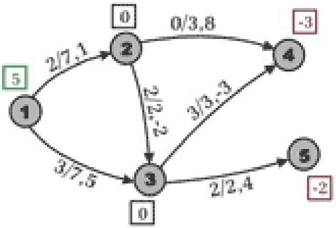
Algoritmo de Caminhos Mais Curtos Sucessivos com Potenciais



Terceiro caminho $s - 1 - 3 - 5 - t$ e novos potenciais.



Não há mais caminhos entre s e t na rede residual.



Rede original e fluxo de custo ótimo, igual a 12.

Dúvidas?