PCC173/BCC463 - Otimização em Redes

Marco Antonio M. Carvalho

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

UFOP
Universidade Federal de Ouro Preto

decom
departamento de computação

Conteúdo

Aviso

Fonte

Este material é baseado nos livros

- Goldbarg, M., & Goldbarg, E. (2012). *Grafos: conceitos, algoritmos e aplicações*. Elsevier.
- Goldbarg, M. C., & Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier.

Licença

Caracterização

Uma classe de algoritmos de fluxo máximo caracteriza-se por forçar o fluxo através da identificação de gargalos (ou **bloqueios**) de fluxo em uma rede em **camadas**.

Serão apresentados neste tópico os algoritmos de **Dinitz** e **MPM**.

Histórico

Proposto originalmente pelo soviético Yefim Dinitz em 1970, em resposta a um exercício proposto por George Adelson-Velsky, co-inventor das árvores AVL (e considerado um dos pais da computação).

Em um artigo, o autor alega que não conhecia o algoritmo de Ford&Fulkerson quando inventou o algoritmo:

"*Ignorance sometimes has its merits. Very probably, DA would not have been invented then, if the idea of possible saturated edge desaturation had been known to the author.*"

Posteriormente, em 1975, uma modificação do algoritmo foi publicada pelo israelense Shimon Even, creditando a versão original a um certo "*Dinic*".

Algoritmo de Dinitz

Histórico

A modificação foi motivada por Shimon Even não ter compreendido completamente a descrição original do algoritmo, cujo artigo tinha apenas 4 páginas e era escrito em russo.

Na ausência da compreensão, Shimon Even "completou" à sua maneira a parte do algoritmo que faltava.

Quase vinte anos depois, Dinitz teve a oportunidade de explicar a Shimon Even a tal parte do algoritmo – e de corrigir o nome!

Algoritmo de Dinitz

Princípio

O algoritmo de Dinitz utiliza o conceito de **redes em camadas** na construção dos caminhos de aumento de fluxo.

Uma rede em camadas H é uma rede induzida a partir de uma rede R, em que já circula um dado fluxo F:

- Na camada zero há somente o vértice s da rede R;
- Na camada 1 estão apenas os vértices $y \in Y$ tal que exista um arco utilizável de s para y:
  - A capacidade desses arcos é $\bar{u}(s, y) - f(s, y) + f(y, s)$;

Princípio

- Na camada 2 estão os vértices w, tais que existem arcos utilizáveis dos vértices y da camada 1 para os vértices w:
  - Então são colocados em H arcos dos vértices da camada 1 para os vértices da camada 2 tais que a capacidade destes arcos é dada por $\bar{u}(y, w) - f(y, w) + f(w, y)$;

A construção da rede prossegue até que o vértice t seja incluído na camada L ou até que não existam mais arcos utilizáveis para serem incluídos em H:

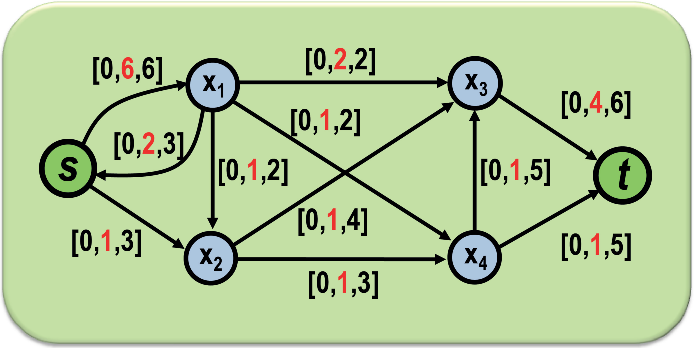
- Caso t não tenha sido incluído não existe uma rede H a ser considerada, situação correspondente a um fluxo máximo circulando na rede R;
- Caso t seja incluído, existe caminho de aumento de fluxo entre os vértices s e t e o fluxo f pode ser aumentado.

Princípio

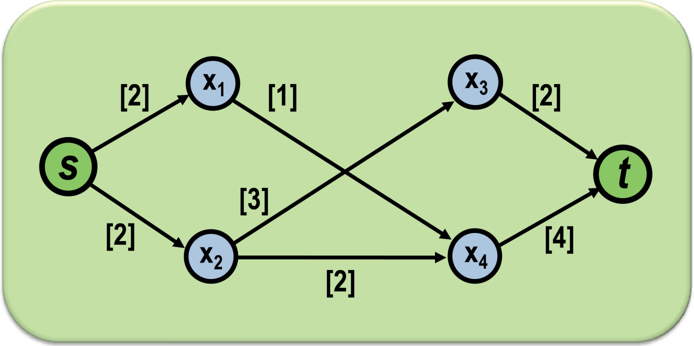
No caso de haver um caminho de aumento de fluxo entre s e t, todos os vértices da camada L diferentes de t são removidos de H.

Examinam-se as camadas k de H, $k = L - 1, \ldots, 1$, removendo-se sucessivamente os vértices que não possuem arcos para a camada $k + 1$.

As figuras a seguir ilustram uma rede em camadas induzida a partir de uma rede R com fluxo viável.



Rede *R* com um fluxo viável.



Rede em camadas.

Os números entre chaves na rede em camadas indicam a capacidade do arco, dada por $\bar{u}(y,w) - f(y,w) + f(w,y)$.

Rede em Camadas

Considerando a figura anterior, na camada 0 só existe o vértice s.

Na camada 1 estão os vértices alcançáveis a partir de s por um arco utilizável: x_1 e x_2.

Embora no arco (s, x_1) esteja passando um fluxo de valor igual a capacidade do arco, no arco (x_1, s) está retornando para s um fluxo de valor 2

- Portanto, $\bar{u}(s, x_1) - f(s, x_1) + f(x_1, s) = 2$;
- Assim, é inserido na rede em camadas um arco (s, x_1) com capacidade 2.

No arco (s, x_2) ainda é possível passar 2 unidades de fluxo, portanto, um arco (s, x_2) com capacidade 2 é inserido na rede em camadas.

Rede em Camadas

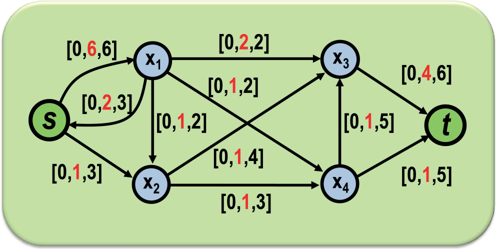
A camada 2 contém os vértices alcançáveis a partir de x_1 e x_2 por um arco utilizável, ou seja, x_3 e x_4.

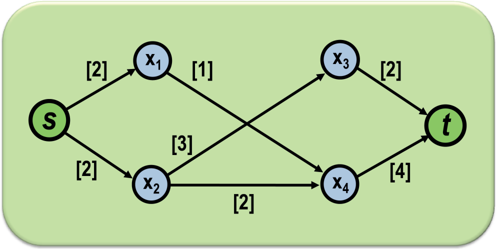
Os arcos (x_1, x_4), (x_2, x_3) e (x_2, x_4) são inseridos na rede em camadas com as capacidades mostradas na figura anterior.

O arco (x_1, x_3) não é inserido na rede em camadas, uma vez que $\bar{u}(x_1, x_3) = f(x_1, x_3)$.

Finalmente, o vértice t é inserido na camada 3.

Algoritmo de Dinitz





Descrição

O algoritmo de Dinitz é composto basicamente por dois laços: um externo e outro interno.

O laço externo constrói novas redes em camadas a partir de um fluxo F.

O laço interno realiza diferentes operações em uma mesma rede em camadas.

Uma vez construída a rede em camadas H, o algoritmo entra no laço mais interno, onde caminhos de aumento de fluxo são encontrado a partir de t.

Em cada um destes caminhos de aumento de fluxo encontrados em H, aumenta-se o fluxo de acordo com o gargalo (ou **bloqueio**).

Descrição

Os caminhos P entre s e t são construídos iterativamente a partir de t:

Suponha que na outra extremidade do arco incidente em t está o vértice v_1, então, a construção do caminho até s continua a partir de v_1.

Esse procedimento é repetido até que um arco (s, v_{l-1}) seja escolhido.

Portanto, após L passos, o caminho P de aumento de fluxo é construído no procedimento **caminhoDeAumento**.

O fluxo F é aumentado na rede no valor da menor capacidade residual de um arco no caminho P pelo procedimento **aumentarFluxo**(P).

Os arcos de P têm suas capacidades atualizadas após isto.

Algoritmo de Dinitz

Descrição

Em cada iteração do laço mais interno, pelo menos uma aresta é saturada – exatamente a(s) relacionada(s) ao(s) gargalo(s).

Os arcos saturados de P são armazenadas no conjunto Sat, pelo procedimento **arcosSaturados**(P).

Este mesmo conjunto é passado como dado de entrada para o procedimento **limpeza( )**, o qual remove todos os arcos saturados, vértices com grau de entrada ou de saída igual a zero (exceto s e t) e outros arcos em que um dos vértices terminais tenha sido removido de H.

Algoritmo de Dinitz

```
Entrada: Rede R com fluxo viável F = 0
Construa a rede em camadas H a partir de R, considerando o fluxo F;
repita
    enquanto existir um caminho entre s e t faça
        P ← caminhoDeAumento(H, s, t);
        F ← aumentarFluxo(P);
        Sat ← arcosSaturados(P);
        H ← limpeza(H, Sat);
    fim
    Construa a rede em camadas H com base na rede residual de R,
     considerando o fluxo F;
até não ser possível construir rede em camadas H entre os vértices s t;
```

Algoritmo de Dinitz

Complexidade

Pode ser mostrado que há, no máximo, $n-1$ gargalos (ou bloqueios) em uma rede em camadas, portanto, o laço externo executa $O(n)$ vezes.

Para cada um deles:

- A construção de uma rede em camadas em cada iteração do laço mais externo pode ser realizada em $O(m)$ por uma BFS;
- Podemos utilizar a DFS para determinarmos os caminhos aumentantes em tempo linear $O(n)$;
- Para cada caminho, são realizadas $O(m)$ limpezas ou aumentos de caminho;
- Resultando em $O(m + nm)$.

Desta forma, o laço interno executa em tempo $O(nm)$, e o externo em tempo $O(n)$. A complexidade geral do algoritmo é, portanto, $O(n^2m)$.

Complexidade

Utilizando árvores dinâmicas, o laço interno tem sua complexidade reduzida para $O(m \; log \; n)$.

Desta forma, a complexidade total se torna $O(mn \; log \; n)$.

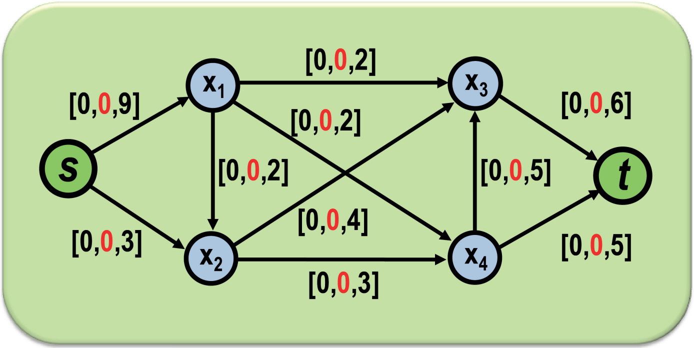
Algoritmo de Dinitz

Contribuição de Even

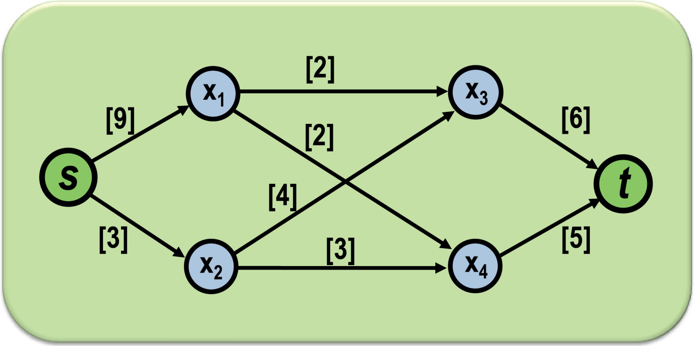
A utilização de BFS e DFS no algoritmo de Dinitz na realidade foi a contribuição de Shimon Even!

A história é relatada em detalhes pelo próprio Yefim Dinitz em um artigo de 2006, dois anos após o falecimento de Shimon Even.

Além disto, o artigo contém a contextualização da escola de algoritmos Russa durante a cortina de ferro e as descrições do algoritmo original e a versão de Even por Yefim Dinitz, invertendo os papéis históricos.

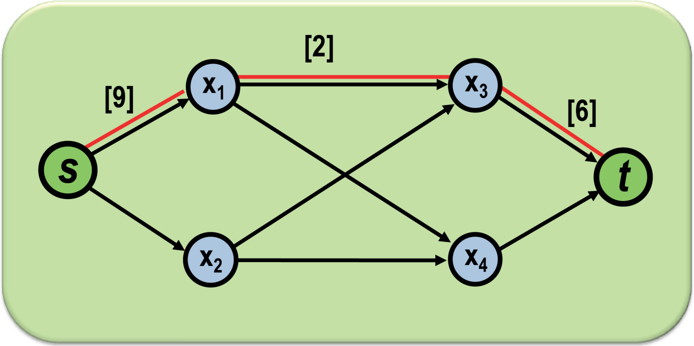


Rede R com um fluxo viável $f = 0$.

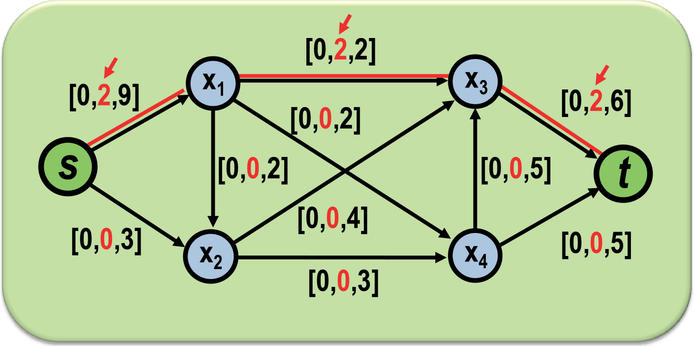


Primeira rede em camadas.

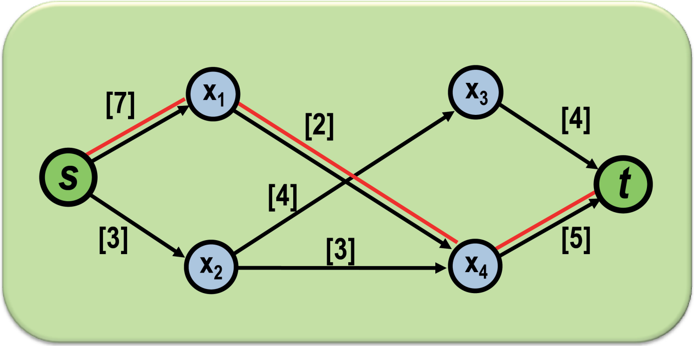
Algoritmo de Dinitz



O caminho de aumento de fluxo $P = s - x_1 - x_3 - t$ é escolhido pelo algoritmo e possui gargalo (ou bloqueio) 2.

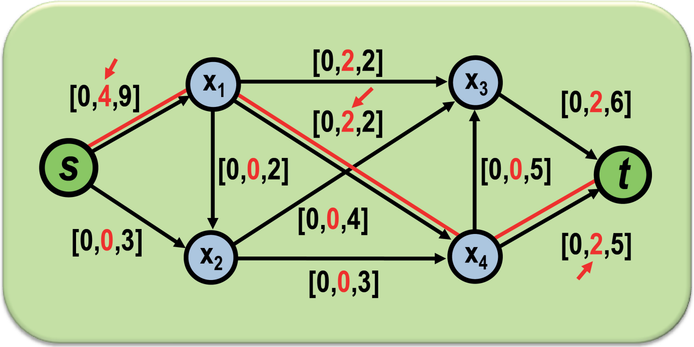


Primeiro aumento de fluxo em R (2 unidades).



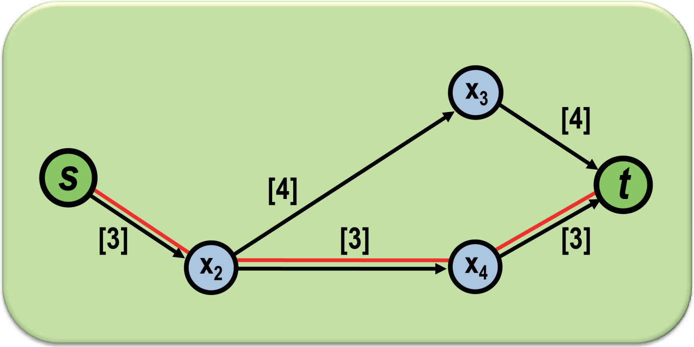
No procedimento de limpeza o arco saturado (x_1, x_3) é removido.
Os arcos (s, x_1) e (x_3, t) tiveram suas capacidades atualizadas.
O caminho de aumento de fluxo $P = s - x_1 - x_4 - t$, de gargalo (ou bloqueio) 2 é escolhido.

Algoritmo de Dinitz



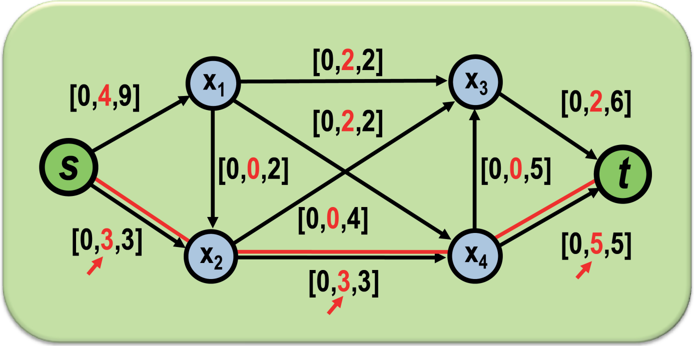
Segundo aumento de fluxo em R (2 unidades).

Algoritmo de Dinitz



O arco (x_1, x_4) é removido de H e, como consequência, o vértice x_1 também é retirado, uma vez que o mesmo não possui arcos de saída.

No caminho de aumento de fluxo $P = s - x_2 - x_4 - t$ todos os arcos possuem capacidade 3, permitindo um aumento de fluxo de 3 unidades.

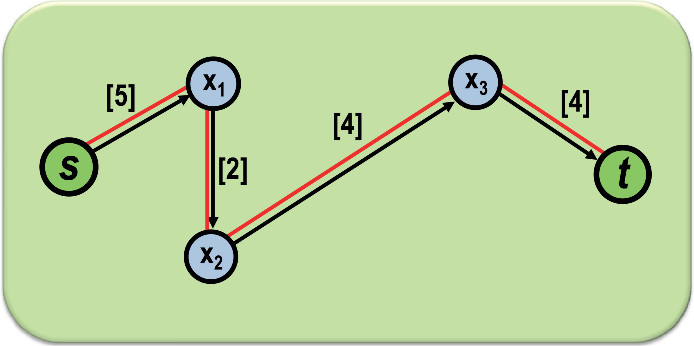


Terceiro aumento de fluxo em R (3 unidades).

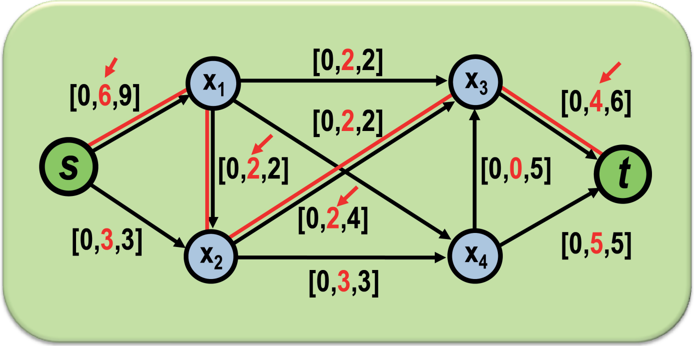
Exemplo

A rede atualizada após a última operação de aumento de fluxo não possui caminho entre s e t, então o laço mais interno termina e o algoritmo constrói uma nova rede em camadas H com base no fluxo circulante, que é mostrada na figura a seguir.

Algoritmo de Dinitz



Único caminho de aumento de fluxo entre s e t: $P = s - x_1 - x_2 - x_3 - t$, com gargalo (ou bloqueio) 2.



Quarto aumento de fluxo em R (2 unidades).

Algoritmo de Dinitz

Exemplo

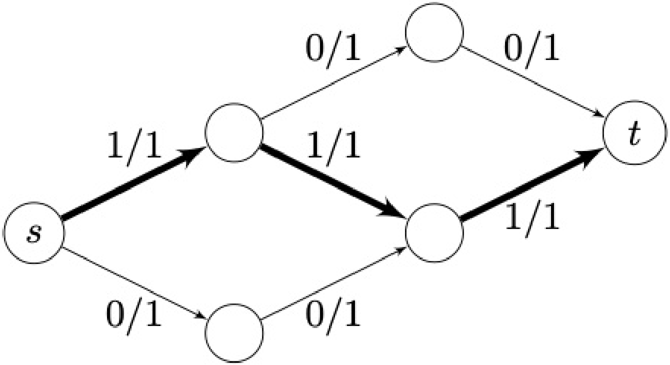
Após o aumento de fluxo anterior, a rede H é desconectada e o laço mais interno termina.

O algoritmo então inicia a construção de uma nova rede em camadas: o vértice s é incluído na camada 0 e o vértice x_1 na camada 1, o único com um arco utilizável a partir de s.

Entretanto, todos os arcos a partir de x_1 estão saturados e uma rede em camadas entre s e t não pode ser construída.

Portanto, o algoritmo termina mostrando que o fluxo máximo nesta rede tem valor 9.

Algoritmo de Dinitz



Nem todo fluxo blocante é máximo. Todo fluxo máximo é blocante.

Fonte: Kyng,R. *Advanced Graph Algorithms and Optimization. Lecture Notes: Classical Algorithms for Maximum Flow II*. ETH Zürich. 2020.

Leitura Recomendada

Yefim Dinitz (2006). "Dinitz' Algorithm: The Original Version and Even's Version". In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman. Theoretical Computer Science: Essays in Memory of Shimon Even. Springer. pp. 218–240. Disponível em: http://www.cs.bgu.ac.il/~dinitz/Papers/Dinitz_alg.pdf.

Dúvidas?