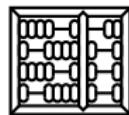


Introdução a Programação Linear

Flávio Keidi Miyazawa



UNIVERSIDADE DE CAMPINAS - UNICAMP
INSTITUTO DE COMPUTAÇÃO - IC



Campinas, 2002-2016

Sumário I

- 1 Introdução
- 2 Resolvedores de Programação Linear
- 3 Exemplos
 - Problema da Dieta
 - Otimização de Portfolio
- 4 Definições
- 5 Matrizes Totalmente Unimodulares
- 6 Fluxos, Caminhos e Cortes
 - Fluxo de Custo Mínimo
 - st -Fluxo Máximo
 - st -Corte Mínimo
 - Caminho Mínimo
- 7 Emparelhamento em Grafos Bipartidos

Programação Linear

Programação linear

- dá a resolução exata para muitos problemas
- dá a resolução aproximada para muitos problemas
- faz parte dos principais métodos para obter soluções ótimas
- faz parte dos principais métodos para obter soluções aproximadas
- dá excelentes delimitantes para soluções ótimas
- pode ser executada muito rapidamente
- há diversos programas livres e comerciais

Obs.: Algumas teorias sobre a estrutura dos programas lineares serão apresentadas para se entender os resultados das reduções, mas o foco da aula será nas reduções para Programas Lineares.

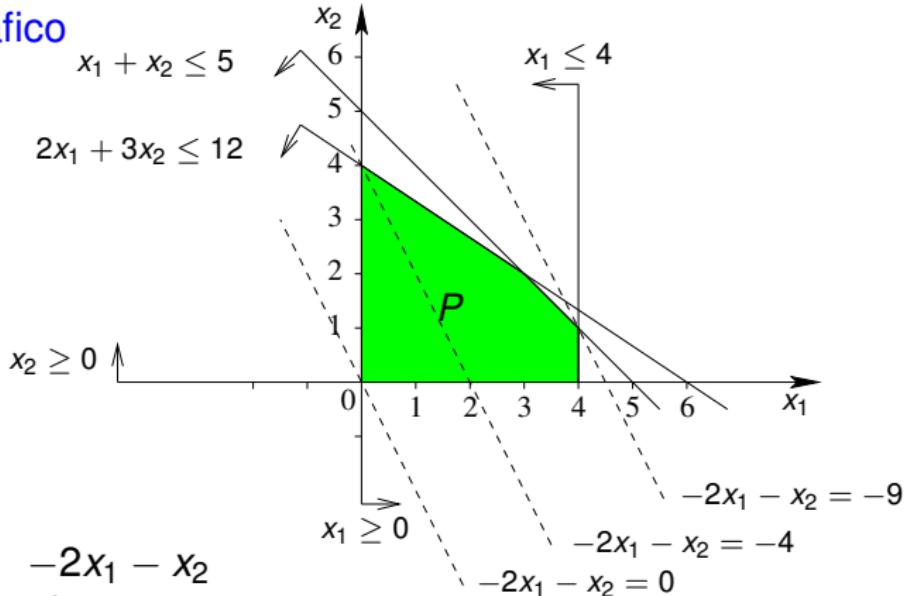
Programação Linear

Problema PL: Dados matriz $A = (a_{ij}) \in \mathbb{Q}^{n \times m}$, vetores $c = (c_i) \in \mathbb{Q}^n$ e $b = (b_i) \in \mathbb{Q}^m$, encontrar vetor $x = (x_i) \in \mathbb{Q}^m$ (se existir) que

$$\begin{array}{ll}\text{minimize} & c_1x_1 + c_2x_2 + \cdots + c_mx_m \\ \text{sujeito a} & \left\{ \begin{array}{lcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_n & \leq & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_n & \geq & b_2 \\ & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_n & = & b_n \\ x_i \in \mathbb{Q} \end{array} \right. \end{array}$$

Teorema: *PL pode ser resolvido em tempo polinomial.*

Exemplo gráfico



minimize $-2x_1 - x_2$

sujeito a

$$\begin{cases} x_1 + x_2 \leq 5 \\ 2x_1 + 3x_2 \leq 12 \\ x_1 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

Sol. ótima: $x_1 = 4$ e $x_2 = 1$

Valor da sol. ótima = -9

Teorema: *Uma solução que é vértice do PL pode ser encontrada em tempo polinomial.*

Algoritmos polinomiais para resolver PL:

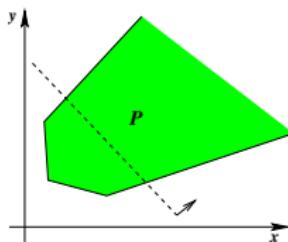
- Algoritmo dos elipsóides (Khachiyan'79) e
- Método dos pontos interiores (Karmarkar'84).

Algoritmos exponenciais para resolver PL:

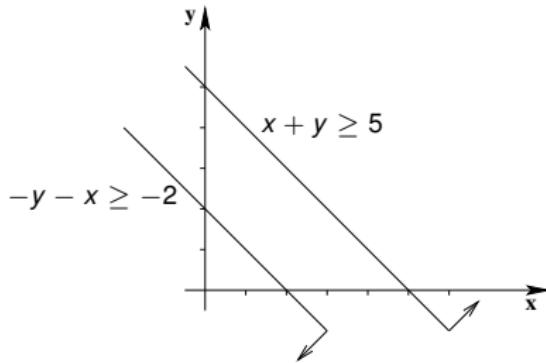
- Método simplex (Dantzig'47) (tempo médio polinomial).

Nem sempre encontramos uma solução ótima

- Sistema ilimitado



- Sistema inviável



Resolvedores de Sistemas Lineares

Programas comerciais

- CPLEX / IBM: www.ilog.com/products/cplex/
- XPRESS: www.dashoptimization.com/
- GUROBI: <http://www.gurobi.com/>

Programas livres

- SCIP: <http://scip.zib.de/>
- HiGHS: <https://highs.dev/>
- COIN-CLP: www.coin-or.org/Clp/
- SoPlex / Zib: www.zib.de/Optimization/Software/Soplex
- GLPK / Gnu: www.gnu.org/software/glpk/glpk.html
- LEMON / Coin-OR: <http://www.coin-or.org/>
Biblioteca de grafos usando solver GLPK ou COIN-CLP

Problema da Dieta

São dados:

- m alimentos: ali_1, \dots, ali_m
- n nutrientes: nut_1, \dots, nut_n
- preço p_j de cada alimento ali_j
- recomendações diárias r_i de cada nutriente nut_i , para uma dieta balanceada
- quantidade q_{ij} do nutriente nut_i no alimento ali_j ,

Objetivo: Encontrar uma dieta mais barata respeitando as recomendações nutricionais

Exemplo do Problema da Dieta I

Nutrientes, com recomendações diárias mínima e máxima

Nutriente	mínimo	máxima
nut_1 = cálcio	800	1200
nut_2 = ferro	10	18

Alimentos, com preço e composição nutricional:

Alimento	preço	cálcio	ferro
ali_1 = carne de boi (kg)	20.0	110.00	29.00
ali_2 = feijão cozido (kg)	6.0	170.00	15.00
ali_3 = leite desnatado (lt)	2.0	1150.00	0.00

Exemplo do Problema da Dieta I

Formulação Linear

Variáveis

- x_1 quantidade de *carne de boi* a comprar
- x_2 quantidade de *feijão cozido* a comprar e
- x_3 quantidade de *leite desnatado* a comprar.

Função objetivo

- Minimizar o preço pago por todos os alimentos. I.e.,

$$\text{minimize } 20,0x_1 + 6,0x_2 + 2,0x_3$$

Restrições

- quantidades x_1 , x_2 e x_3 devem ser valores válidos e satisfazer restrições nutricionais

Exemplo do Problema da Dieta I

Restrições nutricionais

- Quantidade de cálcio consumido dentro das recomendações diárias

$$110x_1 + 170x_2 + 1150x_3 \geq 800$$

$$110x_1 + 170x_2 + 1150x_3 \leq 1200$$

- Quantidade de ferro consumido dentro das recomendações diárias

$$29x_1 + 15x_2 \geq 10$$

$$29x_1 + 15x_2 \leq 18$$

Restrições de não negatividade

- Nenhuma quantidade pode ser negativa

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0$$

Exemplo do Problema da Dieta I

Formulação Linear

$$\begin{array}{ll}\text{minimize} & 20,0x_1 + 6,0x_2 + 2,0x_3 \\ \text{sujeito a} & \left\{ \begin{array}{lcl} 110x_1 + 170x_2 + 1150x_3 & \geq & 800 \\ 110x_1 + 170x_2 + 1150x_3 & \leq & 1200 \\ 29x_1 + 15x_2 & \leq & 10 \\ 29x_1 + 15x_2 & \leq & 18 \\ x_1 \geq 0 & x_2 \geq 0 & x_3 \geq 0 \end{array} \right. \end{array}$$

Resolvendo-se:

Obtemos uma dieta de 5,194 reais com

$x_1 = 0$ kg. de carne de boi,

$x_2 = 0,666$ kg. de feijão cozido e

$x_3 = 0,597$ lt. de leite desnatado.

A quantidade diária de cálcio nesta dieta é 800 e de ferro é 10.

Trecho de implementação usando LEMON

```
# Programa em Julia, usando JuMP e Gurobi
import Pkg
Pkg.build("Gurobi")                      # Pkg.build("HiGHS")
Pkg.add("Gurobi")                         # Pkg.add("HiGHS")
using JuMP, Gurobi                         # using JuMP, HiGHS
model = Model(Gurobi.Optimizer)            # model = Model(HiGHS.Optimizer)

@variable(model, x1 >= 0 )
@variable(model, x2 >= 0 )
@variable(model, x3 >= 0 )

@objective(model, Min, 20.0*x1 + 6.0*x2 + 2.0*x3 )

@constraint(model, 800 <= 110*x1 + 170*x2 + 1150*x3 <= 1200 )
@constraint(model, 10 <= 29*x1 + 15*x2           <= 18 )

optimize!(model)

println("Optimal objective: ", objective_value(model), ". "
       "x1=", value(x1), " x2=", value(x2), " x3=", value(x3))
```

Exemplo do Problema da Dieta II

Nutrientes, com recomendações diárias mínima e máxima

Nutriente	mínimo	máxima
nut_1 = fósforo	800	1200
nut_2 = vitamina C	60	90

Alimentos, com preço e composição nutricional:

Alimento	preço	fósforo	vitamina C
ali_1 = carne de boi (kg)	20.0	1800.00	0.00
ali_2 = feijão cozido (kg)	6.0	490.00	10.00

Variáveis

- x_1 quantidade de *carne de boi* a comprar
- x_2 quantidade de *feijão cozido* a comprar

Exemplo do Problema da Dieta II

Formulação Linear

$$\begin{array}{ll} \text{minimize} & 20,0x_1 + 6,0x_2 \\ \text{sujeito a} & \left\{ \begin{array}{l} 1800x_1 + 490x_2 \geq 800 \\ 1800x_1 + 490x_2 \leq 1200 \\ 10x_2 \geq 60 \\ 10x_2 \leq 90 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{array} \right. \end{array}$$

Este sistema é inviável:

- O único a ter vitamina C é o feijão (e bem pouco)
- Por $10x_2 \geq 60$ temos que $x_2 \geq 6$
I.e., para suprir necessidade de vitamina C, precisamos ingerir pelo menos 6 kg. de feijão por dia!!
- 6kg. de feijão contém $6 * 490 = 2940$ mg de fósforo acima do limite diário permitido de 1200mg de fósforo.

Exemplo de Otimização de Portfolio

- Temos 100.000 reais para investir em ações
- As ações selecionadas e a porcentagem de retorno esperado em 1 ano são:

Empresa	Retorno (em %)
emp_1 = Petrobrás (petróleo/estatal)	9.0%
emp_2 = Vale do Rio Doce (siderurgia)	10.2%
emp_3 = Votorantim (siderurgia)	6.5%
emp_4 = Texaco (petróleo)	9.5%
emp_5 = Sanasa (água/estatal)	8.5%

Exemplo de Otimização de Portfolio

A recomendação dos especialistas é a seguinte:

- Recomenda se investir pelo menos 25% e no máximo 55% em empresas estatais.
- Petrobrás e Texaco são empresas do mesmo setor (petróleo). Recomenda-se que o investimento nas duas não passe de 55 %.
- Vale do Rio Doce e Votorantim são do mesmo setor (siderurgia) recomenda que o investimento nas duas não passe de 45 %.
- Apesar da Vale do Rio Doce ter a maior taxa de retorno, há boatos que ela pode estar maquiando faturamento. Recomenda-se que a quantidade de investimento nela não passe de 60% do total de investimento feito em empresas de siderurgia.

Exemplo de Otimização de Portfolio

Formulação Linear

Variáveis

- x_1 quantidade de investimento na *Petrobrás*
- x_2 quantidade de investimento na *Vale do Rio Doce*
- x_3 quantidade de investimento na *Votorantim*
- x_4 quantidade de investimento na *Texaco*
- x_5 quantidade de investimento na *Sanasa*

Função objetivo

- Maximizar o lucro esperado,

$$\text{maximize } 0,090x_1 + 0,102x_2 + 0,065x_3 + 0,095x_4 + 0,085x_5$$

Restrições

- quantidades x_1, \dots, x_5 devem ser valores válidos e devem satisfazer recomendações dos especialistas

Exemplo de Otimização de Portfolio

Restrições impostas por especialistas

- Recomenda se investir pelo menos 25% e no máximo 55% em empresas estatais.

$$x_1 + x_5 \geq 25000$$

$$x_1 + x_5 \leq 55000$$

- Petrobrás e Texaco são empresas do mesmo setor (petróleo). Recomenda-se que o investimento nas duas não passe de 55 %.

$$x_1 + x_4 \leq 55000$$

- Vale do Rio Doce e Votorantim são do mesmo setor (siderurgia) recomenda que o investimento nas duas não passe de 45 %.

$$x_2 + x_3 \leq 45000$$

Exemplo de Otimização de Portfolio

- Apesar da Vale do Rio Doce ter a maior taxa de retorno, há boatos que ela pode estar maquiando faturamento. Recomenda-se que a quantidade de investimento nela não passe de 60% do total de investimento feito em empresas de siderurgia.

$$x_2 \leq 0.6(x_2 + x_3)$$

$$\text{i.e., } -0.4x_2 + 0.6x_3 \geq 0$$

Demais Restrições

- Total investido é 100.000

$$x_1 + x_2 + x_3 + x_4 + x_5 = 100000$$

- Nenhuma quantidade pode ser negativa

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0$$

Exemplo de Otimização de Portfolio

Formulação Linear

$$\begin{array}{ll} \text{maximize} & 0,090x_1 + 0,102x_2 + 0,065x_3 + 0,095x_4 + 0,085x_5 \\ \text{sujeito a} & \left\{ \begin{array}{l} x_1 + x_2 + x_3 + x_4 + x_5 = 100000 \\ x_1 + x_5 \geq 25000 \\ x_1 + x_5 \leq 55000 \\ x_1 + x_4 \leq 55000 \\ x_2 + x_3 \leq 45000 \\ -.4x_2 + .6x_3 \geq 0 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \quad x_5 \geq 0 \end{array} \right. \end{array}$$

Resolvendo-se:

Obtemos uma lucro estimado de 9094 reais investindo

$$x_1 = 0 \text{ na Petrobrás}, \quad x_2 = 12000 \text{ na Vale do Rio Doce},$$

$$x_3 = 8000 \text{ na Votorantim}, \quad x_4 = 55000 \text{ na Texaco e}$$

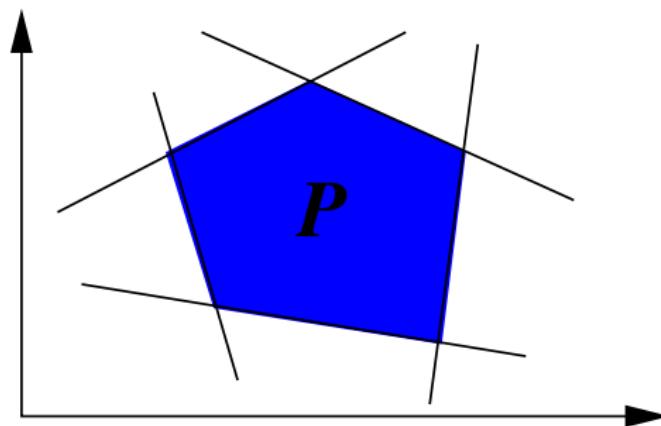
$$x_5 = 25000 \text{ na Sanasa.}$$

ALGUMAS DEFINIÇÕES

Def.: Chamamos o conjunto de pontos P ,

$$P := \left\{ x \in \mathbb{Q}^n : \begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_n & \leq & b_1 \\ \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_n & \geq & b_n \end{array} \right\}$$

como sendo um **poliedro**.



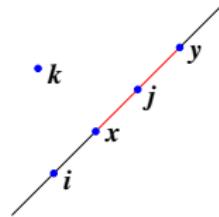
Def.: Dado conjunto de pontos S , dizemos que y é uma **combinação convexa** dos pontos de S se $y = \alpha_1y_1 + \alpha_2y_2 + \dots + \alpha_ny_n$, onde $y_i \in S$, $\alpha_i \geq 0$ e $\sum_i \alpha_i = 1$

Exemplo: Os pontos que são combinação convexa de dois pontos x e y são:

$$\text{conv}(\{x, y\}) := \{\alpha_1x + \alpha_2y : \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1\}$$

substituindo $\alpha_2 = 1 - \alpha_1$, temos

$$\text{conv}(\{x, y\}) := \{\alpha x + (1 - \alpha)y : 0 \leq \alpha \leq 1\}$$



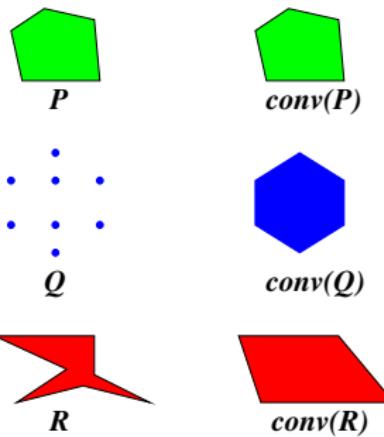
j é combinação convexa de x e y , mas i e k não são.

Exercício: Mostre que se $x, y \in \text{conv}(S)$ e $z \in \text{conv}(x, y)$, então temos também que $z \in \text{conv}(S)$.

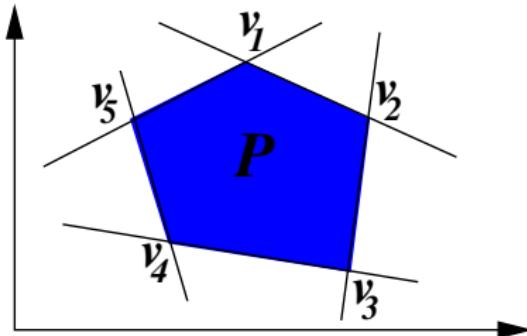
Def.: O *fecho convexo* de S , denotado por $\text{conv}(S)$ é o conjunto dos pontos que são combinação convexa dos pontos de S .

Exemplo: Dado dois pontos $x, y \in \mathbb{R}^n$, qualquer ponto que esteja no segmento de reta que liga x a y está em $\text{conv}(\{x, y\})$.

Exemplo:



Def.: Os vértices ou pontos extremais de P são os pontos de P que não podem ser escritos como combinação convexa de outros pontos de P



Vértices de P : v_1, v_2, v_3, v_4, v_5

Note que um vértice do poliedro é um ponto único que satisfaz um determinado conjunto de restrições na igualdade.

Determinantes e Resolução de Sistemas Lineares

Determinante por Laplace

Dada matriz A de ordem n e coluna j ,

$$|A| = \sum_{i=1}^n a_{ij} \cdot (-1)^{i+j} |A^{i,j}|,$$

onde $A^{i,j}$ é a matriz A removendo a linha i e coluna j

Resolução de Sistemas Lineares por Cramer

Dado matriz quadrada A e vetor b , encontrar x tal que $Ax = b$:

$$x_i = \frac{|A_b^i|}{|A|},$$

onde A_b^i é a matriz A com a coluna j trocada pelo vetor b .

Def.: Submatriz de A é uma matriz obtida de A removendo linhas e colunas.

Teorema: *Todo vértice y de um poliedro, definido por uma matriz A , é determinado por uma submatriz não singular A' de A tal que $A' \cdot y = b'$, onde b' contém os correspondentes elementos de b das linhas de A' .*

Def.: Se todo vértice de um poliedro tem apenas componentes inteiras, então o poliedro é dito ser inteiro.

Corolário: *Se P é um poliedro definido por matriz A e vetor b (e.g., $Ax \leq b$), tal que toda submatriz de A tem determinante em $\{-1, 0, +1\}$ e b é inteiro, então todo vértice do poliedro é inteiro.*

Def.: Uma matriz A é dita ser totalmente unimodular (TU) se para qualquer submatriz quadrada A' de A (selecionando linhas e colunas de A) temos $\det(A') \in \{-1, 0, +1\}$.

Teorema: Se A é TU, então para todo vetor inteiro b temos que $P = \{x : Ax \leq b\}$ é inteiro.

Lema: Se A e B são matrizes TU, então as seguintes matrizes também são TU:

- 1 Matriz obtida de A removendo uma linha (coluna)
- 2 Matriz obtida de A duplicando uma linha (coluna)
- 3 Matriz obtida de A trocando duas linhas (colunas)
- 4 A^T e $-A$
- 5 $(A | I)$ e $\begin{pmatrix} A \\ I \end{pmatrix}$
- 6 $(A | -A)$

Prova. Itens 1., 2., e 3., seguem da propriedade de determinante e de matriz TU.

4. Determinante da transposta é igual ao da original e multiplicar linha por -1 apenas troca sinal do determinante.
5. Basta aplicar regra de Laplace nos elementos da submatriz de I .
6. Duplicar colunas e trocar sinal da coluna mantém TU. □

Teorema: Se A é TU e b e u são vetores inteiros, então os seguintes poliedros são inteiros:

- ① $\{x : Ax \geq b\}$
- ② $\{x : Ax \leq b, x \geq 0\}$
- ③ $\{x : Ax = b, x \geq 0\}$
- ④ $\{x : Ax = b, 0 \leq x \leq u\}$

Prova. Note que o poliedro $\{x : Ax = b\}$ é dado por $\{x : Ax \leq b, Ax \geq b\}$.

No item 4., basta ver que a matriz
$$\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix}$$
 é TU. □

Lema: Se A é a matriz de incidência de grafo direcionado, então A é TU.

Prova. Por indução na ordem da submatriz. A base é direta.

Seja B uma submatriz quadrada de A .

- ① Se há uma coluna z só com elementos nulos: Neste caso o determinante de B é 0.
- ② Se há coluna z com exatamente um elemento não nulo $t \in \{-1, +1\}$: Neste caso, o determinante é $+t$ ou $-t$ vezes a submatriz removendo a linha e coluna de t (regra de Laplace).
- ③ Caso contrário, toda coluna tem -1 e $+1$. Como a soma de todas as linhas é nula, o determinante é nulo.

□

De maneira geral, vale que:

Lema: Se A é matriz formada por elementos em $\{-1, 0, 1\}$, e cada coluna tem no máximo uma ocorrência de -1 e no máximo uma ocorrência de 1 , então A é TU.

Prova. Exercício.

□

Lema: Se A é a matriz de incidência de um grafo bipartido não-direcionado $G = (X, Y, E)$, então A é TU.

Prova. Multiplique por -1 todas as linhas de A correspondentes aos vértices de Y . Note que isto só muda o sinal dos subdeterminantes.

Isto nos dá uma matriz de incidência de um grafo direcionado e a prova segue pelo lema anterior. □

Exercícios:

- Mostre que se A é a matriz de incidência de um grafo direcionado e B uma matriz obtida a partir de A transformando alguns elementos não nulos em 0, então B também é TU.
- Seja $G = (N, A)$ um grafo direcionado com conjunto de nós N e arcos A e dois nós distintos s e t . Apresente um poliedro definido por variável $x \in [0, 1]^{|A|}$ indexada em A , com número polinomial de restrições, e conjunto de vértices \mathcal{V} , tal que, se $\hat{x} \in \mathcal{V}$, então \hat{x} representa os caminhos de s a t , com $\hat{x}_a = 0$ para todo $a \in \delta^-(s) \cup \delta^+(t)$. Isto é, são caminhos sem arcos entrando em s ou saindo de t .

Fluxo de Custo Mínimo

Considere um sistema de produção de bens onde há centros consumidores e produtores.

- Todos os bens produzidos devem ser todos consumidos.
- Os bens produzidos chegam até os consumidores através de rotas.
- Cada rota tem uma capacidade máxima de escoamento (direcionada).
- Cada rota tem seu custo para transportar cada unidade do bem.
- **Objetivo:** Transportar os bens dos produtores para os consumidores minimizando custo para transportar os bens.

Aplicações importantes na Computação:

- Transferência de dados em rede de computadores, deteção de “gargalos” da rede na transferência.
- Projeto de vias de tráfego no planejamento urbano.
- Deteção da capacidade de transmissão entre pontos de redes de telecomunicações.

Notação:

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

- Dado $S \subseteq V$, denotamos por $\delta(S)$ o conjunto de arestas com um extremo em S e outro em $V \setminus S$.
- Dado $v \in V$, denotamos por $\delta(v)$ o conjunto $\delta(\{v\})$.

Seja $D = (V, E)$ um grafo direcionado.

- Dado $S \subseteq V$, denotamos por $\delta^+(S)$ o conjunto de arcos com início em S e fim (ponta) em $V \setminus S$.
- Dado $v \in V$, denotamos por $\delta^+(v)$ o conjunto $\delta^+(\{v\})$.
- Dado $S \subseteq V$, denotamos por $\delta^-(S)$ o conjunto de arcos com fim (ponta) em S e início em $V \setminus S$.
- Dado $v \in V$, denotamos por $\delta^-(v)$ o conjunto $\delta^-(\{v\})$.

Definições

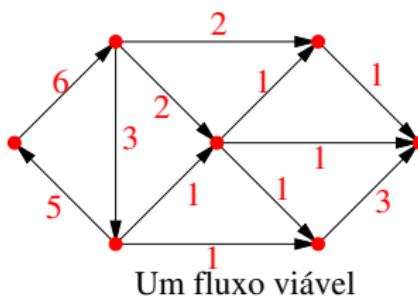
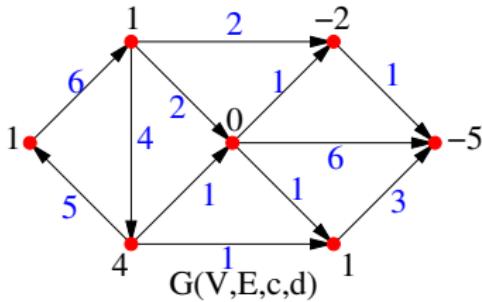
Seja $G = (V, E)$ um grafo direcionado com função de capacidades nas arestas $c : E \rightarrow \mathbb{Q}^+$, demandas $b : V \rightarrow \mathbb{Q}$ e custos nas arestas $w : E \rightarrow \mathbb{Q}^+$.

Def.: Dado um vértice $i \in V$, se $b_i < 0$ dizemos que i é um *consumidor* e se $b_i > 0$ dizemos que i é um *produtor*.

Def.: Dizemos que $x : E \rightarrow \mathbb{Q}^+$ é um *fluxo* em G se

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = b_v, \quad \forall v \in V, \quad e \quad 0 \leq x_e \leq c_e, \quad \forall e \in E$$

Ex.:



Def.: Dado um fluxo $x : E \rightarrow \mathbb{Q}^+$ (respeitando c e b) definimos o **custo do fluxo** x como sendo $\sum_{e \in E} w_e x_e$.

Problema FLUXO DE CUSTO MÍNIMO: Dados um grafo direcionado $G = (V, E)$, capacidades $c : E \rightarrow \mathbb{Q}^+$, demandas $b : V \rightarrow \mathbb{Q}^+$ e uma função de custo nas arestas $w : E \rightarrow \mathbb{Q}^+$, encontrar um fluxo $x : E \rightarrow \mathbb{Q}^+$ de custo mínimo.

Encontrar x tal que

$$\begin{aligned} & \min \sum_{e \in E} w_e x_e \\ & \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = b_v \quad \forall v \in V \\ & 0 \leq x_e \leq c_e \quad \forall e \in E \end{aligned}$$

Teorema: *Se as capacidades nas arestas e as demandas dos vértices são inteiros (i.e., $c : E \rightarrow \mathbb{Z}^+$ e $b : V \rightarrow \mathbb{Z}^+$) então os vértices do poliedro do fluxo são inteiros.*

Prova. Segue do fato que a matriz de incidência de um grafo direcionado é TU. □

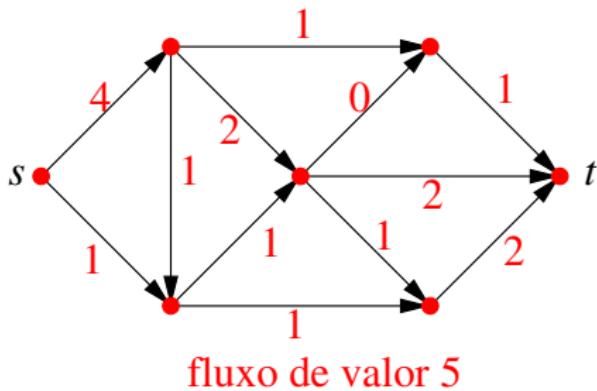
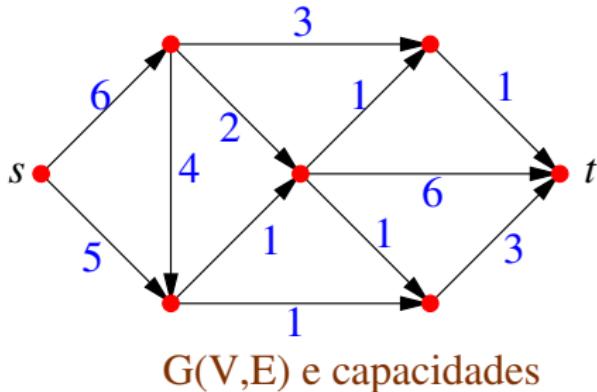
Subproblemas

- Fluxo máximo de um vértice s a um vértice t (st -fluxo)
- Problema do corte de capacidade mínima
- Problema do caminho mínimo (com pesos não negativos)
- Emparelhamento de peso máximo em grafos bipartidos

Fluxo Máximo de s para t (st -fluxo máximo)

- Não temos custo para transportar os bens.
- Cada rota tem uma capacidade máxima de escoamento (direcionada).
- Só temos um produtor (vértice s) de produção arbitrariamente grande.
- Só temos um consumidor (vértice t) de consumo arbitrariamente grande.
- Nós internos apenas repassam bens.
- **Objetivo:** Maximizar o transporte de bens de s para t , respeitando restrições de capacidade do fluxo.

Exemplo:



Seja $G = (V, E)$ um grafo direcionado, capacidades $c : E \rightarrow \mathbb{Q}^+$ e vértices s e t .

Def.: Dizemos que $x : E \rightarrow \mathbb{Q}^+$ é um st -fluxo se

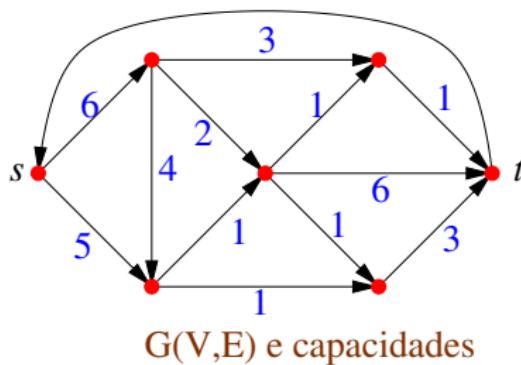
$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \quad \forall v \in V \setminus \{s, t\}$$
$$0 \leq x_e \leq c_e \quad \forall e \in E$$

Def.: Dado um st -fluxo x para $G = (V, E, c, s, t)$, definimos o valor do fluxo x como sendo $x(\delta^+(s)) - x(\delta^-(s))$.

Problema st -FLUXO MÁXIMO Dado um grafo direcionado $G = (V, E)$, capacidades $c : E \rightarrow \mathbb{Q}^+$ e vértices s e t , encontrar um fluxo de s para t de valor máximo.

Formulação Linear

- Para colocar no formato do problema de fluxo de custo mínimo, adicione uma aresta $t \rightarrow s$.

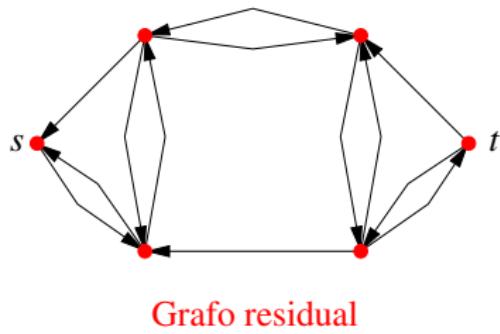
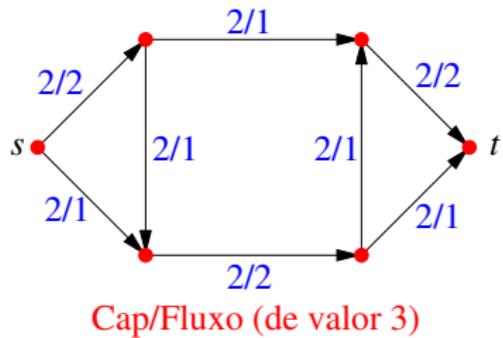


$$\begin{array}{ll} \text{maximize} & x_{ts} \\ \text{sujeito a} & \left\{ \begin{array}{lcl} \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e & = & 0 & \forall v \in V \\ 0 \leq x_e \leq c_e & & \forall e \in E. \end{array} \right. \end{array}$$

Corolário: Se as capacidades c_e são inteiras, então os vértices do poliedro do fluxo são inteiros.

Fluxo Máximo por Caminhos Aumentadores

Grafo Residual: Dado um grafo direcionado $G = (V, E)$, capacidades $c : E \rightarrow \mathbb{Q}^+$, vértices s e t e fluxo f de s para t , o grafo residual é o grafo $G_f = (V, A)$ onde $(u, v) \in A$ sse $(u, v) \in E$ e $f(u, v) < c(u, v)$ ou $(v, u) \in E$ e $f(u, v) > 0$.



Fluxo Máximo por Caminhos Aumentadores

Def.: Dado um fluxo f , um caminho aumentador de s a t é uma sequência de vértices e arestas $P = (v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k)$ onde

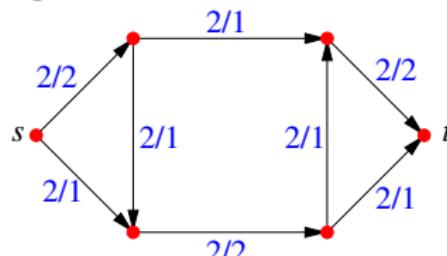
- $s = v_1$ e $t = v_k$ (o caminho vai de s a t)
- $e_i = (v_i, v_{i+1})$ ou $e_i = (v_{i+1}, v_i)$
- Se $e_i = (v_i, v_{i+1})$ então $f(e) < c(e)$
(é possível mandar mais fluxo pela aresta e , aumentando o fluxo que chega em v_{i+1})
- Se $e_i = (v_{i+1}, v_i)$ então $f(e) > 0$
(é possível diminuir o fluxo pela aresta e , aumentando o fluxo que chega em v_{i+1})

Caminho aumentador dado por um caminho no grafo residual

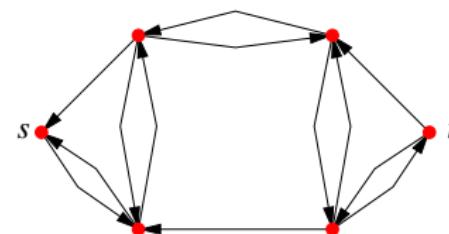
Fluxo Máximo por Caminhos Aumentadores

Idéia: Se existir caminho aumentador de s para t , podemos aumentar o fluxo atual.

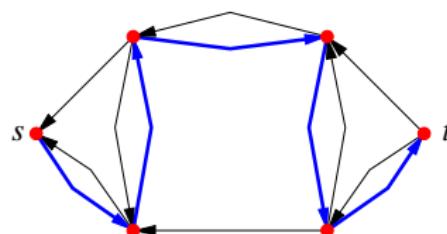
Exemplo: [de Caminho aumentador]



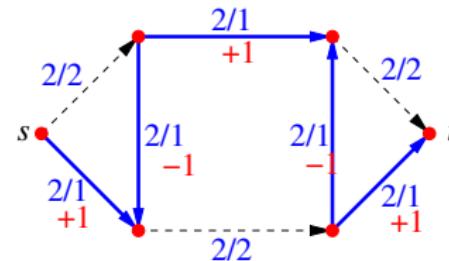
Cap/Fluxo (de valor 3)



Grafo residual



Caminho aumentador

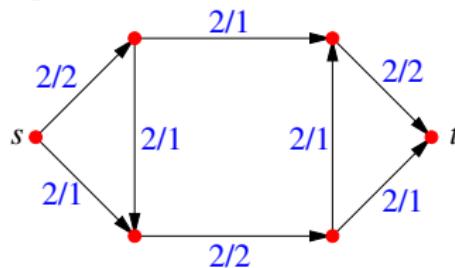


Caminho aumentador de valor 1

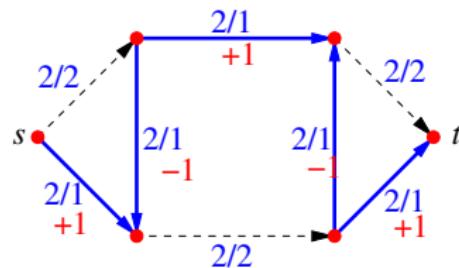
Fluxo Máximo por Caminhos Aumentadores

Idéia: Um caminho aumentador de s para t aumenta o fluxo atual.

Exemplo: [de Caminho aumentador]



Cap/Fluxo (de valor 3)



Caminho aumentador de valor 1

Dado caminho aumentador $(s = v_1, e_1, v_2, e_2, \dots, e_{k-1}, v_k = t)$, defina $\Delta(e_i)$ como

$$\Delta(e_i) = \begin{cases} c(e_i) - f(e_i) & \text{se } e_i = (v_i, v_{i+1}) \\ f(e_i) & \text{se } e_i = (v_{i+1}, v_i) \end{cases}$$

O valor do caminho aumentador é o valor $\min\{\Delta(e_i) : 1 \leq i < k\}$.

Def.: A aresta $e_i \in P$ é dita saturada se $\Delta(P) = \Delta(e_i)$.

Fluxo Máximo por Caminhos Aumentadores

Dado fluxo f , como econtrar um caminho aumentador ?

- Use um algoritmo de busca em largura ou profundidade no grafo residual
- Considerando que cada vértice enxerga os arcos que saem e os que entram nele
- Partindo de s , se a busca atingir t obtenha o valor do caminho aumentador

Algoritmo Ford-Fulkerson

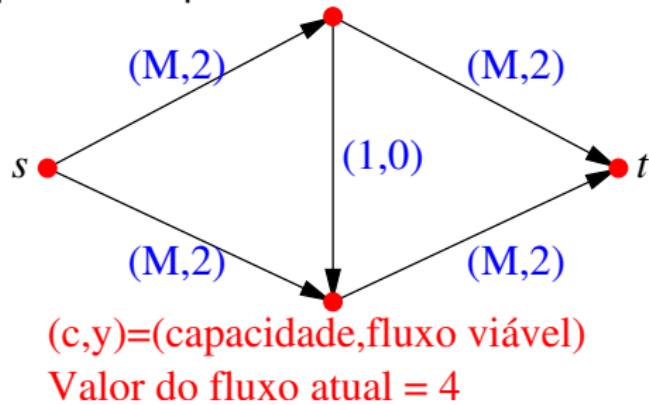
FORD-FULKERSON

- 1 Faça $f(e) \leftarrow 0$ para todo $e \in E$.
- 2 Enquanto existir caminho aumentador P no grafo residual G_f faça
- 3 Seja $P = (s = v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k = t)$
- 4 Seja $\Delta(P) = \min\{\Delta(e) : e \in P\}$ valor do caminho aumentador P
- 5 Faça $f(e_i) \leftarrow \begin{cases} f(e_i) + \Delta(P) & \text{se } e_i = (v_i, v_{i+1}) \\ f(e_i) - \Delta(P) & \text{se } e_i = (v_{i+1}, v_i) \end{cases}$ p/ $i = 1, \dots, k-1$
- 6 devolva f

O Problema do Fluxo Máximo é resolvido por várias resoluções do Problema da Busca em Grafos.

Proposição: O algoritmo de Ford-Fulkerson não é de tempo polinomial.

Considere o grafo seguinte, com capacidade M nas arestas, exceto na aresta do meio, que tem capacidade 1:



A cada iteração, use o caminho aumentador que usa a aresta de capacidade 1. Valor do fluxo máximo: $2M$.

Número de iterações: $O(M)$.

Algoritmo Edmonds-Karp

EDMONDS-KARP

FORD-FULKERSON usando BFS para obter caminhos aumentadores nos grafos residuais.

Lema: Se H e H' grafos residuais no início de iterações consecutivas, P_v e P'_v caminhos aumentadores de s a v em H e H' , resp. então $|P_v| \leq |P'_v|$, $\forall v \in V$

Prova. Por contradição. Suponha que exista v tal que $|P_v| > |P'_v|$.

Dado caminho P (ou P', P_v, \dots), também usamos P para seu tamanho.

Pegue primeira ocorrência do contraexemplo com P'_v mínimo, f e f' fluxos nos inícios das it. consecutivas (' para segunda . I.e., temos

$$P_v > P'_v.$$

Algoritmo Edmonds-Karp

Seja uv a última aresta de P'_v e $P'_v = Q'_u \parallel uv$ (Q'_u trecho até u).

$$P_u \leq Q'_u = P'_v - 1 < P_v - 1 \quad (\text{pois } u \text{ não é contraexemplo.})$$

Temos $uv \notin H$, caso contrário, tomando $Q_v = P_u \parallel uv$

$$P_v \leq Q_v = P_u + 1 \leq Q'_u + 1 = P'_v < P_v.$$

Portanto, $vu \in H$ a caminho aumentador da it. anterior:

$$P_v = P_u - 1 \leq P'_u - 1 = P'_v - 2.$$

Contradição.

□

Algoritmo Edmonds-Karp

Lema: Cada aresta (u, v) pode ser saturada no máximo $|V|/2$ vezes.

Prova. Se (u, v) fica crítica em f , ela só volta a aparecer quando passar por iteração de fluxo f' que usa (v, u) .

$$P_v^f = P_u^f + 1$$

$$P_v^{f'} = P_u^{f'} + 1$$

Portanto

$$P_u^{f'} = P_v^{f'} + 1 \geq P_v^f + 1 = P_u^f + 2.$$

I.e., a distância até u aumenta de 2. □

Teorema: O algoritmo de Edmonds-Karp tem complexidade de tempo $O(VE^2)$.

st-Corte Mínimo

Seja $G = (V, E)$ grafo direcionado, capacidades $c : E \rightarrow \mathbb{Q}^+$ e vértices $s, t \in V$.

Def.: Um *st*-corte é um conjunto $S \subseteq V$ tal que $s \in S$ e $t \in \bar{S}$ (onde $\bar{S} := V \setminus S$). Também é denotado pelo conjunto de arestas $\delta(S)$.

Def.: Definimos a *capacidade de um st-corte* S , como sendo a soma

$$c(\delta(S)) := \sum_{e \in \delta(S)} c_e.$$

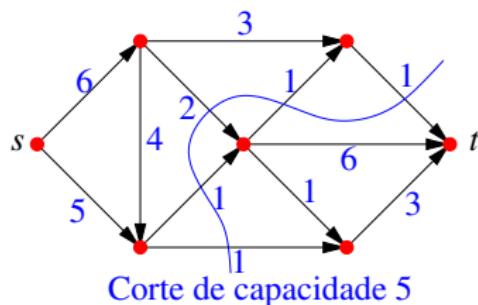
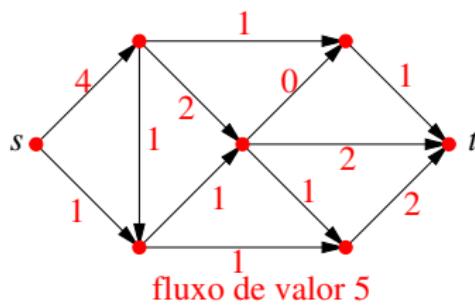
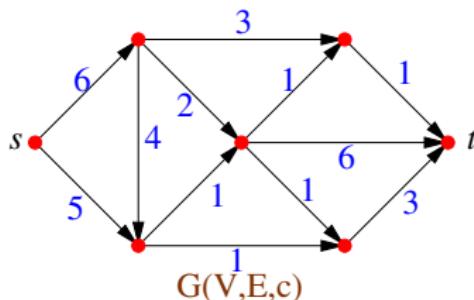
Problema *st*-CORTE MÍNIMO:

Dado grafo direcionado $G = (V, E)$, capacidades $c : E \rightarrow \mathbb{Q}^+$ e vértices $s, t \in V$, encontrar um *st*-corte de capacidade mínima.

Aplicações importantes na Computação: Projeto de redes de conectividade, Detecção de congestionamentos em redes de conectividade, Classificação de dados (data mining), Clusterização, particionamento de circuitos VLSI, etc

Claramente, um st -corte limita superiormente o valor de um fluxo máximo. De fato, vale que:

Teorema: O valor de um st -fluxo máximo de s para t é igual à capacidade de um st -corte mínimo.



Redução do Corte Mínimo para Fluxo Máximo

Def.: Um *st*-corte é um conjunto $S \subset V$ tal que $s \in S$ e $t \notin S$.

Def.: Definimos a capacidade de um *st*-corte S , como sendo a soma

$$c(\delta^+(S)) := \sum_{e \in \delta^+(S)} c_e.$$

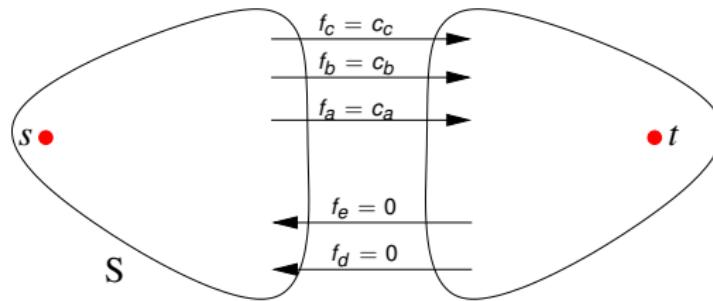
Problema: [do st-Corte Mínimo] Dado um grafo direcionado $G = (V, E)$, capacidades $c : E \rightarrow \mathbb{Q}^+$ e vértices s e t , encontrar um *st*-corte de capacidade mínima.

Redução do Corte Mínimo para Fluxo Máximo

Teorema: *O algoritmo de Ford-Fulkerson, obtém um fluxo de valor máximo.*

Prova. Exercício. Dicas:

- Considere a última busca realizada para tentar obter um caminho aumentador.
- Esta busca partiu de s , mas não atingiu t .
- Seja S os vértices atingidos pela busca.
- Mostre que S é um corte com capacidade igual ao valor do fluxo encontrado, com a seguinte cara:



□

Fluxo Máximo × Corte Mínimo

Do teorema anterior, conclua o seguinte teorema:

Teorema: [do Fluxo Máximo-Corte Mínimo] *O valor do fluxo máximo de s para t é igual à capacidade do menor corte separando s e t .*

Prova. Exercício. Dica: Use a prova que o Algoritmo Ford-Fulkerson obtém fluxo com valor igual ao de um corte. □

Exercício: Reduza o problema do st -Corte Mínimo para o problema do Fluxo de Valor Máximo.

Caminhos Disjuntos e Teoremas de Menger

Exercício: Apresente algoritmos de tempo polinomial:

Problema: [Caminhos Disjuntos em Grafos direcionados] Dado grafo direcionado e vértices s e t , encontrar o maior número de st -caminhos direcionados disjuntos nos arcos.

Problema: [Corte Separador de s e t em grafos direcionados] Dado grafo direcionado e vértices s e t , encontrar o menor número de arcos cuja remoção desconecta todos os st -caminhos.

Problema: [Caminhos Disjuntos em Grafos não-direcionados] Dado grafo não-direcionado e vértices s e t , encontrar o maior número de st -caminhos disjuntos nas arestas.

Problema: [Corte Separador de s e t em grafos não-direcionados] Dado grafo não-direcionado e vértices s e t , encontrar o menor número de arestas cuja remoção desconecta todos os st -caminhos.

Caminhos Disjuntos e Teoremas de Menger

Exercícios: Apresente algoritmos de tempo polinomial:

Problema: [Caminhos Disjuntos nos Vértices em Grafos direcionados]

Dado grafo direcionado e vértices s e t , encontrar o maior número de st -caminhos direcionados disjuntos nos vértices.

Problema: [Corte de Vértices que separa s e t em grafos direcionados]

Dado grafo direcionado e vértices s e t , encontrar o menor número de vértices cuja remoção desconecta todos os st -caminhos.

Problema: [Caminhos Disjuntos nos Vértices em Grafos

não-direcionados] Dado grafo não-direcionado e vértices s e t ,

encontrar o maior número de st -caminhos disjuntos nos vértices.

Problema: [Corte de Vértices que separa s e t em grafos

não-direcionados] Dado grafo não-direcionado e vértices s e t ,

encontrar o menor número de vértices cuja remoção desconecta todos os st -caminhos.

Exercícios:

- Faça um programa que, dado um st -fluxo de valor máximo, encontra um st -corte mínimo em tempo linear.
- Dado um grafo *não-direcionado*, com capacidades nas arestas, indicando a capacidade de fluxo que pode ir de um extremo ao outro de uma aresta, em qualquer uma das direções. O problema do fluxo máximo de um vértice s a um vértice t é encontrar um fluxo que sai de s e chega em t , respeitando a capacidade nas arestas e a lei de conservação do fluxo em cada vértice interno. Reduza este problema ao problema de encontrar um st -fluxo máximo em grafo direcionado.
- Dado um grafo não-direcionado com capacidades nas arestas, e dois vértices s e t , resolva o problema de encontrar um corte de capacidade mínima separando s e t .

Caminhos Disjuntos e Teoremas de Menger

Exercício: Apresente algoritmos de tempo polinomial

Def.: Um grafo conexo não-direcionado G é k -aresta conexo (resp. k -vértice conexo) se a remoção de menos que k arestas (resp. vértices) quaisquer nos dá um grafo conexo.

Problema: Mostre que um grafo não-direcionado é k -aresta conexo (resp. k -vértice conexo) se e somente se há k caminhos aresta disjuntos (resp. vértice disjuntos) entre quaisquer par de vértices.

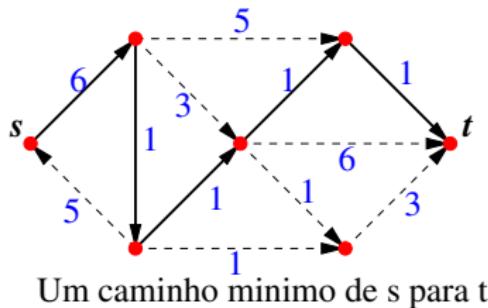
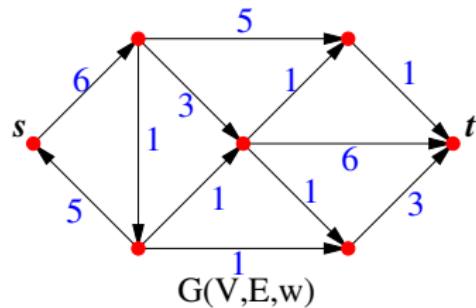
Problema: Decida se um grafo não-direcionado é k -aresta conexo (resp. k -vértice conexo), caso contrário, apresente conjunto com menos que k arestas (resp. vértices) cuja remoção nos dá um grafo desconexo.

Problema: Formule a versão direcionada do problema acima e proponha um algoritmo que o resolva em tempo polinomial.

Um excelente livro sobre problemas em fluxo é dado em Ahuja, Magnanti e Orlin; Network Flows: Theory, Algorithms and Applications

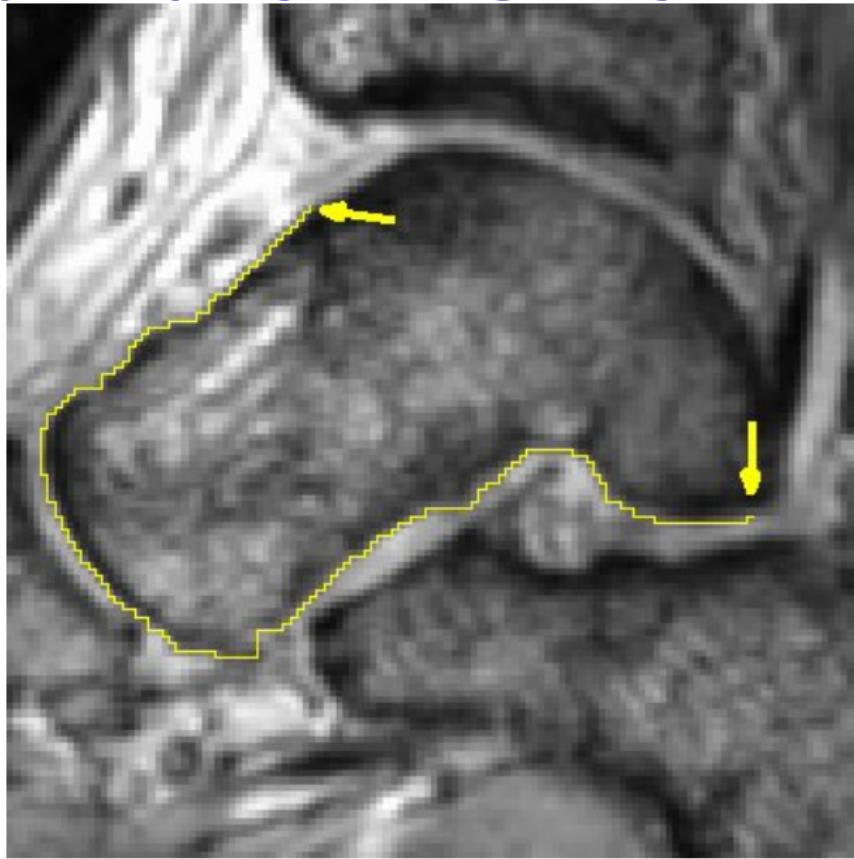
Problema do Caminho Mínimo

Problema CAMINHO MÍNIMO: Dado um grafo direcionado $G = (V, E)$, custos nas arestas $w : E \rightarrow \mathbb{Q}^+$ e vértices s e t , encontrar um caminho de s para t de custo mínimo.



Aplicações: Determinação de rotas de custo mínimo, segmentação de imagens, Escolha de centro distribuidor, reconhecimento de fala, etc

Exemplo de aplicação em segmentação de imagens



Formulação para o Problema do Caminho Mínimo

Considere a seguinte formulação:

$$(P) \quad \begin{array}{ll} \text{minimize} & \sum_{e \in E} w_e x_e \\ \text{sujeito a} & \left\{ \begin{array}{lcl} \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e & = & 0 \quad \forall v \in V \setminus \{s, t\} \\ \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e & = & 1 \\ \sum_{e \in \delta^+(t)} x_e - \sum_{e \in \delta^-(t)} x_e & = & -1 \\ 0 \leq x_e \leq 1 & & \forall e \in E. \end{array} \right. \end{array}$$

- Formulação é caso particular do problema do fluxo de custo mínimo

Corolário: Se x é um ponto extremal ótimo de (P) , então x é inteiro.
I.e., $x_e \in \{0, 1\} \forall e \in E$.

Teorema: Se x é um ponto extremal ótimo de (P) , então as arestas $e \in E$ onde $x_e = 1$ formam um caminho mínimo ligando s a t .

Caminhos Disjuntos de Custo Mínimo

Exercícios:

- 1 Considerando a seguinte formulação para o problema do caminho mínimo:

$$(P) \quad \begin{aligned} \min \quad & \sum_{e \in E} w_e x_e \\ \text{s.a.} \quad & \left\{ \begin{array}{lcl} \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e & = & 0 \quad \forall v \in V \setminus \{s, t\} \\ \sum_{e \in \delta^+(s)} x_e & = & 1 \\ \sum_{e \in \delta^-(t)} x_e & = & 1 \\ 0 \leq x_e \leq 1 & & \forall e \in E \setminus (\delta^-(s) \cup \delta^+(t)) \\ 0 \leq x_e \leq 0 & & \forall e \in \delta^-(s) \cup \delta^+(t) \end{array} \right. \end{aligned}$$

Mostre que o poliedro associado a esta formulação é inteiro.

- 2 Se x é um ponto extremal ótimo de (P) , então as arestas $e \in E$ onde $x_e = 1$ formam um caminho mínimo ligando s a t .

Caminhos Disjuntos de Custo Mínimo

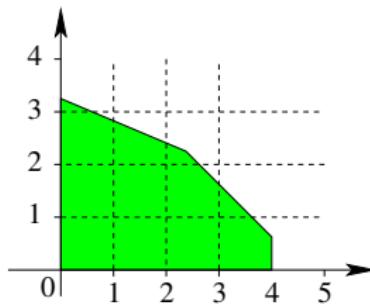
Exercícios:

- 1 Considere um grafo não-direcionado $G = (V, E, w)$, com pesos nas arestas $w : E \rightarrow \mathbb{Q}^+$, vértices s e t e inteiro positivo k . Projete um algoritmo de tempo polinomial que encontra k caminhos disjuntos nas arestas, P_1, \dots, P_k , que ligam o vértice s ao vértice t , e cujo peso total das arestas é mínimo; ou, se for o caso, escreva que não existem estes k caminhos.
- 2 Descreva o problema do exercício anterior, mas para grafos direcionados. Projete um algoritmo de tempo polinomial para resolvê-lo.
- 3 Descreva os problemas dos dois exercícios anteriores, mas para caminhos disjuntos nos vértices internos (os caminhos só se interceptam nos vértices s e t).

Def.: Dado um poliedro P , definimos seu *fecho inteiro*, P_I , como sendo o fecho convexo dos vetores inteiros de P . I.e.,

$$P_I := \text{conv}\{x \in P : x \text{ é inteiro}\}$$

Exemplo: Exemplo de poliedro P , conjunto I dos pontos inteiros em P e o fecho convexo de I .



Poliedro P

Pontos inteiros I

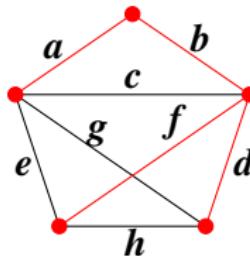
$P_I := \text{conv}(I)$

Def.: Dado um conjunto finito e ordenado E , dizemos que $\chi^A := (\chi_e^A : e \in E)$ é o **vetor de incidência** de A , $A \subseteq E$; onde $\chi_e^A = 1$ se $e \in A$ e 0 caso contrário.

Exemplo: Se $E = (a, b, c, d, e, f, g)$ e $A = \{a, c, e, f\}$ e χ^A é um vetor de incidência de A em E

$$\chi^A = (1, 0, 1, 0, 1, 1, 0).$$

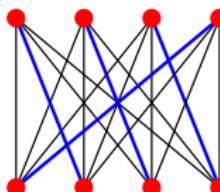
Exemplo: Seja $G = (V, E)$ o grafo abaixo com ordenação das arestas $E = (a, b, c, d, e, f, g, h)$ e T o subgrafo definido pelas arestas em vermelho.



Então o vetor de incidência das arestas de T em E é $\chi^T = (1, 1, 0, 1, 0, 1, 0)$.

Emparelhamento em Grafos Bipartidos

Def.: Dado um grafo $G = (V, E)$, dizemos que $M \subseteq E$ é um emparelhamento de G se M não tem arestas com extremos em comum.



Problema EMPARELHAMENTO-BIPARTIDO: Dados um grafo bipartido $G = (V, E)$, e custos nas arestas $c : E \rightarrow \mathbb{Z}$, encontrar emparelhamento $M \subseteq E$ que maximize $c(M)$.

Aplicações: Encontrar atribuição de candidatos para vagas maximizando aptidão total, atribuição de motoristas de veículos, formação de equipes (eg. com um chefe e subordinados), etc.

Exercício: Reduza o problema de encontrar um Emparelhamento de Cardinalidade Máxima em grafos bipartidos ao Problema do Fluxo Máximo. Mostre que a aplicação do Algoritmo Ford-Fulkerson nos dá um algoritmo de tempo polinomial para este problema.

Formulação para Emparelhamento em Grafos Bipartidos

Programa Linear Inteiro

$$\begin{array}{ll}\text{maximize} & \sum_{e \in E} c_e x_e \\ \text{sujeito a} & \left\{ \begin{array}{lcl} \sum_{e \in \delta(v)} x_e & \leq & 1 \quad \forall v \in V \\ 0 \leq x_e & \leq & 1 \quad \forall e \in E \\ x_e & \text{inteiro} & \forall e \in E \end{array} \right.\end{array}$$

PLI e não PL: Note que a formulação acima só não é um programa linear, devido a restrição de integralidade em x_e .

Relaxação Linear

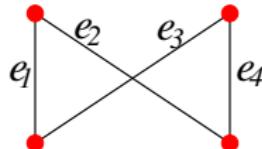
$$(LP_{Emp}) \quad \begin{array}{ll} \text{maximize} & \sum_{e \in E} c_e x_e \\ \text{sujeito a} & \begin{cases} \sum_{e \in \delta(v)} x_e \leq 1 & \forall v \in V \\ 0 \leq x_e \leq 1 & \forall e \in E \end{cases} \end{array}$$

Teorema: Os vértices do poliedro LP_{Emp} são inteiros.

Prova. Exercício. □

Exemplo:

Considere o seguinte grafo bipartido com custos unitários:



Programa Linear

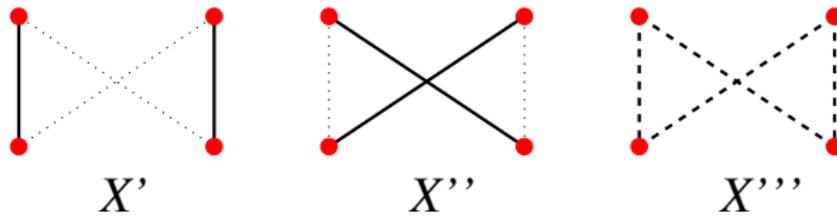
$$\begin{array}{ll}\text{maximize} & x_{e_1} + x_{e_2} + x_{e_3} + x_{e_4} \\ \text{sujeito a} & \left\{ \begin{array}{l} x_{e_1} + x_{e_2} \leq 1, \\ x_{e_3} + x_{e_4} \leq 1, \\ x_{e_1} + x_{e_3} \leq 1, \\ x_{e_2} + x_{e_4} \leq 1, \\ 0 \leq x_{e_1} \leq 1, \\ 0 \leq x_{e_2} \leq 1, \\ 0 \leq x_{e_3} \leq 1, \\ 0 \leq x_{e_4} \leq 1 \end{array} \right.\end{array}$$

Variáveis do programa linear correspondente:

$$X = (x_{e_1}, x_{e_2}, x_{e_3}, x_{e_4})$$

Os seguintes vetores são soluções ótimas do programa linear:

$$X' = (1, 0, 0, 1) \quad X'' = (0, 1, 1, 0) \quad X''' = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$$



- X' e X'' são vértices do poliedro do emparelhamento
- X''' é solução ótima, mas é combinação convexa de X' e X''
($X''' = \frac{1}{2}X' + \frac{1}{2}X''$). I.e., X''' não é vértice de P_{Emp} .

Problema do Transporte

Problema PROBLEMA DO TRANSPORTE: São dados um conjunto A de fornecedores e um conjunto B de consumidores, onde $A = \{1, \dots, m\}$ e $B = \{1, \dots, n\}$, todos relativos a um mesmo material. O fornecedor $i \in A$ produz a_i unidades e o consumidor $j \in B$ consome b_j unidades. O custo para transportar uma unidade do fornecedor i para o consumidor j é de c_{ij} , para todo $i \in A$ e $j \in B$. Considere que $T = \sum_i a_i = \sum_j b_j$. O problema consiste em determinar a forma mais barata para transportar T unidades dos fornecedores para os consumidores, satisfazendo a produção e consumo dos produtores e consumidores.

Aplicações: Transporte de materiais, abastecimento (energia, água,...), transmissão de dados/broadcast

Problema do Transporte

Exercício: Resolva o Problema do Transporte.

Exercício: Resolva uma variação do Problema do Transporte (capacitado nas arestas) supondo que o fornecimento x_{ij} é no máximo um valor u_{ij} , dado na entrada, para todo $i \in A$ e $j \in B$.

Exercício: Resolva uma variação do Problema do Transporte supondo que cada aresta tem uma capacidade máxima, $\sum_{i \in A} a_i \geq \sum_{j \in B} b_j$ (não necessariamente valendo na igualdade) devemos atender todos os consumidores, mas os fornecedores só precisam respeitar capacidade de produção (não necessariamente, a produção de um fornecedor i precisa ser consumida).

Exercício: b -matching: Dados grafo bipartido $G = (A, B, E)$ com função de peso nas arestas $w : E \rightarrow \mathbb{Q}$, e função limitadora de grau $b : A \cup B \rightarrow \mathbb{Z}^+$, encontrar um conjunto de arestas M tal que o número de arestas de M incidentes a um vértice i é no máximo b_i e o peso total das arestas de M é máximo.

Exercícios

Exercício: Dados grafo bipartido $G = (V, E)$ com função de peso nos vértices $w : V \rightarrow \mathbb{Q}$, encontrar um conjunto de vértices $I \subseteq V$ que não tem arestas incidentes em comum (conjunto independente).

Apresente uma formulação em programação linear inteira para resolver este problema, cuja relaxação apresente apenas vértices inteiros. Com isso, projete um algoritmo de tempo polinomial para encontrar um conjunto independente em G de peso máximo.

Exercício: Dados grafo bipartido $G = (V, E)$ com função de peso nos vértices $w : V \rightarrow \mathbb{Q}$, encontrar um conjunto de vértices $C \subseteq V$ tal que toda aresta $e \in E$ tem pelo menos um dos extremos em C (cobertura por vértices). Faça como no exercício anterior e projete um algoritmo de tempo polinomial para encontrar uma cobertura por vértices em G de peso mínimo. Outra maneira de resolver o problema da cobertura por vértices, é reduzindo este problema para o anterior. Mostre como isto pode ser feito.

Exercícios

Exercício: Dados grafo bipartido $G = (V, E)$ com função de peso nos vértices $w : V \rightarrow \mathbb{Q}$, encontrar um conjunto de arestas $F \subseteq E$ tal que para todo vértice $v \in V$, há pelo menos uma aresta de F incidente em v (cobertura por arestas). Faça como no exercício anterior e projete um algoritmo de tempo polinomial para encontrar uma cobertura por vértices em G de peso mínimo.