Projeto e Análise de Algoritmos

Programação Linear

Flávio Keidi Miyazawa et al.

Primeiro Semestre de 2017

Programação Linear

Problema da Dieta

São dados:

- m tipos de alimentos: $ali_1, \ldots, ali_m$
- n tipos de nutrientes: $nut_1, \ldots, nut_n$
- preço p_j de cada alimento ali_j
- recomendação diária mínima min_i e máxima max_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 de instância

Dieta I:

Recomendações diárias:

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

Alimentos:

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

Escrevendo formalmente

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.

Custo da dieta

- **Função de custo:**

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

- **Objetivo:**

minimizar o preço pago por todos os alimentos

Restrições

- as quantidades x_1, x_2 e x_3 devem ser números reais positivos e devem satisfazer as restrições nutricionais

Especificando as restrições

Restrições nutricionais

- Limites para quantidade de cálcio:

$$110x_1 + 170x_2 + 1150x_3 \geq 800$$
$$110x_1 + 170x_2 + 1150x_3 \leq 1200$$

- Limites para quantidade de ferro:

$$29x_1 + 15x_2 \geq 10$$
$$29x_1 + 15x_2 \leq 18$$

Restrições de não negatividade

- Nenhuma quantidade pode ser negativa:

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

Formulação Linear

Obtemos o que chamamos de formulação linear:

$$\begin{array}{ll}
\text{minimize} & 20{,}0x_1 + 6{,}0x_2 + 2{,}0x_3 \\
\text{sujeito a} & \left\{\begin{array}{rcl}
110x_1 + 170x_2 + 1150x_3 & \geq & 800 \\
110x_1 + 170x_2 + 1150x_3 & \leq & 1200 \\
29x_1 + \ \ 15x_2 & \geq & 10 \\
29x_1 + \ \ 15x_2 & \leq & 18 \\
\multicolumn{3}{c}{x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0}
\end{array}\right.
\end{array}$$

Resolvendo-se essa formulação, descobrimos que a dieta mais barata custa 5,194 reais. Uma dieta com esse custo é comprar:

$x_1 = 0$ kg de carne de boi,
$x_2 = 0{,}666$ kg de feijão cozido e
$x_3 = 0{,}597$ L de leite desnatado.

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

Programação Linear

Por que estudar 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.

Definição

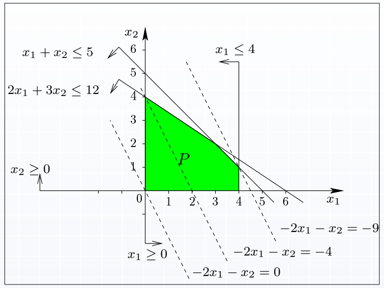
PL

No Problema de Programação Linear (PL), é dada uma matriz $A = (a_{ij}) \in \mathbb{Q}^{n \times m}$, vetores $c = (c_i) \in \mathbb{Q}^n$ e $b = (b_i) \in \mathbb{Q}^m$, e queremos encontrar um vetor $x = (x_i) \in \mathbb{Q}^m$ tal que

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

ou decidir que não existe um tal vetor.

Exemplo gráfico



$$
\begin{array}{ll}
\text{minimize} & -2x_1 - x_2 \\
\text{sujeito a} & \left\{\begin{array}{r}
x_1 \leq 4 \\
x_1 + x_2 \leq 5 \\
2x_1 + 3x_2 \leq 12 \\
x_1 \geq 0 \\
x_2 \geq 0
\end{array}\right.
\end{array}
$$

Solução ótima: $x_1 = 4$ e $x_2 = 1$ Valor da solução ótima: -9

Algoritmos polinomiais

Teorema

PL pode ser resolvido em tempo polinomial.

Algoritmos polinomiais:

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

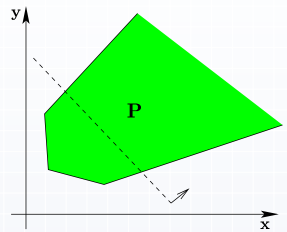
Um algoritmo exponencial:

- Método simplex (Dantzig'47)

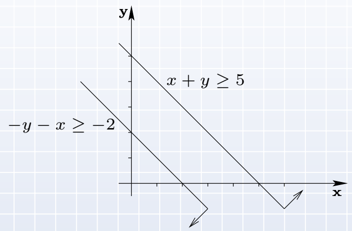
Observação: O simplex tem tempo médio polinomial e na prática costuma ser mais rápido.

Nem sempre encontramos uma solução ótima

- Sistema ilimitado:



- Sistema inviável:



Outra dieta

Dieta II:
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

Outra dieta: tentando resolver

Formulação linear:

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

Este sistema é inviável pois:

- Somente feijão contém vitamina C (e bem pouco):
  - Como $10x_2 \geq 60$, temos que $x_2 \geq 6$
  - Ou seja, precisamos de 6 kg de feijão por dia!
- Mas o feijão também tem fósforo
  - 6 kg de feijão contêm $6 \times 490 = 2940$ mg de fósforo
  - Bem acima do limite diário permitido de 1200 mg

Resolvedores de Sistemas Lineares

- Implementar um algoritmo para resolver PL é não trivial
- Na indústria aparecem PLs de milhões de variáveis
- Normalmente utilizamos um pacote (*solver*)

Programas comerciais:

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

Programas livres

- SCIP: http://scip.zib.de/
- COIN-CLP: http://www.coin-or.org/Clp/
- SoPlex / Zib: http://www.zib.de/Optimization/Software/Soplex
- GLPK / Gnu: http://www.gnu.org/software/glpk/glpk.html
- LEMON / Coin-OR: http://www.coin-or.org/
  Biblioteca de otimização/grafos para solver GLPK ou COIN-CLP

Trecho de implementação usando LEMON

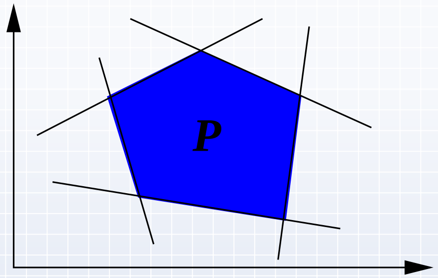
```
#include <lemon/lp.h> // g++ ex_lp1.cpp -lemon -lglpk -o ex_lp1
using namespace lemon;
using namespace std;
int main() {
  Lp lp;                             // Cria um PL
  Lp::Col x1 = lp.addCol();          // Adiciona duas variáveis
  Lp::Col x2 = lp.addCol();
  lp.addRow(   x1 +     x2 <= 5 );   // Adiciona duas restrições
  lp.addRow( 2*x1 + 3*x2  <= 12);
  lp.colLowerBound( x1, 0 );         // Limites inferior e superior de variável
  lp.colUpperBound( x1, 4 );
  lp.colLowerBound( x2, 0 );
  lp.obj( -2*x1 - x2 );              // Define função-objetivo
  lp.min();                          // Direção: queremos minimizar a função
  lp.solve();                        // Resolve o sistema
  if (lp.primalType()==Lp::OPTIMAL){ // Imprimindo valor das variáveis
    cout << "Valor da funcao objetivo: " << lp.primal() << endl;
    cout << "x1 = " << lp.primal(x1) << endl;
    cout << "x2 = " << lp.primal(x2) << endl;
  } else {
    cout << "Nao encontrou solucao otima." << endl;
  }
  return 0;
}
```

Definições

Algumas Definições

Um politopo[1] é um conjunto de pontos da forma

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



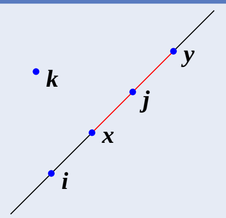
[1] Quando $n = 3$, o politopo também é chamado de poliedro.

Combinação convexa

Dado um conjunto de pontos $S = \{y_1, y_2, \ldots, y_n\}$, dizemos que y é uma combinação convexa dos pontos de S se

- $y = \alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n$, onde $\alpha_i \geq 0$ e
- $\sum_i \alpha_i = 1$.

Exemplo

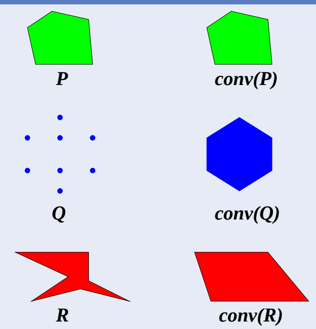


j é combinação convexa de x e y, mas i e k não são.

Fecho convexo

O fecho convexo de S, denotado por $conv(S)$ é o conjunto dos pontos que são combinação convexa dos pontos de S.

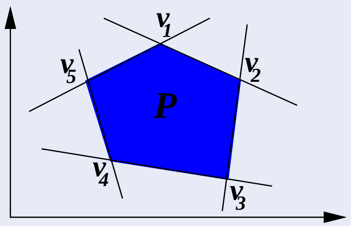
Exemplos



Vértices do politopo

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

Exemplo



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

Note que um vértice do poliedro é o único ponto que satisfaz um determinado conjunto de igualdades simultaneamente.

Determinantes e Resolução de Sistemas Lineares

Dada matriz quadrada A de ordem n e coluna j, podemos calcular o determinante pela fórmula de Laplace:

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

onde $A^{i,j}$ é a submatriz obtida ao se remover a linha i e a coluna j.

Método de Cramer

Se $|A| \neq 0$, então podemos revolver o sistema de equações $Ax = b$ obtendo x da seguinte forma:

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

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

Politopo inteiro e Matriz Totalmente Unimodular

Uma submatriz de A é uma matriz obtida removendo linhas e colunas de A.

Teorema

Todo vértice y do politopo 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 elementos de b das linhas correspondentes de A'.

- Um politopo é inteiro se todos os seus vértices contêm apenas componentes inteiras.
- Uma matriz A é chamada de totalmente unimodular (TU) se o determinante de toda submatriz quadrada A' está em $\{-1, 0, +1\}$.

Teorema

Se A é totalmente unimodular, então para todo vetor inteiro b o politopo $P = \{x : Ax \leq b\}$ é inteiro.

Propriedades de matrizes TU

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. $(A|I)$
6. $(A|-A)$

Explicação:
1, 2 e 3. Seguem da definição matriz TU.
4. Determinante da transposta é igual ao da original; multiplicar linhas por -1 apenas inverte 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.

Aplicando em politopos

Teorema

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

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

Note que $\{x : Ax = b\}$ e $\{x : Ax \leq b,\ Ax \geq b\}$ são iguais.

Para demonstrar o item 4, observe que a matriz $\begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix}$ é TU.

Condições

De maneira geral, vale que:

Lema

Se A é uma 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: Por indução na ordem da submatriz. Para ordem 1, basta ver que a matriz só contém números -1, 0 ou 1. Agora seja B uma submatriz quadrada de A.

1. Se há coluna de elementos nulos, o determinante de B é 0.
2. Se há coluna com exatamente um elemento não nulo $t \in \{-1, +1\}$, então 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. ■

Matriz de incidência

Lema

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

Prova: Segue do Lema anterior.

Lema

Se A é a matriz de incidência de um grafo bipartido $G = (V, E)$, com partes X, Y, 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. ■

Exemplo: portfólio de investimentos

Otimização de Portfólio

- Temos 100.000 reais para investir em ações
- As ações selecionadas e a porcentagem de retorno esperado em um 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%

Algumas restrições

A recomendação dos especialistas é a seguinte:

- investa pelo menos 25% e no máximo 55% em estatais
- Petrobrás e Texaco são empresas do mesmo setor (petróleo); o o investimento nas duas não deve passar de 55%
- Vale do Rio Doce e Votorantim são do mesmo setor (siderurgia); o investimento nas duas não deve passar 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.

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

Restrições

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

Restrições impostas por especialistas

- Invista pelo menos 25% e no máximo 55% em estatais:
$$x_1 + x_5 \geq 25000$$
$$x_1 + x_5 \leq 55000$$
- Petrobrás e Texaco são empresas do mesmo setor (petróleo); o investimento nas duas não deve passar de 55%:
$$x_1 + x_4 \leq 55000$$
- Vale do Rio Doce e Votorantim são do mesmo setor (siderurgia); o investimento nas duas não deve passar de 45%:
$$x_2 + x_3 \leq 45000$$

Restrições impostas por especialistas (cont)

- 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:
$$x_2 \leq 0{,}6(x_2 + x_3)$$
que o mesmo que
$$-0.4x_2 + 0.6x_3 \geq 0$$

Demais Restrições

- Total investido é 100.000:
$$x_1 + x_2 + x_3 + x_4 + x_5 = 100000$$
- Nenhuma quantidade pode ser negativa:
$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0$$

Formulação Linear

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

Resolvendo-se obtemos uma lucro estimado de 9094 reais investindo:
$x_1 = 0$ na Petrobrás, $x_2 = 12000$ na Vale do Rio Doce,
$x_3 = 8000$ na Votorantim, $x_4 = 55000$ na Texaco e
$x_5 = 25000$ na Sanasa.

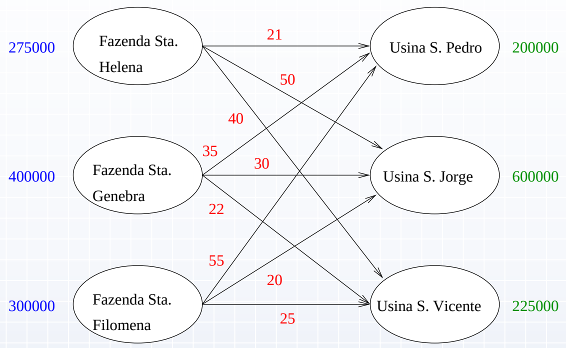
Problema de Fluxo de Custo Mínimo

Um problema de transporte

O Grupo CanaBraba possui três fazendas com canaviais e três usinas de produção de álcool. Os administradores da empresa estão organizando a logística da colheita da cana para a safra deste ano.

O transporte da cana das fazendas para as usinas é terceirizado. O custo do quilômetro rodado por tonelada de cana transportada que é cobrado pela transportadora é fixo independente do trajeto realizado. A produção das fazendas e a capacidade de processamento das usinas, ambas dadas em toneladas, assim como as distâncias em quilômetros entre as fazendas e as usinas são esquematizadas a seguir.

Cadeia de fornecimento da CanaBraba



Qual deve ser a quantidade de cana transportada de cada fazenda para cada usina de modo a minimizar o custo total do transporte?

Formulação

- Variáveis:
  $x_{ij} := \begin{cases} \text{quantidade de cana transportada} \\ \text{da fazenda } i \text{ para a usina } j \end{cases}$
- Função objetivo:
  minimize $z = \sum_{i=1}^{3} \sum_{j=1}^{3} d_{ij} x_{ij}$
- Restrições de capacidades das usinas:
  $x_{11} + x_{21} + x_{31} \leq 200000$ (Usina São Pedro)
  $x_{12} + x_{22} + x_{32} \leq 600000$ (Usina São Jorge)
  $x_{13} + x_{23} + x_{33} \leq 225000$ (Usina São Vicente)
- Escoamento da produção das fazendas:
  $x_{11} + x_{12} + x_{13} = 275000$ (Fazenda Santa Helena)
  $x_{21} + x_{22} + x_{23} = 400000$ (Fazenda Santa Genebra)
  $x_{31} + x_{32} + x_{33} = 300000$ (Fazenda Santa Filomena)
- Não negatividade:
  $x_{ij} \geq 0$ para todo $(i, j) \in \{1, 2, 3\} \times \{1, 2, 3\}$

Problema do Fluxo de Custo Mínimo

O problema de fluxo de modela um sistema de produção de bens onde há centros consumidores e produtores:

- todos itens produzidos devem ser consumidos
- item produzidos chegam até os consumidores por rotas
- uma rota tem capacidades máxima de escoamento
- uma rota tem custo por unidade para transportar um item

Objetivo: transportar itens de produtores para consumidores minimizando custo total de transporte

Algumas outras aplicações: detecção de "gargalos" da rede na transferência, projeto de vias de tráfego urbano, etc.

Definição

Sejam $G = (V, E)$ um grafo direcionado, capacidades $c : E \to \mathbb{Q}^+$, demandas $b : V \to \mathbb{Q}$ e custos $w : E \to \mathbb{Q}^+$.

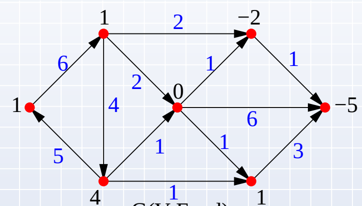
- um vértice $v \in V$ com $b_v < 0$ é chamado de consumidor
- um vértice $v \in V$ com $b_v > 0$ é chamado de produtor

Fluxo

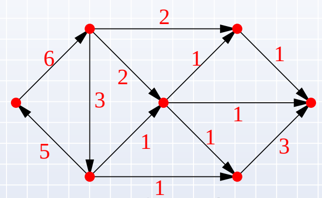
Uma função $x : E \to \mathbb{Q}^+$ é um fluxo em G se

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

$$0 \leq x_e \leq c_e, \forall e \in E$$



G(V,E,c,d)



Um fluxo viável

Problema do Fluxo de Custo Mínimo

Dado um fluxo $x : E \to \mathbb{Q}^+$ o custo do fluxo x é $\sum_{e \in E} w_e x_e$.

Problema do Fluxo de Custo Mínimo

Dados grafo direcionado $G = (V, E)$, capacidades $c : E \to \mathbb{Q}^+$, demandas $b : V \to \mathbb{Q}^+$ e função de custo $w : E \to \mathbb{Q}^+$ encontre fluxo $x : E \to \mathbb{Q}^+$ que:

$$\begin{array}{rl} & \min \sum_{e \in E} w_e x_e \\ \text{s.a} & \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = b_v \quad \forall v \in V \\ & 0 \le x_e \le c_e \quad \forall e \in E \end{array}$$

Integralidade do politopo

Teorema

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

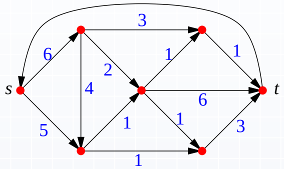
Prova: Segue do fato que a matriz de incidência de um grafo orientado é TU.

Subproblemas

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

Exemplo: reduzindo fluxo máximo para programação linear

Adicione uma aresta $t \to s$ sem capacidade:

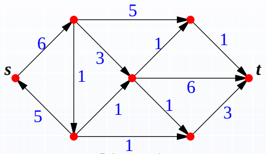


G(V,E) e capacidades

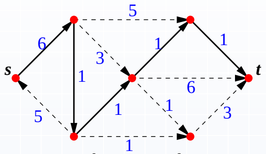
$$\begin{array}{ll} \text{maximize} & x_{ts} \\ \text{sujeito a} & \left\{ \begin{array}{rcll} \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e & = & 0 & \forall v \in V \\ 0 \le & x_e & \le c_e & \forall e \in E. \end{array} \right. \end{array}$$

Lembrando: se capacidades c_e são inteiras, existe fluxo máximo inteiro!
Uma outra prova: Basta ver que os vértices do poliedro são inteiros.

Exemplo: caminho mínimo como programação linear



G(V,E,w)



Um caminho mínimo de s para t

- Dado um grafo direcionado $G = (V, E)$ com pesos $w : E \to \mathbb{Q}^+$ e vértices s e t, encontrar um caminho de s para t de custo mínimo.
- Considere o vetor $(x_e)_{e \in E}$ em que x_e se e faz parte do caminho mínimo. Esse vetor x é chamado de vetor característico do caminho.

Exemplo: formulação de caminho mínimo

Consideramos o caso particular do problema de fluxo de custo mínimo em que s é produtor e t é consumidor.

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

Se x é um ponto extremal de (P), então x é um vetor característico de um caminho de s até t!

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.

Introdução a Dualidade em Programação Linear

Dualidade em Programação Linear

Considere o seguinte PL de minimização:

$$\begin{array}{ll} \text{minimize} & c_1x_1 + c_2x_2 + c_3x_3 \\ \text{sujeito a} & \left\{ \begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 & \geq & b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 & = & b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 & \leq & b_3 \\ x_1 \geq 0, \quad x_3 \leq 0 & & \end{array} \right. \end{array}$$

Vamos delimitar o valor ótimo do LP.
Multiplique as restrições por $y_1 \geq 0$, y_2 e $y_3 \leq 0$:

$$\begin{array}{rcl} y_1(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) & \geq & y_1b_1 \\ y_2(a_{21}x_1 + a_{22}x_2 + a_{23}x_3) & = & y_2b_2 \\ y_3(a_{31}x_1 + a_{32}x_2 + a_{33}x_3) & \geq & y_3b_3 \end{array}$$

Agora vamos somar as inequações

Dualidade em Programação Linear

$$
\begin{array}{lcl}
y_1(a_{11}x_1 + a_{12}x_2 + a_{13}x_3) & \geq & y_1 b_1 \\
y_2(a_{21}x_1 + a_{22}x_2 + a_{23}x_3) & = & y_2 b_2 \\
y_3(a_{31}x_1 + a_{32}x_2 + a_{33}x_3) & \geq & y_3 b_3 \\
\hline
(y_1 a_{11} + y_2 a_{21} + y_3 a_{31})x_1 + & & \\
(y_1 a_{12} + y_2 a_{22} + y_3 a_{32})x_2 + & \geq & (y_1 b_1 + y_2 b_2 + y_3 b_3) = yb \\
(y_1 a_{13} + y_2 a_{23} + y_3 a_{33})x_3 & &
\end{array}
$$

Comparando com a função objetivo $cx = c_1x_1 + c_2x_2 + c_3x_3$, percebemos que se tivéssemos

$$
\begin{array}{ll}
c_1x_1 & \geq (y_1 a_{11} + y_2 a_{21} + y_3 a_{31})x_1 \\
c_2x_2 & \geq (y_1 a_{12} + y_2 a_{22} + y_3 a_{32})x_2 \\
c_3x_3 & \geq (y_1 a_{13} + y_2 a_{23} + y_3 a_{33})x_3
\end{array}
$$

então teríamos $cx \geq yb$, ou seja, mostraríamos que yb é limitante de cx.

Dualidade em Programação Linear

Como temos $x_1 \geq 0$ e $x_3 \leq 0$, sabemos que para obter as condições anteriores é suficiente que

$$
\begin{array}{ll}
c_1x_1 & \geq (y_1 a_{11} + y_2 a_{21} + y_3 a_{31})x_1 \\
c_2x_2 & = (y_1 a_{12} + y_2 a_{22} + y_3 a_{32})x_2 \\
c_3x_3 & \leq (y_1 a_{13} + y_2 a_{23} + y_3 a_{33})x_3
\end{array}
$$

Naturalmente, queremos dentre todos os vetores y, o que melhor delimita $cx \geq yb$, i.e., o vetor cujo valor yb é máximo.
Obtemos o seguinte problema de maximização:

$$
\begin{array}{ll}
\text{maximize} & y_1 b_1 + y_2 b_2 + y_3 b_3 \\
\text{sujeito a} & \begin{cases}
y_1 a_{11} + y_2 a_{21} + y_3 a_{31} \leq c_1 \\
y_1 a_{12} + y_2 a_{22} + y_3 a_{32} = c_2 \\
y_1 a_{13} + y_2 a_{23} + y_3 a_{33} \geq c_3 \\
y_1 \geq 0, \quad y_3 \leq 0
\end{cases}
\end{array}
$$

Nomes dos sistemas

Problema Primal

$$
\begin{array}{ll}
\text{minimize} & c_1x_1 + c_2x_2 + c_3x_3 \\
\text{sujeito a} & \begin{cases}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leq b_3 \\
x_1 \geq 0, \quad x_3 \leq 0
\end{cases}
\end{array}
$$

Problema Dual

$$
\begin{array}{ll}
\text{maximize} & y_1 b_1 + y_2 b_2 + y_3 b_3 \\
\text{sujeito a} & \begin{cases}
y_1 a_{11} + y_2 a_{21} + y_3 a_{31} \leq c_1 \\
y_1 a_{12} + y_2 a_{22} + y_3 a_{32} = c_2 \\
y_1 a_{13} + y_2 a_{23} + y_3 a_{33} \geq c_3 \\
y_1 \geq 0, \quad y_3 \leq 0
\end{cases}
\end{array}
$$

Problemas primal e dual em geral

Considere:

- uma A uma matriz com m linhas e n colunas
- um vetor c com n constantes
- um vetor b com m constantes

Primal e Dual

$$
(Primal) \quad \begin{array}{ll} \text{min} & c\,x \\ \text{s.a} & Ax \geq b \end{array} \qquad\qquad (Dual) \quad \begin{array}{ll} \text{max} & b\,y \\ \text{s.a} & A^T y \leq c \end{array}
$$

- O programa primal tem:
  - n variáveis
  - m restrições
- O programa dual tem:
  - m variáveis
  - n restrições

Por que sempre podemos escrever um programa linear assim?

Folgas Complementares

Considere um programa primal (P) de minimização e seu programa dual (D) de maximização.

Lema

Para todas soluções $x \in P$ e $y \in D$, vale $cx \geq yb$.

Denote por A_i a i-ésima linha de A. Se $x \in P$ e $y \in D$, dizemos que:

- x satisfaz folgas complementares primais se para todo j vale:
$$x_j = 0 \quad \text{ou} \quad A_j^T = c_j$$
- y satisfaz folgas complementares duais se para todo i vale:
$$y_i = 0 \quad \text{ou} \quad A_i = c_j$$

Informalmente, ou a variável é 0, ou a restrição correspondente é justa.

Lema

Se x e y são soluções viáveis de (P) e (D), então $c\,x = y\,b$ se, e somente se, x e y satisfazem folgas complementares.

Teorema da dualidade

Denotamos por $\mathrm{OPT\text{-}LP}(P)$ o valor ótimo de um programa linear (P).

Teorema

Considere um programa primal (P) de minimização e seu programa dual (D) de maximização. Vale exatamente uma possibilidade:

1. *(P) e (D) são viáveis e $\mathrm{OPT\text{-}LP}(P) = \mathrm{OPT\text{-}LP}(D)$.*
2. *(P) é viável, (D) é inviável e $\mathrm{OPT\text{-}LP}(P) = -\infty$.*
3. *(P) é inviável, (D) é viável e $\mathrm{OPT\text{-}LP}(D) = \infty$.*
4. *(P) e (D) são inviáveis.*

Programação Linear Inteira

Programação Linear Inteira

- Um programa linear conjunto somente restrições lineares sobre variáveis racionais
- Portanto, o espaço de busca é convexo!
- Assim, podemos resolvê-lo em tempo polinomial — e um conjunto enorme de problemas modelados como PL

Mas há muitos problemas que não sabemos modelar como um PL:

- Caixeiro Viajante, Problema da Mochila, Árvore de Steiner, etc.

E se além da linearidade, exigirmos que as variáveis sejam **inteiras**?

$\Rightarrow$ O problema fica muito mais difícil :(

$\Rightarrow$ Mas podemos resolver diversos problemas novos na prática :)

Programação Linear Inteira

PL

No Problema de Programação Linear Inteira (PLI), é Dada UMA matriz $A = (a_{ij}) \in \mathbb{Q}^{n \times n}$, vetores $c = (c_i) \in \mathbb{Q}^n$ e $b = (b_i) \in \mathbb{Q}^m$, e queremos encontrar vetor $x = (x_i) \in \mathbb{Z}^n$ que

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

ou decidir que não exite um tal vetor.

Obs: se permitimos também variáveis racionais além de inteiras, então chamamos do programa de misto

Complexidade computacional

Spoilers: Tópico de MC658.

Teorema

PLI é um problema NP-difícil.

Isso é má notícia; significa que nem sempre sabemos resolvê-lo rapidamente!
Mas há estratégias para resolver problemas NP-difíceis usando PLI:

- por arredondamento da relaxação
- pelo método branch & bound
- pelo método de planos de corte
- pelo método branch & cut
- combinação de métodos

O que elas estratégias têm em comum é que todas começam modelando um problema como PLI, isso é, reduzindo uma instância de um problema difícil para uma instância de PLI.

Modelagem de Problemas como PLI

Modelagem por meio de variáveis 0/1

- útil quando uma solução é formada por decisões binárias
- para cada decisão i, usamos uma variável indicadora x_i:
  - $x_i = 0$ significa que a decisão **não** foi tomada
  - $x_i = 1$ significa que a decisão foi tomada

Exemplo: Seja U um conjunto (de arestas, vértices, etc.) e suponha que uma solução seja representada por $S \subseteq U$ então definimos para cada $i \in U$:

$$x_i = \begin{cases} 0 & \text{se } i \notin S \\ 1 & \text{se } i \in S \end{cases}$$

Chamamos x de vetor característico de S.

Problema da Mochila

Problema da Mochila

Dados itens $U = \{1, \ldots, n\}$, um valor v_i e um tamanho s_i inteiro para cada $i = 1, \ldots, n$ e um inteiro B, queremos encontrar encontrar $S \subseteq U$ que

$$\text{maximiza} \sum_{i \in S} v_i \quad \text{e} \quad \text{respeita} \sum_{i \in S} s_i \leq B.$$

Para cada $i \in U$, defina variáveis indicadoras x_i:

$$x_i = \begin{cases} 1 & \text{indica que o elemento } i \text{ pertence à solução e} \\ 0 & \text{indica que o elemento } i \text{ não pertence à solução.} \end{cases}$$

$$
\begin{array}{ll}
\text{maximize} & \sum_{i \in U} v_i x_i \\
\text{sujeito a} & \left\{ \begin{array}{l} \sum_{i \in U} s_i x_i \leq B \\ x_i \in \{0,1\} \quad \forall i \in U \end{array} \right.
\end{array}
$$

Restrições de Coberturas, Empacotamentos e Partições

Seja E um conjunto de elementos:

- $\mathcal{C}$ uma família de subconjuntos de E
- $\mathcal{C}_e := \{C \in \mathcal{C} : e \in C\}$ a família dos conjuntos que contém e
- x um vetor característico de $\mathcal{S} \subseteq \mathcal{C}$

Algumas condições recorrentes em problemas:

- $\mathcal{S}$ é uma cobertura se

$$\sum_{C \in \mathcal{C}_e} x_C \geq 1 \quad \forall e \in E,$$

- $\mathcal{S}$ é um empacotamento se

$$\sum_{C \in \mathcal{C}_e} x_C \leq 1 \quad \forall e \in E,$$

- $\mathcal{S}$ é uma partição se

$$\sum_{C \in \mathcal{C}_e} x_C = 1 \quad \forall e \in E.$$

Relaxação linear

Programa Inteiro e Programa Relaxado

PLI e Relaxação linear

$$(P_I) \quad \begin{array}{ll} \min & c\,x \\ \text{s.a} & Ax \geq b \\ & x_i \in \{0,1\} \quad \forall i \end{array} \qquad\qquad (P) \quad \begin{array}{ll} \min & c\,x \\ \text{s.a} & Ax \geq b \\ & 0 \leq x_i \leq 1 \quad \forall i \end{array}$$

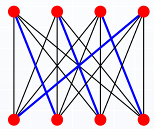
O programa (P) é um programa linear e é chamado de relaxação do PLI:

- a única diferença é:
  - em (P_I), x_i é um variável binária 0 ou 1
  - em (P), x_i está no intervalo $[0,1]$
- se $x \in P_I$ então $x \in P$, ou seja, $P_I \subseteq P$
- o mínimo em P é menor ou igual ao mínimo em P_I

Pergunta: E se fosse um problema de maximização?

Emparelhamento em Grafos Bipartidos: um exemplo

Dados um grafo bipartido $G = (V,E)$, encontrar emparelhamento $M \subseteq E$ de tamanho máximo.



Formulação em Programação Linear Inteira:

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

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

Emparelhamento em Grafos Bipartidos: Relaxação Linear

Relaxação linear:

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

Teorema

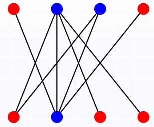
Os vértices do politopo da relaxação linear são inteiros.

Prova: Exercício.

Corolário: existe uma solução ótima integral.

Cobertura por vértices em Bipartidos: outro exemplo

Dado um grafo $G = (V, E)$, uma cobertura por vértices é um conjunto $C \subseteq V$ tal que $G - C$ não tem arestas. Queremos encontrar uma cobertura por vértices $C \subseteq V$ de tamanho mínimo em grafo bipartido.



Formulação em Programação Linear Inteira:

$$\begin{array}{ll} \text{minimize} & \sum_{v\in E} y_v \\ \text{sujeito a} & \begin{cases} y_u + y_v \geq 1 & \forall (uv) \in E \\ y_v \in \{0,1\} & \forall v \in V \end{cases} \end{array}$$

De novo, a formulação acima só não é um programa linear devido a uma restrição de integralidade, em y_v.

Cobertura por vértices em Bipartidos: Relaxação Linear

Relaxação linear:

$$(CV)\quad \begin{array}{ll} \text{minimize} & \sum_{v\in E} y_v \\ \text{sujeito a} & \begin{cases} y_u + y_v \geq 1 & \forall (uv) \in E \\ 0 \leq y_v \leq 1 & \forall v \in V \end{cases} \end{array}$$

Teorema

Os vértices do politopo da relaxação linear são inteiros.

Prova: Exercício.

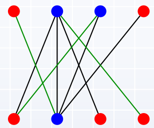
Corolário: existe uma solução ótima integral.

Investigando a dualidade de CV

Primal e dual de CV

$$(P)\quad \begin{array}{ll} \min & \sum_{v\in E} y_v \\ \text{s.a} & y_u + y_v \geq 1 \, \forall (uv) \in E \\ & 0 \leq y_v \leq 1 \, \forall v \in V \end{array} \qquad (D)\quad \begin{array}{ll} \max & \sum_{e\in E} x_e \\ \text{s.a} & \sum_{e\in\delta(v)} x_e \leq 1 \, \forall v \in V \\ & 0 \leq x_e \leq 1 \quad \forall e \in E \end{array}$$

O dual da relaxação (CV) é a relaxação do emparelhamento (E)!



Teorema (Kőnig)

Dado um grafo bipartido G, o tamanho do emparelhamento máximo é igual ao tamanho da cobertura por vértices mínima.

Prova: Segue da integralidade dos politopos e folgas complementares.