Programação Linear
Prof°. Antônio Carlos Moretti

Aula 20 - Método Simplex e Análise de Complexidade
Conceitos de Complexidade Computacional
Complexidade do Método Simplex
O Método dos Elipsóides
O Método dos Elipsóides para PL

1. Método Simplex e Análise de Complexidade

O Método Simplex, descrito ao longo do curso, é uma ferramenta computacional extremamente eficiente. Para certos problemas, no entanto, pelo menos na teoria, o método tem se mostrado ineficiente. Isto nos leva ao estudo da complexidade computacional da Programação Linear. A análise do pior dos casos mostra que o Método Simplex e suas variantes podem efetuar um número exponencial de trocas de bases para alcançar uma solução ótima e o método torna-se impraticável para se resolver problemas de Programação Linear de grande porte. Portanto, muito trabalho foi feito para desenvolver um algoritmo polinomial para PPL.

O primeiro algoritmo polinomial para PL foi proposto por L.G. Khachian, em 1979. Ele era baseado no *Método das Seções Centrais* e no *Método de Descida do Gradiente Generalizado com Espaço de Dilatação*, os quais foram desenvolvidos para otimização não-linear. Na teoria, o Método de Khachian, conhecido como o *Método dos Elipsóides*, tem um limitante superior para o tempo de execução melhor que o Método Simplex, mas isto parece ter pouco valor prático, uma vez que na prática a performance do Método Simplex é muito melhor que a do Método dos Elipsóides.

2. Conceitos de Complexidade Computacional

O conceito de análise de complexidade foi introduzido em 1970 para avaliar a performance de um algoritmo. A análise do pior dos casos mede o grau de dificuldade na resolução do problema sob a visão mais pessimista. A complexidade computacional nos dá um índice de acessar o crescimento do esforço computacional de um algoritmo como uma função do tamanho do problema na análise do pior dos casos. A complexidade de um algoritmo é medida neste contexto pelo número de operações elementares, tais como adições, multiplicações e comparações, as quais dependem do algoritmo e do tamanho total dos dados de entrada na representação binária.

Para um processo iterativo geral, sua complexidade é determinada pelo produto do número total de iterações e do número de operações em cada iteração. O número total de iterações certamente depende da precisão desejada, enquanto o número de operações elementares depende da representação binária dos dados de entrada.

Considere o PPL abaixo:

$$\left.\begin{array}{l} Min\ z = c^t x \\ sa \\ \qquad Ax = b \\ \qquad x \geq 0 \end{array}\right\} \quad (1)$$

onde $A \in \Re^{m\,x\,n}$ com $m, n \geq 2$ e seus elementos são todos inteiros (possivelmente convertidos de algum dado racional para esta forma). Ao especificar os valores de *m*, *n*, *A*, *b*, *c*, nós definimos um exemplar do PL. Se além disso, nós definimos o tamanho dos dados de entrada de m exemplar como sendo o número de bits necessários para codificar todos os dados dados do problema, e o denotamos por *L*, então o tamanho de um exemplar do problema pode ser representado pela tripla (*m, n, L*). Consequentemente, a complexidade de um algoritmo para resolver um PPL torna-se função de *m, n* e *L* , isto é, *f(m, n, L)*.

Suponha, agora, que exista uma constante $\tau > 0$, tal que o número total de operaçõs elementares requerido pelo algoritmo para resolver qualquer exemplar do problema não seja maior que τ *f(m, n, L)*.Neste caso, nós dizemos que o algoritmo é da ordem de complexidade *O (f(m, n, L))* .

Quando a função *f(m, n, L)* é uma função polinomial de *m, n* e *L,* o algoritmo é dito ser limitado polinomialmente ou apenas de complexidade polinomial. Caso contrário, o algoritmo é dito ser não-polinomial.

Note que no sistema binário, precisamos de $(r + 1)$ bits para representar um inteiro positivo $\xi \in \lfloor 2^r , 2^{r+1} \rfloor$, onde r é um inteiro não-negativo. Portanto, para um inteiro positivo ξ , nós precisamos de $\lceil \log(1+\xi) \rceil$ bits para representá-lo. Adicionando mais um bit para o sinal, temos um total de 1 + $\lceil \log(1+\xi) \rceil$ dígitos binários para codificar um inteiro arbitrário. Portanto, o tamanho dos dados de entrada do PL definido em (1) é dado por :

$$
\begin{aligned}
L = & \lceil 1 + \log(1+m) \rceil + \lceil 1 + \log(1+n) \rceil + \sum_{j=1}^{n} \left\{ 1 + \lceil 1 + \log(1 + |c_j\,) \rceil \right\} \\
& + \sum_{i=1}^{m} \sum_{j=1}^{n} \left\{ 1 + \lceil \log(1 + |a_{ij}|) \rceil \right\} + \sum_{i=1}^{m} \left\{ 1 + \lceil \log(1 + |b_i|) \rceil \right\}
\end{aligned}
$$

Em nossa análise de complexidade, como precisamos apenas de um limitante superior no esforço computacional, nós não precisamos definir de maneira exata o tamanho *L* do exemplar do problema.

Uma estimativa comum é dada por :

$$
L = \left\lceil \begin{array}{l} 1 + \log(m) + \log(n) + \displaystyle\sum_{j=1}^{n} \left\{ 1 + \log(1 + |c_j|) \right\} \\ + \displaystyle\sum_{i=1}^{m} \sum_{j=1}^{n} \left\{ 1 + \log(1 + |a_{ij}|) \right\} + \displaystyle\sum_{i=1}^{m} \left\{ 1 + \log(1 + |b_i|) \right\} \end{array} \right\rceil
$$

ou ainda

$$
L = \sum_{j=1}^{n} \lceil 1 + \log(1 + |c_j|) \rceil + \sum_{i=1}^{m} \sum_{j=1}^{n} \lceil 1 + \log(1 + |a_{ij}|) \rceil + \sum_{i=1}^{m} \lceil 1 + \log(1 + |b_i|) \rceil
$$

3. Complexidade do Método Simplex

A complexidade computacional do Método Simplex depende do número total de iterações e do número de operações elementares necessárias em cada iteração. Implementações diferentes resultam em complexidades diferentes. Variantes do Método Simplex foram feitas para conseguir melhores performances computacionais. Não é difícil estimar que o Método Simplex Revisado efetua $m(n-m)+(m+1)^2$ multiplicações e $m(n+1)$ adições em cada iteração. O ponto chave é que ambas operações são da ordem de *O (m, n)*.

Quantas iterações são necessárias? Cada iteração do Método Simplex vai de um ponto extremo para outro ponto extremo adjacente. Para um PPL na sua forma padrão tem-se

$$C(n,m)=\frac{n!}{m!(n-m)!}\geq\left(\frac{n}{m}\right)^m\geq 2^m \quad \text{para } n\geq 2m$$

pontos extremos, logo, é possível que para um determinado problema, nós tenhamos que visitar um número exponencial de pontos extremos, antes de alcançarmos a solução ótima.

O medo do esforço exponencial foi confirmado por exemplos especialmente "desenhados" para a análise do pior dos casos. O primeiro destes exemplos foi dado por V. Klee e G.L. Minty em 1971, para mostrar que o Método Simplex de Dantzig deveria percorrer todos os $\left(2^n-1\right)$ pontos extremos antes de alcançar a solução ótima.

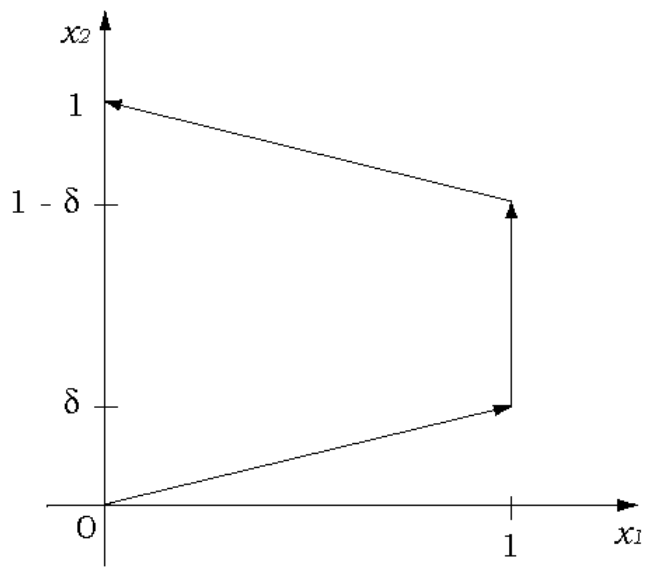
Exemplo 1 : (Klee & Minty : Ver artigo *"How good is the Simplex Algorithm?"* em *Inequalities III* , editor *O. Shisha, Academic Press, NY,* pags. 154-179, 1972).

Para $0<\delta<\frac{1}{2}$,

$$\begin{aligned} &\textit{Maximize} \quad x_n \\ &\quad sa \\ &\qquad\qquad 0 \le x_1 \le 1 \\ &\qquad \delta x_{i-1} \le x_i \le 1 - \delta x_{i-1} \quad , i = 2,3,\ldots,n \\ &\qquad\qquad x_i \ge 0 \end{aligned}$$

Obviamente, a origem é uma SBF. Se nós começarmos com a origem e aplicarmos como regra de entrada na base, a regra do melhor custo reduzido, então temos que o Método Simplex precisa de $\left(2^n - 1\right)$ iterações para visitar cada ponto extremos da região factível.

Para $n = 2$, temos:



Para $n = 2$, precisamos de $2^2 - 1 = 3$ iterações para alcançarmos o ponto ótimo.

Variantes do Método Simplex podem trocar as regras de entrada na base par evitar precorrer todos os pontos extremos. Mas, diferentes exemplos "patológicos" foram reportados para diferentes variantes. Assim, numa análise do pior dos casos, o Método Simplex tem complexidade exponencial.

Contudo, exemplos "patológicos" raramente ocorrem na maioria dos problemas reais. Têm sido observado que problemas reais de tamanhos moderados requerem de $4m$ à $6m$ iterações do Método Simplex para serem resolvidos.

4. O Método dos Elipsóides

Depois que o exemplo de Klee & Minty mostrou que o Método Simplex tinha uma complexidade exponencial no pior dos casos, ficou a pergunta: *"Existe um algoritmo polinomial para resolver um PPL ? "*

Uma resposta afirmativa foi dada por Khachian, em 1979. Ele mostrou como o Método dos Elipsóides usado para resolver problemas de Programação Convexa (da qual PL é um caso especial) pode ser adaptado para resolver um PPL em tempo polinomial.

Considere um sistema linear com n incógnitas e m inequações (estritas), isto é,

$$\left.\sum_{j=1}^{n} a_{ij}\, x_j < b_i \quad , \; i = 1,2,\ldots,m \right\} \quad (2)$$

ou simplesmente $Ax < b$, com $A \in \Re^{m\,x\,n}$, $b \in \Re^m$ e $x \in \Re^n$.

Nosso objetivo é achar uma solução de (2), se ela existir. O Método dos Elipsóides começa com uma esfera cujo raio é grande o suficiente para incluir a região factível (se ela existir) de (2). Denotando por P o conjunto de ponto factíveis de (2) que intersecionam o elipsóide inicial, o algoritmo prossegue construindo uma série de elipsóides, E_k, tais que $P \subseteq E_k$ .Os elipsóides são construídos de tal maneira que os seus volumes decrescem geometricamente. Como o volume de P é positivo quando $P \neq \emptyset$, nós podemos provar que depois de um número polinomial de iterações, o centro do elipsóide atual pertence a P ou concluiremos que não existe uma solução factível para P.

Descrição do Método dos Elipsóides em termos algébricos:

Dado um número não-negativo r e um ponto $z \in \Re^n$, uma esfera centrada em z com raio r é definida por:

$$S(z,r) = \left\{ x \in \Re^n : \sum_{i=1}^{n} (x_i - z_i)^2 \leq r^2 \right\} = \left\{ x \in \Re^n : (x - z)^t (x - z) \leq r^2 \right\}$$

O volume de $S(z, r)$ será denotado por $Vol(S(z, r))$.

Dada uma matriz A_{nxn} não-singular, e um ponto $c \in \Re^n$, uma transformação afim $T(A, c)$ transforma todo ponto $x \in \Re^n$ em um novo ponto $A(x - c) \in \Re^n$.

Um elipsóide é a imagem da esfera unitária $S(0,1)$ sob alguma transformação afim, isto é,

$$E = \left\{ x \in \Re^n : (x - c)^t A^t A (x - c) \leq 1 \right\}$$

O ponto c é o centro de E e o volume de E é dado por:

$$Vol(E) = \det\left(A^{-1}\right).Vol(S(0,1))$$

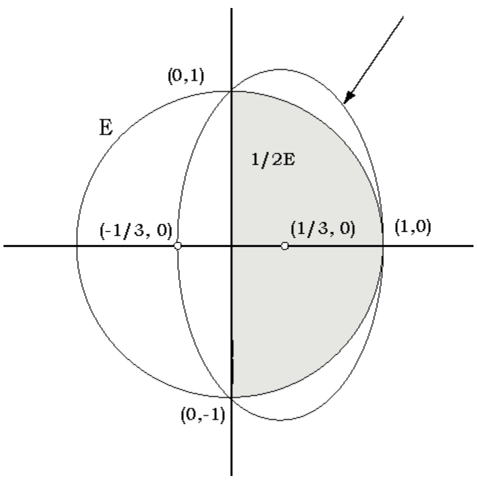
E além disso, vamos definir um semi-elipsóide como:

$$\frac{1}{2}E = \left\{ x \in E : a^t x \geq a^t c \right\}$$

Exemplo :

Na figura abaixo, $E = S(0,1)$ é uma esfera unitária no plano, a área hachurada é $1/2E$ dada pela intersecção de E com o semi-plano $\{(x_1, x_2) \in \Re^2 : x_1 \geq 0\}$.

Agora nós construímos um elipsóide $\overline{E}$, tal que $\overline{E}$ seja o elipsóide de volume mínimo que contém $1/2EI$.



$$\overline{E} = \left\{ x \in \Re^2 : \tfrac{9}{4}\left(x_1 - \tfrac{1}{3}\right)^2 + \tfrac{3}{4}x_2^2 \leq 1 \right\}$$

$$Vol(\overline{E}) = \tfrac{4\sqrt{3}}{9} Vol\left(S(0,1)\right)$$

O centro de $\overline{E}$ está em $(1/3 , 0)$ e a matriz que define o elipsóide é

$$A = \begin{bmatrix} 3/2 & 0 \\ 0 & 3/2 \end{bmatrix}$$

Nós podemos generalizar o resultado do exemplo acima para o caso n-dimensional.

Para $E = S(0,1) \subseteq \Re^n$ com $\frac{1}{2}E = \{x \in E : x_1 \geq 0\}$ podemos construir um novo elipsóide

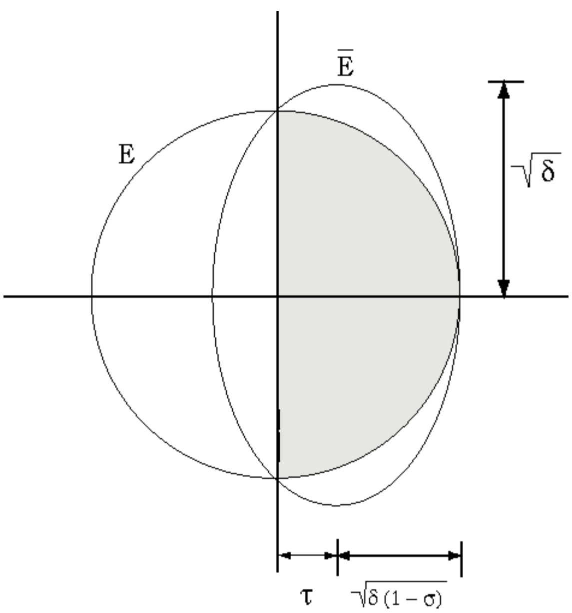
$$\overline{E} = \left\{ x \in \Re^n \; : \left(\frac{n+1}{n}\right)^2 \left(x_1 - \frac{1}{n+1}\right)^2 + \left(\frac{n^2-1}{n^2}\right) \sum_{i=2}^{n} x_i^2 \le 1 \right\}$$

cujo centro está em $\left(\frac{1}{n+1}, 0, \ldots, 0\right)$e tem volume

$$Vol(\overline{E}) = \left(\frac{n}{n+1}\right)\left(\frac{n^2}{n^2-1}\right)^{(n-1)/2} .Vol(E)$$

A matriz A associada é uma matriz diagonal n x n cujos elementos são $(n+1)/n$ como primeiro elemento diagonal e $\{(n^2+1)/n\}^{1/2}$ são os elementos restantes.

Na figura abaixo, o elipsóide E é determinado por três parâmetros:



$\tau = 1/(n+1)$
$\sigma = 2/(n+1)$
$\delta = n^2/(n^2-1)$

Comparando $\overline{E}$ à E, temos que $\overline{E}$ move seu centro da origem para $(\tau, 0, \ldots, 0)$, reduz na direção x_1 pelo fator $(\delta(1-r))^{1/2} = n/(n+1)$e se expande em todas as direções ortogonais pelo fator $\sqrt{\delta} = n/\sqrt{n^2-1}$. Portanto, nós chamamos τ, σ e δ como coeficientes de passo, dilatação e expansão, respectivamente.

Note que as transformações afins preservam as proporções de volumes e todo elipsóide pode ser transformado numa esfera unitária se usarmos uma transformação afim apropriada.

Agora, nós vamos enunciar os seguintes lemas:

Lema 1: Todo elipsóide $1/2E$ está contido em um elipsóide $\overline{E}$ cujo volume é $e^{-1/2(n+1)}$ vezes menor que o volume de E.

Para um conjunto poliédrico P que está contido em E, se o centro de E está fora de P, então P estaria contido e, algum elipsóide $1/2E$ e, consequentemente, em um elipsóide menor $\overline{E}$.

Lema 2: O menor elipsóide E que contém um conjunto poliédrico P tem seu centro em P .

O Lema 2 sugere um esquema iterativo para resolver um sistema de inequações lineares. Se o conjunto poliédrico P definido pelo sistema de inequações está contido em um elipsóide E_k na k-ésima iteração, então nós podemos checar se o centro de E_k pertence a P ou não. Se pertencer a P, então paramos o processo dando como resposta o centro de E_k ; caso contrário, substituimos E_k por um elipsóide menor $E_{k+1} = \overline{E}_k$ e repetimos o processo. O Lema 1 nos diz que o volume de E_k se reduz por no mínimo $e^{-1/2(n+1)}$ depois de cada iteração, portanto, este esquema iterativo requer apenas um número polinomial de iterações para encontrar em ponto factível o nos dar uma indicação de que tal ponto não existe.

Para começar com o processo iterativo, considere o seguinte resultado:

Lema 3: Se o sistema de inequações $Ax < b$ tem solução, então existe $x \in \Re^n$ tal que

$$-2^L \leq x_j \leq 2^L \qquad j = 1, 2, \ldots, n$$

onde L é o número de bits necessários para codificar os dados de entrada, com $c_j = 0$ para todo j.

Portanto, nós podemos definir $E_0 = S(0,2^l)$ como uma esfera n-dimensional de raio igual a 2^{2L}. Neste caso, o conjunto poliédrico P estará contido em E_0 e podemos começar o processo iterativo.

Para terminar o processo iterativo, usamos o seguinte resultado:

Lema 4: Se o sistema de inequações $Ax < b$ tem uma solução, então o volume de suas soluções dentro do cubo $\left\{x \in \Re^n : |x_i| \leq 2^L \;, \forall i\right\}$ é no mínimo $e^{-(n+1)L}$.

Assim sendo, nós podemos terminar o processo iterativo quando $Vol(E_k) < 2^{-(n+1)L}$. Neste caso, o sistema $Ax < b$ não tem solução.

5. O Método dos Elipsóides para PL

Considere o sistema linear $Ax < b$ descrito da seguinte maneira:

$$a_i^t x < b_i \quad , i = 1,2,\ldots,m \qquad (*)$$

Além disso, seja E_k um elipsóide definido por

$$\left\{x \in \Re^n : (x - x^k)^t B_k^{-1}(x - x^k) \leq 1\right\}$$

onde x^k é o centro de E_k e $B_k^{-1} = A_k^t A_k$.

Se x^k não é solução de (*), então existe um índice i tal que $a_i^t x \geq b_i$ e a solução está contida em

$$\tfrac{1}{2}E_k = \left\{x \in E_k : a_i^t x \leq a_i^t x^k\right\} = \left\{x \in E_k : -a_i^t x \geq -a_i^t x^k\right\}$$

Baseado nos parâmetros de passo, dilatação e expansão, temos que o novo elipsóide $\overline{E}_k$ é da forma:

$$x^{k+1} = x^k - \tau\left(B_k a_i / \sqrt{a_i^t B_k a_i} \right) \qquad (1)$$

e

$$B_{k+1} = \delta\left(B_k - \sigma\left(B_k a_j \left(B_k a_j\right)^t / \left(a_i^t B_k a_i\right)\right)\right) \qquad (2)$$

Como B_k^{-1} é simétrica e definida positiva, podemos mostrar que B_{k+1}^{-1} também será e o conjunto

$$\left\{ x \in \Re^n : (x - x^k)^t B_k^{-1} (x - x^k) \leq 1 \;,\; a_i^t x < b_i \right\}$$

está contido no elipsóide

$$E_{k+1} = \left\{ x \in \Re^n : (x - x^{k+1})^t B_{k+1}^{-1} (x - x^{k+1}) \leq 1 \right\}$$

E, o volume de E_{k+1} será dado por

$$\left(\frac{n}{n+1} \right)\left(\frac{n^2}{n^2 - 1} \right)^{(n+1)/2} Vol\left(E_k \right)$$

<u>Algoritmo Passo a Passo</u>

Passo 1: Faça $k = 0$

$E_k = S(0, 2^{2L})$

$B_k = 2^{2L} I$

$x^k = 0$

Passo 2: Se x^k satisfaz $Ax < b$, então pare.
Caso contrário, identifique um índice i tal que $a_i^t x^k \geq b_i$, calcule x^{k+1} e B_{k+1} de acordo com (1) e (2).
Faça $k = k + 1$.

Passo 3: Se $Vol(E_k) < 2^{-(n+1)L}$, então pare, com a conclusão de que $Ax < b$ não tem solução.
Caso contrário, volte ao Passo 2.

Lembre-se que depois de cada iteração, o volume de E_k é reduzido, no mínimo, por um fator de $e^{-1/2(n+1)}$. Como o volume inicial de S_0 é $Vol(S(0, 2^{2L}))$ e o volume do elipsóide final é $2^{-(n+1)L}$, temos que o número total de iterações é no máximo $O(n^2L)$.

Como observamos ao longo deste texto, o Método dos Elipsóides resolve um problema de factibilidade linear, portanto, para se resolver um PPL nós precisamos resolver o sistema linear formado pelas condições de otimalidade de KKT:

$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \end{aligned}$$

$$\begin{aligned} A^t w &\leq c \\ w &\geq 0 \end{aligned}$$

$$c^t x - b^t w \leq 0$$

Se existe uma solução x^* factível para este sistema, então x^* é solução ótima do problema

$$\begin{aligned} &Min\ z = c^t x \\ &\quad sa \\ &\qquad Ax \geq b \\ &\qquad x \geq 0 \end{aligned}$$