TÉCNICAS ALTERNATIVAS PARA RESOLUÇÃO DE UM PROBLEMA DE PROGRAMAÇÃO LINEAR *

** Adriana Carlos Paz

*** Charles Bezerra Martins

*** Neudson Cavalcante Albuquerque

*** Rodrigo Moraes L. de Araújo Costa

**** Plácido Rogério Pinheiro

O principal objetivo deste artigo é mostrar técnicas alternativas de solução de um problema de Programação Linear diferentes do método Simplex. Depois de algumas considerações sobre complexidade de algoritmos, é exposto o Método das Projeções; o Método de Khachiyan, o Método Simplex Dividido e o Algoritmo de Karmarkar.

ABSTRACT

The principal purpose of this article is to show alternative techniques to solve a Linear Programming Problem different from Simplex Method. After some considerations about complexity of algorithms, we talk about the Projection Method, the Khachiyan's Method, The Simplex Splitting Method and the Karmarkar's Algorithm.

INTRODUÇÃO

Numa época em que os avanços tecnológicos fascinam as pessoas com o vislumbramento de novos e diversos horizontes, dia após dia, não só o novo é objeto de pesquisa. O ser humano não mais pode se dar ao luxo de apenas buscar o novo, mas, num mundo onde os recursos devem ser cada vez mais divididos, quando não totalmente escassos, a racionalização do que já existe é ponto vital para a manutenção dos espaços já conquistados. Neste interim, uma palavra soa com personalidade, a otimização cuja essência encontra-se intimamente ligada ao campo da Pesquisa Operacional.

Novamente, a matemática se presta como ferramenta para a solução de problemas de nosso meio, ora feita, alicerçada pela computação eletrônica — viabilizando o grande montante de cálculos necessários. Em particular, merece destaque nessa área a Programação Linear, que modela grande parte dos problemas relacionados com a otimização.

A partir da Segunda Grande Guerra, as pesquisas tomaram grande impulso. Até meados da década de 80, o seu principal expoente foi o algoritmo Simplex, de Dantzig (6).

Depois de modelado o problema em equações lineares e definida a função objetivo (Problema de Programação Linear—PPL), o Método Simplex busca a solução ótima percorrendo a região determinada pelo conjunto de soluções desse sistema (politopo), através de seus pontos extremos (vértices). Embora apresentando desempenho satisfatório na maioria das situações, o algoritmo Simplex demonstra problemas de convergência (?). Em razão disso, tem-se buscado o seu aperfeiçoamento, como o Simplex Revisado (2); mas, também, simultaneamente, tem-se pesquisado o desenvolvimento de novos métodos, que tem dividi-

---

* *Este trabalho faz parte de um projeto de iniciação científica parcialmente financiado pela Fundação Universidade Estadual do Ceará.*

** *Graduada em Ciência da Computação (UECe), aluna do Curso de Mestrado da PUC-RJ*

*** *Aluno de Bacharelado em Ciência da Computação (UECe)*

**** *Mestre em Matemática pela UFC e Prof. do Departamento de Matemática da UECe e da UNIFOR.*

do espaço com o Simplex, criando novas perspectivas de soluções ótimas.

Nesse artigo, iremos apresentar sucintamente a evolução dos algoritmos de pontos interiores, partindo do Método das Projeções de Agmon (1), passando pelas Elipsóides de Khachyian, pelo Simplex Dividido de Levin, e, por fim, o Algoritmo de Karmarkar, que é apontado como um possível substituto para o Método Simplex.

COMPLEXIDADE

Antes de iniciarmos o estudo dos métodos propostos, vamos apresentar algumas noções de complexidade de algoritmos.

Podemos afirmar que a análise de um algoritmo utiliza um conjunto de critérios de avaliação cujo objetivo é fornecer uma série de informações a cerca da performance de um algoritmo de um método específico.

Os critérios utilizados são o tempo requerido para resolver um problema com um dado método e a quantidade de espaço de armazenamento necessário para representar o problema e seus resultados intermediários na memória de um computador.

O fator tempo torna-se importante pois dá uma idéia do esforço demandado para a computação do método já que o tempo é diretamente proporcional ao custo de computação.

Passaremos agora a definir tamanho de um problema, que é a quantidade de símbolos necessários para resolvê-lo.

Para muitos problemas de programação matemática, basta medir o tamanho do problema em termos de n (número de variáveis) e m (número de restrições).

Contudo, apenas o tamanho de um problema não oferece ao usuário uma indicação precisa de quanto tempo o computador levará para resolver o problema: um programa pequeno pode levar mais tempo para encontrar uma solução do que um programa grande.

Para superar esta dificuldade, pensou-se então em distinguir entre o tempo requerido no caso mediano (na média) e no pior caso. Esta maneira, porém, não seria confiável, visto que, dependeria do tempo de processamento do equipamento, sendo portanto, um cálculo que variaria de computador para computador.

Com o objetivo de conseguir uma expressão que fosse invariável em relação ao computador utilizado, o tempo necessário foi substituído pelo número de operações elementares requeridas.

As operações elementares são adição, subtração, multiplicação, divisão, comparação e raiz quadrada. O número de operações elementares requerido para a solução de um problema com um dado algoritmo pode ser calculado como o número de operações elementares por iteração, multiplicado pelo número de iterações requeridas no pior caso.

C (n) = OPER = ELEM * W-ELEM

OPER = número de operações necessárias

ELEM = número de operações necessárias por iteração

W-ELEM = número de iterações necessárias no pior caso (Worst case)

Esta expressão é chamada de função tempo — complexidade C (n) onde "n" denota o tamanho do problema.

Se C (n) é quando muito uma função polinomial, o algoritmo em questão será chamado polinomialmente limitado, caso contrário, o algoritmo é referenciado como exponencialmente limitado.

Para simplificar os problemas, apenas aquele termo que é responsável pelo maior crescimento de C (n) é que irá determinar a ordem da função tempo-complexidade.

Exemplo:

$$\text{Se } C(n) = 1600n^5 + 2000n^4 + 1000n^2$$

Então o algoritmo que irá solucionar a função é polinomialmente limitado na ordem de n^5 ou em $0(n^5)$.

No início de 1965, Edmonds chamou de eficientes os algoritmos polinomialmente limitados e de ineficientes os exponencialmente limitados. A razão para essa afirmação pode ser atestada no exemplo seguinte:

Suponha que temos um problema com tamanho de n = 60 (um problema pequeno para os padrões práticos) e assuma que há 2 algoritmos que solucionam esse problema: o primeiro com $c(n) = n^6$ (polinomialmente limitado) e o segundo com $C(n) = 2^n$ (portanto, exponencialmente limitado). Suponha ainda que um dado computador gasta um nanossegundo, isto é, 10^{-9}s por operação elementar. Observa-se que o algoritmo $0(n^6)$ levaria 47,7s, enquanto que o algoritmo $0(2^n)$ levaria 36,6 anos para a solução do problema proposto.

Vale a pena observar que o fato de um algoritmo ser $0(2^n)$ significa que pelo menos uma instância do problema requer 2^n passos, no entanto existem algoritmos exponenciais bem sucedidos na prática. Por exemplo, mostra-se (Schrijver, 1986) (?) que o Problema de Programação Linear a seguir requer 2^n iterações do algoritmo do simplex para a obtenção da solução ótima:

$$\begin{aligned}
&\text{Maximizar } 2^{n-1}X_1 + 2^{n-2}X_2 + \ldots + 2X_{n-1} + X_n \\
&\text{Sujeito a } X_1 && \leqslant 5 \\
&\quad 2^2X_1 + X_2 && \leqslant 5^2 \\
&\quad 2^3X_1 + 2^2X_2 + X_3 && \leqslant 5^3 \\
&\quad 2^nX_1 + 2^{n-1}X_2 + \ldots + 2^2X_{n-1} + X_n && \leqslant 5^n \\
&\quad X_1, X_2, \ldots X_n \geqslant 0
\end{aligned}$$

No entanto o algoritmo do simplex tem, na prática, comportamento médio polinomial." (7)

MÉTODO DAS PROJEÇÕES

Surgido em 1954, oriundo das pesquisas de Agmon, Motzkin e Schonberg (13), este método marca o início da resolução de PPL's por meio de algoritmos não-simplex.

Este método visa encontrar um ponto viável num politopo dado, ou seja, uma solução viável para um sistema de inequações ou equações lineares simultâneas.

A idéia básica é começar com um ponto qualquer $X^0 \in R^n$ e determinar se ele é ou não viável. Se o for, o ponto atende às exigências, e o procedimento acaba. Caso contrário, traça-se uma projeção ortogonal de X^0 ao hiperplano violado (ou a um deles, caso exista mais de uma violação), dando origem a um novo ponto X^1, que é a nova solução.

Este procedimento é repetido quantas vezes for necessário.

PASSO 1: $Ax^k <= b$?

Caso afirmativo → Parar, x^k é uma solução para o problema.

Caso negativo → Selecione a i-ésima restrição que satisfaz a $ai.x^k > bi$ e vá para o PASSO 2.

PASSO 2: Calcule $x^{k+1} = x^k + \dfrac{bi - ai.x^k}{ai ai.T}\, ai T$

faça $k = k + 1$

vá para o PASSO 1

Segue um exemplo numérico:

I: $-x1 - x2 \leqslant -6$ Tal que $a1. = [-1, -1]$ $b1 = -6$

II: $1{,}5x1 - 0{,}5x2 \leqslant 3$ $a2. = [1{,}5 - 0{,}5]$ $b2 = 3$

III: $-x2 \leqslant -2$ $a3. = [0, -1]$ $b3 = -2$

Fazendo $x^0 = [0\ 0]^T$

$$x^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{-6 - [-1 -1]\,[0\ 0]^T}{[-1, -1]\,[-1, -1]} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -3 \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Que não viole nenhuma das restrições, sendo portanto, o vetor solução. Deve-se observar no entanto que nem sempre uma solução sai logo da primeira itereção.

Motzkin e Schonberg generalizaram o procedimento de Agmon, acrescentando um certo $\alpha \in)0; 2)$.

$$x^{k+1} = x^k + \alpha . \frac{bi - ai . x^k}{ai . ai . T} . ai . T$$

É lógico que para $\alpha = 1$, a nova relação coincide com a fórmula de Agmon e um ponto x^k é projetado diretamente sobre um hiperplano Hi, mas, para $\alpha = 1/2$, o próximo ponto seria projetado no meio do caminho entre x^n e sua projeção sobre o hiperplano Hi (isto é, uma "subprojeção" para $\alpha \in$) o; 1) e para $\alpha = 2$, o próximo ponto seria projetado num ponto situado a duas distâncias do ponto original ao hiperplano (isto é, uma sobreprojeção para $\alpha \in)1; 2)$).

Motzkin e Schönberg provaram que para $\alpha = 2$, a série de pontos gerada é finita, se o conjunto de soluções viáveis incluir pelo menos um ponto interior.

Observação: muitas vezes, não apenas uma restrição violada é selecionada, mas aquela em que a distância Euclidiana entre o ponto corrente x^k e o hiperplano é máxima.

MÉTODO DAS ELIPSÓIDES

O método das elipsóides, desenvolvido por Khachyian (?) em 1979 foi considerado um grande avanço no mundo científico, visto que foi a primeira vez em que se tornou possível resolver problemas de programação linear em tempo polinomial. Autores como Gács e Lovász (1981); Murray (1979); Aspvall e Stone (1979); Price (1981); Bland, Golldfarb e Todd (1980); e Schrader (1983) publicaram artigos com explicações, interpretações e refinamentos do método a partir de trabalhos de Shor (1970). Uma bibliografia inicial pode ser encontrada em Wolfe (1980).

Respaldado pelo teorema forte das folgas complementares, o Método de Khachiyan encontra a solução ótima de um sistema de inequações lineares. Originalmente projetado para encontrar soluções viáveis para um sistema de inequações do tipo $Ax < b$, caso tal solução exista, o Método das Elipsóides foi adaptado para solucionar o problema na seguinte forma:

Sejam os PPL's primal e dual abaixo:

maximizar $z = c^t x$	minimizar $w = b^t y$
sujeito a $Ax \leqslant b$	sujeito a $A^t y \geqslant c$
$x \geqslant 0$	$y \geqslant 0$

Sabemos do teorema forte da dualidade que sendo x^* e y^* soluções ótimas de seus respectivos PPL's, caso $c^t x^* = b^t y^*$, podemos transformar o PPL em um problema do tipo:

$$P^*: \begin{aligned} Ax &\leqslant b \\ x &\geqslant 0 \\ A^t y &\geqslant c \\ y &\geqslant 0 \\ c^t x &= b^t y \end{aligned}$$

que graças ao teorema das folgas complementares permite ser resolvido pelo Método das Elipsóides, obtendo um ponto que além de ser viável, é ponto ótimo.

Para tanto, o algoritmo de Khachiyan obedece ao seguinte procedimento:

"A é uma matriz de dimensão (m, n) e todos parâmetros Aij e Bi são inteiros. Nós começamos com um ponto $x^0 = (0, 0\ldots, 0)^t$ e a hiperesfera centrada em x^0 grande suficiente para conter pelo menos um dos pontos viáveis caso ele exista. Suponhamos que o ponto corrente x^k não é viável—caso contrário o algoritmo termina—e que o elipsóide corrente E^k (a hiperesfera inicial é uma elipsóide especial) contém pelo menos um ponto viável. Desde que x^k não é ponto viável, existe ao menos uma restrição violada, suponha $A_i . x^k > b_i$ no hiperplano Hi. Agora, Hi é deslocado paralelamente até x^k, resultando em Hi de forma que Hi corta E^k em duas metades: uma chamada $E^{k}+$, que inclui todos os pontos viáveis de E^k (caso existam); e outra chamada $E^{k}-$, que não contém nenhum ponto viável. Seja Hi o hiperplano paralelo a Hi que é tangente a $E^{k}+$, então um novo elipsóide E^{k+1}, com centro x^{k+1}, é construído, de modo que a casca de E^{k+1} inclua os seguintes pontos: a interseção de Hi com E^k e a de Hi com $E^{k}+$ (note que E^{k+1} está contido em $E^{k}+$). Então, x^{k+1} é um novo ponto, e o procedimento é repetido se necessário.

Para a execução deste procedimento, algumas questões devem ser levantadas, como o tamanho da hiperesfera inicial, e considerações de convergência.

De modo a encontrar uma expressão para o tamanho da hiperesfera inicial, devemos a priori especificar o número de símbolos requeridos para a codificação de um problema determinado em um computador digital, obedecendo a representação binária. Chamemos essa quantidade de L; e seja um sistema de ordem (m, n), teremos

$$L = \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\log 2 \, |A_{ij}|\right] + \sum_{i=1}^{m} \left[\log 2 \, |b_i|\right] + \left[\log 2 \, |mn|\right] + 2m(n+1) + 4$$

Este número ajudará a determinar o tamanho da esfera, que deve ser ao mesmo tempo grande o suficiente para conter ao menos um ponto viável e não tão grande que torne muito lento o processo de convergência. Esse valor ótimo para o volume inicial da esfera foi determinado como $V = 2^L$. [7]

A partir deste valor, o Método das Elipsóides deverá convergir em no máximo 6n (n + 1)L iterações (7) dos seguintes passos:

PASSO 1: x^k é solução viável?

Caso afirmativo ⇒ pare, x^k é uma solução para o problema P;

Caso negativo ⇒ vá para o passo 2.

PASSO 2: $K > 6n(n+1)L$?

Caso afirmativo ⇒ pare, não existe solução viável;

Caso negativo ⇒ vá para o passo 3.

PASSO 3: Suponha que a i-ésima restrição foi violada, isto é. $A_i.x^k \geqslant b_i$. Então determine:

$$x^{k+1} = x^K - \frac{1}{n+1} \frac{d^k A_i.^t}{A_i.D^k A_i.^t} \in$$

$$D^{K+1} = \frac{n^2}{n^2+1}\left[D^k - \frac{2}{n+1} \frac{|D^k A_i.^t| \, | D^k A_i.^t|^t}{A_i.D^k A_i.^t}\right]$$

faça k: = K + 1 e vá ao passo 1

(Note que o numerador na última razão da segunda fórmula é o produto de um vetor coluna com um vetor linha).

EXEMPLO: O pequeno exemplo a seguir deve explicar o procedimento acima. Encontre uma solução viável para o problema:

P: $x1 - x2 < -1$

$-x1 \quad < -1$;

Isto é, A1. = (1, −1); A2. = (−1, 0); b1 = −1; b2 = −1; m = 2 e n = 2. Para acelerar o procedimento, faremos L : = 5, sabendo que o círculo com o centro na origem, e raio $2^L = 32$ ainda inclui pontos viáveis. A solução inicial é $x^0 = (0, 0)^t$ e $D^0 = 32I$ (onde I é a matriz identidade); é visível que a solução corrente viola ambas restrições e, então, escolhemos arbitrariamente a primeira. Então:

$$x^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{3} \frac{\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}}{\sqrt{[1, -1]\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}}} = \begin{bmatrix} -4/3 \\ 4/3 \end{bmatrix} \text{ e}$$

$$D^1 = \frac{4}{3}\left[\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix} - \frac{2}{3} \frac{\left[\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right]\left[\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right]^J}{[1, -1]\begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}}\right] = \begin{bmatrix} \frac{256}{9} & \frac{128}{9} \\ \frac{128}{9} & \frac{256}{9} \end{bmatrix}$$

x1 satisfaz a primeira restrição mas viola a segunda; portanto calculamos

$$x^2 = \begin{bmatrix} -4/3 \\ 4/3 \end{bmatrix} - \frac{1}{3} \frac{\begin{bmatrix} 256/9 & 128/9 \\ 128/9 & 256/9 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}}{\sqrt{[-1, 0]\begin{bmatrix} 256/9 & 128/9 \\ 128/9 & 256/9 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}}} = \begin{bmatrix} \frac{4}{9} \\ \frac{20}{9} \end{bmatrix} \text{ e}$$

$$D^2 = \frac{4}{3}\left[\begin{bmatrix} \frac{256}{9} & \frac{128}{9} \\ \frac{128}{9} & \frac{256}{9} \end{bmatrix} \frac{-2}{3} \frac{\left[\begin{bmatrix} \frac{256}{9} & \frac{128}{9} \\ \frac{128}{9} & \frac{256}{9} \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right]\left[\begin{bmatrix} \frac{256}{9} & \frac{128}{9} \\ \frac{128}{9} & \frac{256}{9} \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right]^J}{[-1, 0]\begin{bmatrix} \frac{256}{9} & \frac{128}{9} \\ \frac{128}{9} & \frac{256}{9} \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}}\right] = \begin{bmatrix} \frac{1024}{81} & \frac{512}{81} \\ \frac{512}{81} & \frac{2560}{81} \end{bmatrix}$$

a solução x^2 obtida ainda viola a segunda restrição, logo, teremos:

$$x^3 = \begin{bmatrix} 4/9 \\ 20/9 \end{bmatrix} \frac{-1}{3} \frac{\begin{bmatrix} \frac{1024}{81} & \frac{512}{81} \\ \frac{512}{81} & \frac{2560}{81} \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}}{\sqrt{[-1, 0]\begin{bmatrix} 1024/81 & 512/81 \\ 512/81 & 2560/81 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}}} = \begin{bmatrix} \frac{44}{27} \\ \frac{76}{27} \end{bmatrix} = \begin{bmatrix} 1\,\frac{17}{27} \\ 2\,\frac{22}{27} \end{bmatrix} \approx \begin{bmatrix} 1,63 \\ 2,81 \end{bmatrix}$$

que é uma solução viável, e, portanto, dispensamos o cálculo de D^3.

Para uma maior eficiência, sugerimos que se trabalhe com matrizes particionadas, já que a implementação do algoritmo ainda oferece certas dificuldades. Entretanto, da mesma forma que somente após o desenvolvimento das técnicas de inversão numérica de matrizes grandes e médias o algoritmo Simplex obteve sua robustez e maior utilidade; cremos que o algoritmo de Khachyian se reforçará com os avanços em campos correlatos por ele abertos, como o da Matemática Aplicada e da Análise Numérica.

MÉTODO DO SIMPLEX DIVIDIDO

A técnica do Simplex Dividido, introduzida por A. Ju Levin (1965) e depois desenvolvida por L.A. Levin e Boris Yamnitsky, resolve um sistema linear utilizando estruturas lineares, os Simplices, ao contrário do Método de Khachiyan, que utiliza elipsóides, ou seja, estruturas não-lineares. A novidade apresentada por Yamnitsky-Levin deve-se a implementação do método com complexidade polinomial.

Inicialmente, a definição de Simplex faz-se necessária, haja vista encontrar-se disposta em um contexto diferente do associado ao termo quando nos referimos ao Método Simplex propriamente dito.

Um conjunto de pontos Vo, V1, . . . , Vn em R^n, é dito estar numa posição geral, quando nenhum hiperplano passa por eles; a casca convexa formada por esse conjunto de pontos é chamada de Simplex. Um Simplex em R^2 é um triângulo, e um Simplex em R^3 é um tetraedro.

O método inicia com um Simplex $S^{\emptyset}$ possuindo, no mínimo, um ponto viável Definimos P: (x1 $Ax \leq$

b) como o politopo aberto formado pelas restrições, assim $P \cap S^{\emptyset} = (\ )$. O centro de qualquer Simplex S^k, $k = \emptyset \ldots n$, com vértices vo, v1, ..., vn é um ponto $x^k = \frac{1}{n+1} \sum_{l=\emptyset}^{n}$ ve O Simplex inicial $S^{\emptyset}$ e seu centro são definidos pelo método dos elipsóides correspondendo a hiperesfera inicial e sua origem, respectivamente. A hiperesfera inicial de Khachiyan deve ser circunscrita por algum Simplex com o objetivo de se obter o Simplex $S^{\emptyset}$. O método termina quando o centro do Simplex corrente é um ponto viável.

Supondo uma iteração qualquer onde o Simplex é S^k e seu centro x^k, tal que x^k viola no mínimo uma restrição. Então, o hiperplano que violou qualquer uma das restrições é deslocado em direção a x^k, dividindo o Simplex S^k em um Simplex S^{k+} e um tetraedro T^{k-}, onde S^{k+} contém todos os pontos viáveis de S^k, e T^{k-} não contém pontos viáveis. Agora S^{k+} é circunscrito por um novo Simplex S^{k+1} e a relação de volume obtida:

$$v(S^k) < v(S^{k+1})$$

O centro do Simplex S^{k+1}, x^{k+1}, é determinado, a sua viabilidade testada e o procedimento é repetido até que o centro do Simplex corrente seja viável.

Vamos agora colocar o algoritmo em uma maneira mais formal. Consideremos que uma solução viável deva ser encontrada para o sistema geral $Ax \leq B$ $x \in R^n$. Tomemos o Simplex inicial com vértices $v^{\emptyset}$, $v^1, \ldots, v^n$, incluindo no mínimo um ponto viável. O centro do Simplex é:

$$x^{\emptyset} = \frac{1}{n+1} \sum_{l=0}^{n} v^l$$

Iniciaremos com $k = 0$. O algoritmo do método pode ser estabelecido:

PASSO 1: Se $Ax^k < B$ então
Caso afirmativo ⇒ Parar, x^k é uma solução para o problema
Caso negativo ⇒ Selecione a i-ésima restrição onde $A_i, x^k > b_i$ e vá para o PASSO 2.
PASSO 2: Calcule $S_i(v^l)$, a folga do i-esimo vértice com respeito i-ésima restrição da seguinte forma

$$S_i(v^l) = A_i.(x^k - v^l),\ \ell = 0,1,\ldots,n)$$

e determine u tal que $S_i(v^u) = \max_l (S_i(v^l))$.

PASSO 3: Calcule $\lambda^l = 1 - \frac{S_i(v^l)}{n^2 S_i(v^u)}$ $\ell = 0,1 \ldots, n$

e determine os vértices do novo Simplex como

$$v^l = v^u + \frac{1}{\lambda^l}(x^l - v^u),\ \ell = \emptyset,1,\ldots,n$$

O centro do novo Simplex será:

$$x^{k+1} = \frac{1}{n+1} \sum_{l=0}^{n} v^l$$

Atualize $K = k+1$ e vá para o PASSO 1.

A convergência do Método do Simplex Dividido depende do fato de que os Simplices estão "encolhendo". Este fato pode ser observado pela razão dos volumes de dois Simplices sucessivos:

$$Rs(n) = \frac{v(s^{k+1})}{v(s^k)} = \left[\frac{N-1}{N}\right]\left[\frac{N^2}{N^2-1}\right]^N < e^{-\frac{1}{2(N+1)^2}} < 1$$

Vejamos um exemplo. Considere o sistema linear abaixo:

$$\begin{aligned} P:\ & x1 - x2 < (\delta 1) \\ & -x1 < -1\ (\delta 2) \\ & x1, x2 \in R \end{aligned}$$

Tomemos o simplex inicial com os seguintes vértices:

$$v_0 = [8, 2.5]^T,$$

$v1 = [-1{,}5{,}7.5]^T$ e $V^2 = [1,-1]^T$ com centro

$$x^o = \frac{1}{3}\left(\begin{bmatrix} 8 \\ 2.5 \end{bmatrix} + \begin{bmatrix} -1{,}5 \\ 7{,}3 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$$

Como o ponto X^0 viola a primeira restrição, então $i = 1$ e as folgas dos três vértices são calculadas:

$$S1(v^a) = A1.(x^o - v^o) = [1,-1]\left(\begin{bmatrix} 2.5 \\ 3 \end{bmatrix} - \begin{bmatrix} 8 \\ 2.5 \end{bmatrix}\right) = -6$$

$$S1(v^1) = A1.(x^o - v^1) = [1,-1]\left(\begin{bmatrix} 2.5 \\ 3 \end{bmatrix} - \begin{bmatrix} -1.5 \\ 7.5 \end{bmatrix}\right) = 8.5$$

$$S1(v2) = A1.(x^o - v2) = [1,-1]\left(\begin{bmatrix} 2.5 \\ 3 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) = -2.5$$

A maior folga é $S_1(V^1) = 8.5$, assim, $\mu = 1$ e os deslocamentos são:

$$\lambda^0 = 1 - \frac{S1(v^0)}{n^2 S1(v^1)} = 1 - \frac{(-6)}{4.8.5} = \frac{20}{17}$$

$$\lambda^1 = 1 - \frac{S1(v^1)}{n^2 S1(v^1)} = 1 - \frac{8.5}{4.8.5} = \frac{3}{4}$$

$$\lambda^2 = 1 - \frac{S1(v^2)}{n^2 S1(v^1)} = 1 - \frac{(-2.5)}{4.8.5} = \frac{73}{68}$$

Calculemos os novos vértices do simplex como:

$$v^o = v^1 + \frac{1}{\lambda o}(v^o - v^1) = \left[\frac{263}{40}\ \frac{130}{40}\right]^T = [6{,}575{,}3.25]^T$$

$$v^1 = v^1 + \frac{1}{\lambda 1}(v^1 - v^1) = [-1{,}5{,}7{,}5]^T$$

$$v^2 = v^2 + \frac{1}{\lambda 2}(v^2 - v^1)\ \left[\frac{121}{146}\ \frac{-61}{140}\right] = [0{,}829,-0{,}418]^T$$

o novo centro do simplex S^1 e $X^1 = \frac{1}{3}$ •

$$\cdot\left(\begin{bmatrix} 263/40 \\ 130/40 \end{bmatrix} + \begin{bmatrix} -3/2 \\ 15/2 \end{bmatrix} + \begin{bmatrix} 121/146 \\ -61/146 \end{bmatrix} \cong \begin{bmatrix} 1.968 \\ 3.444 \end{bmatrix}\right.$$

Como λ^1 é visível, o procedimento termina.

ALGORITMO DE KARMARKAR

Dentre os algoritmos que buscam resolver Problemas de Programação Linear, este é considerado um marco, pois representou um salto de qualidade do ponto de vista teórico e prático.

Do ponto de vista teórico, um algoritmo cuja complexidade é polinomial é melhor que outro, que solucione o mesmo problema, com complexidade exponencial (g). Antes do algoritmo de Karmarkar já existia o algoritmo Simplex (15), ainda hoje amplamente difundido e utilizado em todo o mundo. Entretanto, sua complexidade no pior caso é exponencial, que é uma característica indesejável. Já existia também, antes de Karmarkar, o algoritmo de Khachiyan, que é polinomial mas não pode ser praticado devido a dificuldade de implementá-lo. A grande vantagem do algoritmo de Karmarkar é ser polinomial e implementável, tornando-o superior ao Método Simplex.

Karmarkar apresentou seu método no artigo "A New Polinomial-Time Algorithm for Linear Programming" (10) em 1984. Entretanto, para uma leitura inicial é indicado o artigo "Karmarkar's Algorithm" (14).

O algoritmo de Karmarkar não resolve diretamente um PPL, e sim um problema onde o conjunto de soluções está restrito a interseção de um simplex (que já foi definido nesse artigo na seção MÉTODO DO SIMPLEX DIVIDIDO) e um sistema de equações lineares. O passo inicial é inscrever uma circunferência no simplex de forma que seus centros coincidam. A interseção entre a circunferência e a região limitada definirá o ponto procurado assim que uma projeção da função objetivo indicar a direção a ser tomada. O passo iterativo é projetar o simplex em outro, de forma que o ponto encontrado seja projetado no centro do simplex imagem. Acha-se um ponto melhor da mesma forma que no passo inicial e projeta-o de volta ao simplex original. O passo é repetido com os sucessivos pontos encontrados até que um parâmetro de término seja alcançado.

O problema restrito tem a seguinte forma:

minimizar $c^t x$
sujeito a $x \in \Omega \cap \Delta^n$

onde c, x pertencem a R^{n+1} e $\Omega = (x \mid Ax = 0)$ é o espaço solução de um sistema homogêneo de equações lineares, e $\Delta^n = (x \mid \sum_{i=1}^{n} xi = 1)$ é um simplex de ordem n contido no R^{n+1}.

Ainda nesse problema, tem-se que observar as seguintes condições:

1. O valor mínimo da função objetivo é zero
2. O centro do simplex é um ponto a_0 e é viável.

O algoritmo para quando encontra um valor x tal que

$$\frac{C^t x}{C^t a\emptyset} < 2 - q,$$

onde q é um parâmetro maior que zero fornecido ao programa.

Para resolver um PPL pelo método de Karmarkar, temos que transformá-lo no problema restrito acima, observando as duas condições.

Quando o valor mínimo da função objetivo não é zero, temos que usar um artifício. Suponha que M seja o valor mínimo. A nova função objetivo será $(c - Met)^t x$, onde e é o vetor unitário. Não trataremos o caso em que o valor M é desconhecido (14).

Para reduzir um PPL ao problema restrito, usa-se uma transformação que leva pontos do R^n ao R^{n+1} da seguinte forma:

$x1 = (u1/a1)/(u1/a1 + \ldots + un/an + 1)$
$xn = (un/an)/(u1/a1) + \ldots + un/an + 1)$
$x_{n+1} = 1/(u1/a1 + \ldots + un/an + 1)$

onde a é um ponto viável conhecido

$a = (a1, a2 \ldots, an)$

De modo que um PPL na forma:

minimizar $c^t u$
sujeito a $Au = b$

pode ser escrito na seguinte forma:

minimizar $c^t x$
sujeito a $(A|b)x = 0$

onde já pode ser aplicado o algoritmo.

Seja n o número de variáveis do problema; uma variável Tolerância com valor 2^{-9}; c, a função objetivo; A, a matriz de restrições; CS, o ponto central do simplex (1/n, . . . , 1/n); V, o valor da função objetivo no centro do simplex; e x, o ponto que se está averiguando como ótimo. Então, o algoritmo de Karmarkar pode ser proposto da forma abaixo:

01. Faça x = CS
02. Faça D ser a matriz diagonal de x
03. Faça B = A-D
04. Adicione uma linha de 1's a B
05. Faça $CP = (I - B^t(BB^t)^{-1}B)\, Dc$
06. Faça $TCP = \sqrt{\sum_{i=1}^{n} (CPi)^2}$
07. Faça $R = \dfrac{1}{\sqrt{N(N-1)}}$
08. Faça $x1 = CS - \alpha R \dfrac{CP}{TCP}$

09. Faça $x = \frac{D - x1}{e^t - D - x1}$

10. Faça Vi ser o valor da função objetivo no ponto x

11. $\frac{V1}{V} >$ Tolerância?

Caso afirmativo ⇒ volte para 2

Caso negativo ⇒ pare, x é o ponto procurado.

O valor q é fornecido e vai determinar a precisão da resposta. O valor α também é dado, seu objetivo é garantir a eficiência do método e possui um valor comum de 0,25 (?).

Para o PPL abaixo, que já está na forma restrita, o algoritmo encontra a solução na 19ª iteração, com Tolerância igual a 0,001 e valores (0,003, 0,2501, 0,7497) quando a solução exata é (0, 0,25, 0,75).

minimzar 3x1 + 3x2 — x3
sujeito a 2x1 — 3x2 + x3 = 0

REFERÊNCIAS BIBLIOGRÁFICAS

01 — AGMON, S. (1954): **The Relaxation Method for Linear Inequalities.** Canadian Journal of Mathematics 6, pp. 382-392.

02 — BAZARAS, M. S.; Jarvis; J. J.: **Linear Programming and Network Flows,** J. Wiley, 1977.

03 — BERRESFORD, G. C.; Rockett, A. M.; Stevenson, J. C.: **Khachiyan's Algorithm,** Parti: A New Solution to Linear Programming Problems; Byte nº 8 (Agosto — 1980).

04 — BREGALDA, P. F.; Oliveira, A. A. F.; Bornstein, C. T.: **Introdução Programação Linear,** Ed. Campus, 1981.

05 — CHVÁTAL, V. (1983): **Linear Programming W. H. Freeman,** New York.

06 — DANTZIG, G. B. (1951): **Application of the Simplex Method to a Transportation Problem. In: Koopmans, T. C. (ed.),** Activity Analysis of Production and Allocation. Cowles Commission Monograph 13, J. Wiley & Sons, New York, pp. 359-373. (1953): **Computational Algorithm for the Revised Simplex Method,** Rand Report RM-1266. The Rand Corporation, Santa Monica, California. (1954): **Variables with Upper Bounds in Linear Programming, Rand Report RM-1271.** The Rand Corporation Santa Monica, California. (1963): **Linear Programming and Extension, Princeton University Press, Princeton, New Jersey. (1983): Reminiscences About the Origins of Linear Programming. In:** Bachem, A., Grotschel, M., Korte, B. (eds). Mathematical) Programming. The State of the Art, Bonn 1982. Springer, Berlin-Heidelberg-New Yor, pp. 78-86.

07 — EISELT, H. A.; Pederzoli, G.; Sandblom, C. — L. (1987): **Continuos Optimization Models;** Walter de Gruyter (Berlin-New York).

08 — GONZAGA, C. C.: **Algoritmos de Pontos Interiores para Programação Linear,** IMPA, 1987.

09 — HAREL, D.: **Algorithmics, The Spirit of Computing, Addison-Wesley,** 1987.

10 — KARMARKAR, N: **"A New Polynomial-Time Algorithm for Linear Programming";** Combinatoria, 4: 373-395, 1984.

11 — KHACHIYAN, L. G. (1979): **Polynomial Algorithm for Linear Programming.** Doklady Akademii Nauk SSSR 244/5, 1093-1096.

12 — MACULAN, N (1979): **Contribuições para a Solução de Problemas de Programação Linear;** Universidade Federal do Rio de Janeiro.

13 — MOTZKIN, T. S., Schonberg, I. J. (1954): **The Relaxation Method for Linear Inequalities.** Canadian Joumal of Mathematics 2,393-404.

14 — ROCKETT, A. M.; Steverson, J. C.: **"Karmarkar's Algorithm";** Byte, 50: 146-160, september 1987.

15 — SCHRIJVER, A. (1986): **Theory of Linear and Integer Programming, John Wiley, New York.**

16 — **WALSH, G. R. (1985):** An Introduction to Linear Programming; **John Wiley & Sons (Chichester-New York-Brisbane-Toronto-Singapore).**