MAC0315 – Programação Linear
Marcos Kazuya Yamazaki, NUSP: 7577622
Victor de Queiros Mattoso Archela dos Santos, NUSP: 7557202

Relatório EP2

Neste exercício programa foi implementado o algoritmo da fase 2 do simplex revisado onde a função $[ind\ v] = simplex(A,b,c,m,n,x)$, calcula o problema de programação linear para problemas onde as soluções viáveis do poliedro são não-degeneradas, dada a matriz A, o vetor b, o vetor da função objetivo c, o número de restrições m, o número de variáveis n e uma solução inicial básica x, que será o nosso ponto de partida.

Basicamente no método simplex neste EP, temos uma solução básica inicial com n variáveis, onde m elementos delas são não-nulas, $m <= n$, que pode ou não ser a solução ótima do sistema, antes de entrar na iteração verificamos quais são os índices básicos e não-básicos e guardamos os índices nos vetores IB e IN respectivamente.

Também achamos a matriz compostas pelas colunas básicas de A e guardamos ela na variável matriz B, $B = \begin{vmatrix} | & | & | & | \\ A(:,IB(1)) & A(:,IB(2)) & \dots & A(:,IB(m)) \\ | & | & | & | \end{vmatrix}$ que logo após calculamos a sua inversa e guardaremos nesta mesma variável.

Dentro do loop while, temos 5 passos principais, e que a partir daí foi implementado o algoritmo do simplex revisado, pois agora já temos os índices da solução básicas e suas respectivas colunas, e também temos a matriz inversa de base $B = B^{-1}$.

No primeiro passo, calculamos o vetor $p' = c'_{IB}B$ e os custos reduzidos $\overline{c_j} = c_j - p'A^j$ para todos os índices não-básicos j, que em octave estava nesta fórmula
$cj(i) = c(IN(i)) - p' * A(:,IN(i))$, já que IN contém os $n-m$ índices não básicos, o vetor cj terá $n-m$ custos reduzidos. Se todos os custos reduzidos forem maiores que zero, a solução x é ótima pois em todos os outros pontos a função objetiva não irá diminuir, e assim o algoritmo para devolvendo o vetor v que será a solução x, e a variável $ind = 0$, indicando que o algoritmo encontrou uma solução ótima.

Caso contrário, o algoritmo segue para o passo 2, escolhemos um índice $IN(i)$ tal que $(cj(i) < 0\ \&\&\ j > IN(i))$, ou seja, pegamos o menor índice em que o custo é negativo (para evitar ciclagem). O passo 2 consiste apenas em calcular as direções viáveis, $u = B * A(:,j) = -d_j$, se todos os elementos de u forem não-positivos, temos um problema ilimitado na direção da função objetivo e o algoritmo para devolvendo $ind = -1$.

Se o algoritmo não parou, no passo 3, temos elementos de $u_i > 0$, calcularemos $t = \frac{x(IB(i))}{u(i)}$ para todo $i\ tal\ que\ u(i) > 0$, e pegamos o $theta^* = \min(t)$, e k tal que $theta^* = \frac{x(IB(k))}{u(k)}$.

Agora, no próximo passo, vamos criar uma nova base, onde o índice $IB(k)$ sai da base e colocaremos o índice j. Também trocando os índices respectivos no vetor IN. Colocando

$IN(indice\ da\ variável\ que\ vai\ entrar\ na\ base) = IB(k)$. Também criaremos uma nova solução básica y que será dada por: $y(j) = theta^*$, $y\big(IB(i)\big) = x\big(IB(i)\big) - theta^*\, u(i)$ para $i\,! = k$ (coordenadas básicas) e $y(IB(k)) = 0$, (não-básicas). Para fazer isso criamos um vetor U que ao fazer a multiplicação $y = x + theta * U$, ela satisfaz as três restrições acima.

Agora que já temos todos os índices básicos e não-básicos atualizados e a nova solução básica, poderíamos fazer igual ao algoritmo ingênuo do método simplex, que pegará as colunas básicas formando uma nova matriz básica e calcular sua inversa, porém calcular a inversa de uma matriz não é nada eficiente pois ela tem um custo de $\sigma(m^3)$ operações aritméticas a cada interação do simplex. Para evitar esses cálculos que são computacionalmente custosas, antes do início de cada iteração temos que ter a matriz inversa conhecida, assim os cálculos de p' e u são basicamente produto de matriz-vetor custando $\sigma(m^2)$ operações aritméticas.

Para encontrar a nova matriz inversa, vamos aproveitar dos valores de u calculado nessa iteração e usaremos a matriz inversa de B, e para gerar a nova matriz de forma eficiênte, supomos uma matriz $[B \mid u]$ com m linhas e $m+1$ colunas, vamos transformar a última coluna dessa matriz num vetor canônico onde $u(k) = 1$, e para todos os outros elementos de u sejam zero, através de operações elementares de linhas. O algoritmo para isso é dividir a $k \cdot ésima$ linha dessa matriz por $u(k)$, e depois para todas as outras linhas l, $l \neq k$, subtrair a $k \cdot ésima$ linha multiplicada por $u(l)$. Num custo aritmético de no máximo $\sigma(m^2)$, pois antes de fazer a modificação em cada linha estou verificando se o valor de $u(l)$ ou $u(k)$, já não é o valor esperado.

Com isso, podemos partir para uma nova iteração.

Exemplo em que o PL tem solução ótima:

$$min(-10 \quad -12 \quad -12 \quad 0 \quad 0 \quad 0)x$$

$$s.a \begin{bmatrix} 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$$

$$x \geq 0$$

Usaremos como solução básica inicial $x = (0 \quad 0 \quad 0 \quad 20 \quad 20 \quad 20)$. A saída obtida foi:

<table>
<tr>
<td>
<pre>
Simplex: Fase 2

----------- Iterando 0 -----------
4 20.00000
5 20.00000
6 20.00000

Valor da função objetivo: 0.00000

Custos reduzidos
1 -10.00000
2 -12.00000
3 -12.00000

Entra na base: 1
</pre>
</td>
<td>
Imprimindo a solução básica associada à base x_4, x_5, x_6.

Como todos os custos reduzidos são negativos, o algoritmo irá continuar, pois essa solução não é ótima. O candidato a entrar na base é o menor índice, que nesse caso é o x_1.
</td>
</tr>
</table>

Direção 4 1.00000 5 2.00000 6 2.00000 Theta* 10.00000 Sai da base: 5 ----------- Iterando 1 ----------- 4 10.00000 1 10.00000 6 0.00000 Valor da função objetivo: -100.00000 Custos reduzidos 5 5.00000 2 -7.00000 3 -2.00000 Entra na base: 2 Direção 4 1.50000 1 0.50000 6 1.00000 Theta* 0.00000 Sai da base: 6 ----------- Iterando 2 ----------- 4 10.00000 1 10.00000 2 0.00000 Valor da funcao objetivo: -100.00000 Custos reduzidos 5 -2.00000 6 7.00000 3 -9.00000 Entra na base: 3 Direção 4 2.50000 1 1.50000 2 -1.00000 Theta* 4.00000 Sai da base: 4	Nesta parte, todas as direções u, foram positivas, então t foi calculado em todas as direções e assim escolhida o mínimo atingido com $k = 2$ ou $k = 3$, mas foi implementado que se tiver um k maior e os valores calculados são iguais, o menor k é escolhido. Tirando o x_5 da base. Novamente em x_2 e x_3 temos os custo negativos, vamos escolher o de menor índice para entrar na base. Como x_6 vale 0 e a direção associada a ela é positiva, temos $theta^* = 0$ que será o mínimo, já que $theta^* \geq 0$. Nesta parte, podemos observar que o custo reduzido associado a x_5 também é negativa, porém ela já esteve uma vez na base, um jeito barato computacionalemnte para a pivotação que evita a ciclagem, é pegar menor índice que satisfaz, neste caso o custo ser negativo. Assim pegamos x_3 para entrar na base.

<table>
<tr>
<td>

```
----------- Iterando  3 -----------
3 4.00000
1 4.00000
2 4.00000

Valor da função objetivo: -136.00000

Custos reduzidos
5 1.60000
6 1.60000
4 3.60000

Solução ótima encontrada com custo -136.00000
1 4.00000
2 4.00000
3 4.00000
4 0.00000
5 0.00000
6 0.00000

FIM da iteracao ---------------------------
```

</td>
<td>Achamos que todos os custos reduzidos sao positivos. Encontramos uma solução ótima.</td>
</tr>
</table>

Exemplo em que o PL ilimitado:

$$min(-1 \quad 0)x$$

$$s.a\ [1 \quad -1]x = [3]$$

$$x \geq 0$$

Usaremos como solução básica inicial $x = (3 \quad 0)$. A saída obtida foi:

<table>
<tr>
<td>

```
Simplex: Fase 2

----------- Iterando  0 -----------
1 3.00000

Valor da função objetivo: -3.00000

Custos reduzidos
2 -2.00000

Entra na base: 2

Direcao
1 -1.00000

O problema eh ilimitado na direção da função
objetivo

FIM da iteracao -----------------------------
```

</td>
<td>Aqui, x_2 é candidato a entrar na base, porém ao calcular a direção obtivemos que o valor também é negativo, ou seja, a função objetiva só descrece nessa direção, pois se $u \leq 0$, tem-se $theta^* = \infty$, logo o custo ótimo é $-\infty$ e o algoritmo para.</td>
</tr>
</table>