

Implementação do Método Preditor-Corretor

Atividade II - MS428-A

Raul Augusto Teixeira RA: 205177 Professor: Aurélio de Oliveira

30/11/2018

1 O Método Preditor Corretor de Pontos Interiores

Os métodos de Pontos Interiores para resolução de problemas de programação linear são uma das duas principais classes de métodos estudadas ao longo do semestre. Diferentemente do método simplex, o método de pontos interiores não atualiza a base para um outro vértice factível, mas parte de um ponto interior à região de factibilidade e atualiza para pontos interiores novos a cada iteração, se aproximando cada vez mais do ponto de otimalidade da função.

Comparados aos métodos Simplex, os métodos de Pontos Interiores normalmente convergem melhor quanto mais aumentam-se os problemas. O maior problema desse método é que, apesar de as iterações serem poucas, cada iteração é muito cara de se realizar. O método Preditor Corretor, particularmente, é o método de pontos interiores mais avançado atualmente conhecido pela comunidade acadêmica, e raramente passa de 200 iterações para haver a convergência ao ponto de otimalidade.

2 Implementação

A implementação do método Preditor Corretor, ao qual se refere este relatório, foi feita em MATLAB, e resolve problemas de programação linear assumindo que esses têm uma solução ótima. Os problemas devem estar na forma:

$$\begin{aligned} \min \quad & f(x) = c^t x \\ \text{s.a.} \quad & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{aligned}$$

2.1 Entradas

O programa recebe como entrada um arquivo contendo A , b e c , onde

- A é a matriz das restrições do problema;
- b é o vetor de restrições do problema;
- c é o vetor de custos do problema.

2.2 Saídas

O programa retorna:

- O valor de $\gamma = x^t z$ a cada iteração;
- O número de iterações realizadas pelo programa;
- Os valores finais da função $c^t x$ e $b^t y$.

3 Resultados

Foram disponibilizados dois problemas para teste do programa: o arquivo "scsd8" e o arquivo "scfxm3".

3.1 Arquivo "scsd8"

- Para $\mu = (\frac{\tilde{\gamma}}{\gamma})^2 * \frac{\tilde{\gamma}}{n}$, a função converge em 11 iterações para $f(x) = 95$;
- Para $\mu = (\frac{\gamma}{n})^2$, o programa converge para o mesmo valor, mas em 13 iterações;
- Para $\mu = (\frac{\gamma}{n^2})$ se $\gamma < 1$ e $\mu = (\frac{\tilde{\gamma}}{\gamma})^2 * \frac{\tilde{\gamma}}{n}$ caso contrário, a função também converge para o mesmo valor, mas em 12 iterações.

Podemos dizer que, nesse caso, o primeiro valor para μ é mais vantajoso, pois a convergência é mais rápida.

3.2 Arquivo "scfxm3"

- Para $\mu = (\frac{\tilde{\gamma}}{\gamma})^2 * \frac{\tilde{\gamma}}{n}$, o método não converge;
- Para $\mu = (\frac{\gamma}{n})^2$, o programa converge para $f(x) = 54901.25471$ em 29 iterações;
- Para $\mu = (\frac{\gamma}{n^2})$ se $\gamma < 1$ e $\mu = (\frac{\tilde{\gamma}}{\gamma})^2 * \frac{\tilde{\gamma}}{n}$ caso contrário, a função também converge para o mesmo valor, e na mesma quantidade de iterações.

Neste outro caso, em que $A_{990 \times 1800}$ é cerca de 50% maior que o anterior, onde temos $A_{397 \times 2750}$, o programa converge apenas para $\mu = (\frac{\gamma}{n})^2$. Podemos concluir, portanto, que possivelmente o primeiro valor de μ é melhor para problemas menores e médios, que o segundo é melhor para problemas maiores e que o terceiro garante a convergência de todos, mas nem sempre é tão bom quanto escolher um só valor fixo para μ .