A new method for solving linear programs

Mohand Bentobache[1,2], Mohand Ouamer Bibi[2]

[1] Laboratory of Pure and Applied Mathematics, LMPA, University of Laghouat, 03000, Algeria

mbentobache@yahoo.com

[2] LaMOS Research Unit, University of Bejaia, 06000, Algeria

mobibi.dz@gmail.com

Mots-clés : *linear programming, hybrid direction, suboptimality estimate.*

1 Introduction

In [1, 2], a new algorithm for solving linear programming problems with bounded variables was suggested. This algorithm uses the concept of hybrid direction in order to move from one feasible solution to a better one. In this work, we suggest a new hybrid direction algorithm for solving linear programs. In Section 2, we state the problem and give some definitions. In Section 3, we present the suggested method. Finally, Section 4 concludes the paper.

2 Problem statement and definitions

Consider the linear programming problem :

$$\max z = c^T x, \text{ subject to } Ax = b, \ l \leq x \leq u, \tag{1}$$

where c and x are n-vectors ; b an m-vector ; A an $(m \times n)$-matrix with $rankA = m < n$; l and u are finite-valued n-vectors. We define the following sets of indices : $I = \{1, 2, \ldots, m\}$, $J = \{1, 2, \ldots, n\}, \ J = J_B \cup J_N, \ J_B \cap J_N = \emptyset, \ |J_B| = m$. The set J_B is called a *support* if $\det(A_B) = detA(I, J_B) \neq 0$. An n-vector x is called a feasible solution (FS) if it satisfies the constraints of problem (1). A FS x^0 is called optimal if it maximizes the objective function $z(x) = c^T x$. A FS x^ϵ is called ϵ-optimal if $z(x^0) - z(x^\epsilon) \leq \epsilon$, where x^0 is an optimal solution for the problem (1) and ϵ is a nonnegative number.

3 An iteration of the method

Let $\{x, J_B\}$ be a support feasible solution (SFS) for the problem (1), ϵ a nonnegative number chosen in advance and $\eta > 0$. We compute the m-vector of multipliers π, the n-vector of reduced costs $\Delta^T = (\Delta_B^T, \Delta_N^T) = (0, \Delta_N^T)$ and the suboptimality estimate β as follows :

$$\pi^T = c_B^T A_B^{-1}, \ \Delta_N^T = \pi^T A_N - c_N^T, \text{ where } c_B = c(J_B), \ c_N = c(J_N), \ A_N = A(I, J_N), \tag{2}$$

$$\beta = \beta(x, J_B) = \sum_{j \in \Delta_j > 0, j \in J_N} \Delta_j (x_j - l_j) + \sum_{\Delta_j < 0, j \in J_N} \Delta_j (x_j - u_j). \tag{3}$$

If $\beta \leq \epsilon$, then the algorithm stops with the ϵ-optimal pair $\{x, J_B\}$. Else, we compute the following sets of indices :

$$J_{NE}^+ = \{j \in J_N : \Delta_j > \eta(x_j - l_j) \text{ and } x_j > l_j\}, J_{NE}^- = \{j \in J_N : \Delta_j < \eta(x_j - u_j) \text{ and } x_j < u_j\},$$

$$J_{NI}^+ = \{j \in J_N : 0 < \Delta_j \leq \eta(x_j - l_j)\}, \ J_{NI}^- = \{j \in J_N : \eta(x_j - u_j) \leq \Delta_j < 0\},$$

$$J_{NR}^+ = \{j \in J_N : \Delta_j > 0 \text{ and } x_j = l_j\}, \ J_{NR}^- = \{j \in J_N : \Delta_j < 0 \text{ and } x_j = u_j\},$$

$$J_{N0} = \{j \in J_N : \Delta_j = 0\}, \ J_{NI} = J_{NI}^+ \cup J_{NI}^-, \ J_{NE} = J_{NE}^+ \cup J_{NE}^-, \ J_{NR} = J_{N0} \cup J_{NR}^+ \cup J_{NR}^-.$$

Let us define the quantities γ and μ as follows :

$$\gamma = \sum_{j \in J_{NI}^+} \Delta_j (x_j - l_j) + \sum_{j \in J_{NI}^-} \Delta_j (x_j - u_j) + \frac{1}{\eta} \sum_{j \in J_{NE}^+ \cup J_{NE}^-} \Delta_j^2, \tag{4}$$

$$\mu = - \sum_{j \in J_{NE}^+} \Delta_j (x_j - l_j) - \sum_{j \in J_{NE}^-} \Delta_j (x_j - u_j) + \frac{1}{\eta} \sum_{j \in J_{NE}^+ \cup J_{NE}^-} \Delta_j^2. \tag{5}$$

We can prove that $\beta = \gamma - \mu \leq \gamma, \ \gamma \geq 0$ and $\mu \geq 0$. We define the direction d as follows :

$$\begin{aligned} &d_j = l_j - x_j, \ \text{if } j \in J_{NI}^+; \ d_j = u_j - x_j, \ \text{if } j \in J_{NI}^-; \\ &d_j = \tfrac{-\Delta_j}{\eta}, \ \text{if } j \in J_{NE}^- \cup J_{NE}^+; \ d_j = 0, \ \text{if } j \in J_{NR}; \ d_B = d(J_B) = -A_B^{-1} A_N d(J_N). \end{aligned} \tag{6}$$

Note that the direction d is feasible : $Ad = 0$. In order to improve the objective function while remaining in the feasible region, we compute the step length θ^0 along the direction d as follows :

$$\theta^0 = \min\{\theta_{j_1}, \theta_{j_2}, 1\}, \ \theta_{j_1} = \min\{\theta_j, j \in J_B\}, \ \theta_{j_2} = \min\{\theta_j, j \in J_{NE}\}, \tag{7}$$

where $\theta_j = (u_j - x_j)/d_j, \ \text{if } d_j > 0; \ \theta_j = (l_j - x_j)/d_j, \ \text{if } d_j < 0; \ \theta_j = \infty, \ \text{if } d_j = 0.$
Then the new FS is $\bar{x} = x + \theta^0 d$. We can prove that $z(\bar{x}) - z(x) = \theta^0 \gamma = \theta^0(\beta + \mu) \geq 0$ (d is an ascent direction) and $\bar{\beta} = \beta(\bar{x}, J_B) = (1 - \theta^0)\beta - \theta^0 \mu \leq \beta$ (the suboptimality decreases).
If $\theta^0 = 1$, then $J_{NE}^+ \cup J_{NE}^- = \emptyset \Rightarrow \mu = 0 \Rightarrow \bar{\beta} = 0$. So $\{\bar{x}, J_B\}$ is optimal.
If $\bar{\beta} \leq \epsilon$, then the algorithm stops with the ϵ-optimal pair $\{\bar{x}, J_B\}$.
If $\theta^0 = \theta_{j_2}$, then we start a new iteration with the pair $\{\bar{x}, J_B\}$. Else ($\theta^0 = \theta_{j_1} < 1$), we compute the n-vector $\kappa = x + d$ and the real number $\alpha_0 = \kappa_{j_1} - \bar{x}_{j_1}$, where j_1 is the index computed in (7). We compute the dual direction t :

$$t_{j_1} = -\text{sign}(\alpha_0); \ t_j = 0, \ j \neq j_1, \ j \in J_B; \ t_N^T = t_B^T A_B^{-1} A_N. \tag{8}$$

We compute the sets : $J_{N0}^+ = \{j \in J_{N0} : t_j > 0\}, \ J_{N0}^- = \{j \in J_{N0} : t_j < 0\}$, and the quantity :

$$\alpha = -|\alpha_0| + \sum_{j \in J_{N0}^+ \cup J_{NE}^+} t_j(\kappa_j - l_j) + \sum_{j \in J_{N0}^- \cup J_{NE}^-} t_j(\kappa_j - u_j). \tag{9}$$

We compute the new reduced costs vector and the new support as follows :

$$\bar{\Delta} = \Delta + \sigma^0 t \text{ and } \bar{J}_B = (J_B \setminus \{j_1\}) \cup \{j_0\}, \text{ where}$$

$$\sigma^0 = \sigma_{j_0} = \min_{j \in J_N}\{\sigma_j\}, \text{ with } \sigma_j = \begin{cases} \frac{-\Delta_j}{t_j}, & \text{if } \Delta_j t_j < 0\,; \\ 0, & \text{if } j \in J_{N0}^- \text{ and } \kappa_j \neq u_j\,; \\ 0, & \text{if } j \in J_{N0}^+ \text{ and } \kappa_j \neq l_j\,; \\ \infty, & \text{otherwise.} \end{cases} \tag{10}$$

We can prove that $\bar{\bar{\beta}} = \beta(\bar{x}, \bar{J}_B) = \beta(\bar{x}, J_B) + \sigma^0 \alpha$. If $\bar{\bar{\beta}} \leq \epsilon$, then the algorithm stops with the ϵ-optimal pair $\{\bar{x}, \bar{J}_B\}$. If $\alpha > 0$, we start a new iteration with the SFS $\{\bar{x}, J_B\}$. Else, we start a new iteration with the SFS $\{\bar{x}, \bar{J}_B\}$.

4 Conclusion

In this work, we have suggested a new hybrid direction method for solving linear programs with bounded variables. In futur work, we will compare it with the simplex algorithm [3] on randomly generated and practical test problems.

Références

[1] Mohand Ouamer Bibi and Mohand Bentobache. A hybrid direction algorithm for solving linear programs. *International Journal of Computer Mathematics*, 92(2) :200–216, 2015.

[2] Mohand Bentobache and Mohand Ouamer Bibi. *Numerical Methods of Linear and Quadratic Programming*. French Academic Presses, Germany, 2016 (in french).

[3] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.