Exponential approximation algorithm for linear programming problems

V.M. Gordunovsky

MGIMO (University) of the MFA of Russia, Moscow, Russia

In this talk we study a noniteration algorithm for calculating the optimal solution of the linear programming problem: \( \sum_{j=1}^{n} c_j \hat{x}_j \rightarrow \min \), \( \hat{x} \in R^n \), \( \sum_{j=1}^{n} \hat{a}_{ij} \hat{x}_j = \hat{b}_i, \ i \in I \), \( I = \{1, \ldots, m\} \), \( \hat{x}_j \geq 0, \ j \in J, \ j = \{1, \ldots, n\} \), where \( \hat{x}_j, \ j \in J \) - variables of the original problem, \( \hat{c}_j > 0, \ j \in J \) - coefficients of the linear form, \( \hat{a}_{ij} \geq 0, \ i \in I, \ j \in J \) - coefficients of the constraints, \( \hat{b}_i > 0, \ i \in I \) - right sides of the constraints. It is assumed that the problem has a solution. This problem we represent in the form: \( F = \sum_{j=1}^{n} x_j \rightarrow \min, \ x \in R^n \), \( \sum_{j=1}^{n} a_{ij} x_j = 1, \ i \in I, \ 0 \leq x_j \leq e^{-1}, \ j \in J \). To do this we transform the constants and variables in following sequence: 1) dividing both sides of the constraints by \( \hat{b}_i \), we get \( \sum_{j=1}^{n} \frac{\hat{a}_{ij}}{\hat{b}_i} \hat{x}_j = 1, \ i \in I \); 2) substituting \( \bar{x}_j = \frac{\hat{x}_j}{x_j^{max}} \), \( x_j^{max} = \max_i \frac{\hat{b}_i}{\hat{a}_{ij}} (\hat{a}_{ij} \neq 0), \ j \in J \), for the original variables \( \hat{x}_j \) we obtain \( \bar{a}_{ij} = e^{\hat{a}_{ij}/\hat{b}_i} x_j^{max}, \ \bar{c}_j = e^{\hat{c}_j} x_j^{max}, \ i \in I, \ j \in J \); 3) substituting \( c_j = \frac{\bar{c}_j}{\epsilon^{max}} \), \( \epsilon^{max} = \max_j \bar{c}_j \) for the coefficients of the objective function; 4) substituting \( x_j = \epsilon \bar{x}_j \), for the variables \( \bar{x}_j \), we get \( a_{ij} = \frac{\bar{a}_{ij}}{\epsilon} \), \( i \in I, \ j \in J \). The corresponding dual problem is \( \sum_{i=1}^{m} \lambda_i \rightarrow \max, \ \lambda \in R^m \), \( y_j = \sum_{i=1}^{m} a_{ij} \lambda_i - 1 \leq 0, \ j \in J \).

The linear programming problem is approximated by the exponential function \[ \varphi(\lambda, \epsilon) = \sum_{i=1}^{m} \tilde{\lambda}_i - \epsilon \sum_{j=1}^{n} \exp\left( \sum_{i=1}^{m} a_{ij} \tilde{\lambda}_i - 1 - \epsilon \right) \epsilon^{-1} \rightarrow \max, \ \tilde{\lambda} \in R^m, \ \\
\tilde{x}_j(\lambda, \epsilon) = \exp\left( \sum_{i=1}^{m} a_{ij} \tilde{\lambda}_i - 1 - \epsilon \right) \epsilon^{-1}, \ j \in J, \ \epsilon > 0 \] (1).

As \( \epsilon \rightarrow 0 \) the optimal value of the approximating function converges to the optimal values of the objective functions of the primal and dual linear problems: \( \sum_{j=1}^{n} \tilde{x}_j \rightarrow \sum_{j=1}^{n} x_j^*, \sum_{i=1}^{m} \tilde{\lambda}_i \rightarrow \sum_{i=1}^{m} \lambda_i^* \) and \( \tilde{x}_j \rightarrow x_j^*, \ \tilde{\lambda}_i \rightarrow \lambda_i^* \), where \( x_j^*, \ j \in J, \ \lambda_i^*, \ i \in I \) are the optimal solutions of the primal and dual linear problems [2]. A necessary and sufficient conditions for the maximum of strictly concave function \( \varphi(\tilde{\lambda}, \epsilon) \) is equality to zero derivatives \( \frac{\partial \varphi(\tilde{\lambda}, \epsilon)}{\partial \tilde{\lambda}_i} = \sum_{j=1}^{n} a_{ij} \tilde{x}_j(\tilde{\lambda}, \epsilon) - 1 = 0, \ i \in I \). We expand
the functions (1) as a power series in powers \((\sum_{i=1}^{m} a_{ij} \hat{\lambda}_i - 1 - \varepsilon), j \in J\) until the first derivative. Now the necessary and sufficient conditions are 
\[
\sum_{k=1}^{m} \hat{\lambda}_k \sum_{j=1}^{n} a_{ij} a_{kj} - \sum_{j=1}^{n} a_{ij} - \varepsilon = 0, i \in I.
\]
As \(\varepsilon \to 0\) we obtain an approximating system of linear equations to calculate the approximate dual variables: \(\sum_{k=1}^{m} \hat{\lambda}_k \sum_{j=1}^{n} a_{ij} a_{kj} = \sum_{j=1}^{n} a_{ij}, i \in I (2).\)

The variables \(\hat{y}_j = \sum_{i=1}^{m} a_{ij} \hat{\lambda}_i - 1\) in (1) is an approximation of the dual constraints \(y_j \leq 0\). If the constraint is active, then \(y_j\) attains its maximum value, and the corresponding \(x_j\) is the optimal solution \(x_j^*\). Hence, \(\max \hat{y}_j\) specifies the index \(j\) of the optimal \(x_j^*\). Further, we exclude corresponding unknown \(x_j\) from the objective function and the constraints of the linear problem. We obtain a problem of smaller dimension to calculate the next index of the optimal \(x_j^*\).

A numerical example illustrates the application of this algorithm: \(\sum_{j=1}^{5} \hat{c}_j \hat{x}_j \to \min, \sum_{j=1}^{5} \hat{a}_{ij} \hat{x}_j = \hat{b}_i, i = 1, 2, 3, j = 1, \ldots, 5, (\hat{c}_j) = (1.20724), (\hat{b}_i) = (1.42), (\hat{a}_{ij}) = \begin{pmatrix} 1.0, 0.4, 0.3, 2.0, 3.0 \\ 0.3, 4.0, 0.3, 3.0, 0.5 \\ 1, 0.1, 3.0, 1.0, 0.2 \end{pmatrix}.\)

The solution of this original linear problem is \(\sum_{j=1}^{5} \hat{c}_j \hat{x}_j^* = 2.4367,\)
\[(\hat{x}_j^*) = (0.0, 0.7597, 0.5530, 0.2651, 0.0).\] After the transformation we get \(F = \sum_{j=1}^{5} x_j \to \min, \sum_{j=1}^{5} a_{ij} x_j = 1, i = 1, 2, 3, 0 \leq x_j \leq e^{-1}, j = 1, \ldots, 5, (a_{ij}) = \begin{pmatrix} 108,7313 & 21,7463 & 46,5991 & 108,7313 & 81,5485 \\ 8,1549 & 54,3656 & 11,6498 & 40,7742 & 3,3979 \\ 54,3656 & 2,7183 & 232,9956 & 27,1828 & 2,7183 \end{pmatrix}.\)

The solution is \(\sum_{j=1}^{5} x_j^* = 0.0224, (x_j^*) = (0.0, 0.0140, 0.0036, 0.0049, 0.0), (\lambda_i^*) = (0.0019, 0.0175, 0.0030), (y_j^*) = (-0.4879, 0.0, 0.0, 0.0, -0.7792).\) The approximating system (2) is \(\begin{pmatrix} 32939,5073 & 7322,3284 & 20005,0400 \\ 7322,3284 & 4831,9243 & 4423,0664 \\ 20005,0400 & 4423,0664 & 57996,2489 \end{pmatrix} \begin{pmatrix} \hat{\lambda}_1 \\ \hat{\lambda}_2 \\ \hat{\lambda}_3 \end{pmatrix} = \begin{pmatrix} 367,3564 \\ 118,3423 \\ 319,9806 \end{pmatrix}.\) Now we compute the approximate value of the dual variables \((\hat{\lambda}_i) = (0.0073, 0.0115, 0.0021)\) and the constraints: \((\hat{y}_j) = (0.0043, -0.2119, -0.0318, 0.3206, -0.3586).\) The greatest \(\hat{y}_4 = 0.3206\) specifies index \(j = 4\) of the optimal solution. Then we exclude variable \(x_4\) from the objective function \(F\) and the constraints \(\sum_{j=1}^{5} a_{ij} x_j = 1, i = 1, 2, 3.\) We transformate problem to the form: \(\sum_{j=1}^{3} a_{ij} x_j = 1, i = 1, 2, 0 \leq x_j \leq e^{-1}, j = 1, 2, 3, 5, (a_{ij}) = \begin{pmatrix} 3.8423 & 233.4043 & 115.6802 & 67.9846 \\ 209,1373 & 119,6861 & 420,1277 & 94,1167 \end{pmatrix}.\) Recalculating the coefficients
of the approximating system (2), we get \(( \begin{pmatrix} 72496,1231 & 83737,7412 \\ 83737,7412 & 243428,4278 \end{pmatrix} ) \begin{pmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \end{pmatrix} = \begin{pmatrix} 420,9113 \\ 843,0678 \end{pmatrix} \). The solution is \(( \tilde{\lambda}_i ) = ( 0.0030 \ 0.0024 ), ( \tilde{y}_j ) = ( -0.4797 \ -0.0095 \ 0.3686 \ -0.5674 \). The greatest \( \tilde{y}_3 = 0.3686 \) specifies the index \( j = 3 \) of the optimal solution \( x_3^* \). We exclude variable \( x_3 \) from the objective function and the constraints. Then we transformate problem to the form: \( \sum_{j=1,2,5} x_j \rightarrow \min, \sum_{j=1,2,5} a_{ij} x_j = 1, i = 1, 0 \leq x_j \leq e^{-1}, j = 1, 2, 5, (a_{ij}) = ( 0.4071 \ 415,1550 \ 173,0553 ) \). Recalculating the coefficients of the approximating system (2), we get \(( \begin{pmatrix} 202301,9342 \\ 54,324 \end{pmatrix} \begin{pmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \end{pmatrix} = \begin{pmatrix} 588,6174 \end{pmatrix} \). The solution of the approximating system (2) is \(( \tilde{\lambda}_1 ) = ( 0.0029 ), ( \tilde{y}_j ) = ( -0.9988 \ 0.2079 \ -0.4965 \). The greatest \( \tilde{y}_2 = 0.2079 \) specifies index \( j = 2 \) of the optimal \( x_2^* \). Thus we have found basis variables \( x_2, x_3, x_4 \) of the optimal solution. In the original set of the constraints we leave only 2, 3 and 4 columns. Resulting system of linear equations determines the optimal values of the basis variables. Finally, solution of the original linear programming problem is \( \sum_{j=1}^{5} \hat{c}_j \hat{x}_j^* = 2.4367, \begin{pmatrix} \hat{x}_j^* \end{pmatrix} = ( 0.0 \ 0.7597 \ 0.5530 \ 0.2651 \ 0.0 ) \).

References
