Selection of Constraints with a New Approach
in Linear Programming Problems

1P.Sumathi and 2A.Gangadharan

1Research Scholar, Department of Mathematics
Eswari Engineering College, Ramapuram, Chennai - 600 089
Tamil Nadu, India

2Professor & Research Supervisor, Department of Mathematics
Eswari Engineering College, Ramapuram, Chennai - 600 089
Tamil Nadu, India

Copyright © 2014 P. Sumathi and A. Gangadharan. This is an open access article distributed under
the Creative Commons Attribution License, which permits unrestricted use, distribution, and
reproduction in any medium, provided the original work is properly cited.

Abstract

Linear programs (LP) play an important role in the theory and practice of
optimization problems. Linear programming deals with problems such as
maximising profits, minimising costs or ensuring you make the best use of
available resources. While formulating a linear programming model, system
analyst and researchers often tend to include all the possible constraints, although
some of them may not be binding at the optimal solution. It is well known that, for
most of the large scale LP problems, only a relatively small percentage of
constraints are binding at the optimal solutions. Researchers have proposed
methods which identify those constraints most likely to be tight at optimality. This
paper proposes a new approach for finding solutions to LP problems by using a
part of the constraints with the help of intercept and projection values of the each
constraint. This method is more efficient when compared with the existing
method. The developed algorithm is implemented by programming language Java
and the computational results are presented. It shows that the proposed method
reflects a significant decrease in the computational effort and is one of the best
alternatives to select the necessary constraints prior to solving an LP problem.
Keywords: Intercept matrix, projection value, Constraint selection, linear programming problem (LPP), Cosine criterion, simplex method

1. Introduction

Linear programming is a tool for solving optimization problems. Many researchers have proposed different algorithms to solve linear programming problems. Solving linear programming problems efficiently has always been a fascinating pursuit for computer scientists and mathematicians. The computational complexity of any linear programming problem depends on the number of constraints and variables of the LP problem. Preprossing is an important technique in the practice of linear programming problem. The reduction of large scale problem can save significant amount of computational effort during the solution of a problem. Many researchers have proposed algorithms for selecting necessary constraints in LP models.

In solving an LP Problem (LPP), it is acknowledged that redundancies do exist in most of the practical LPPs. The importance of detecting and removing redundancy in a set of linear constraints is the avoidance of all the calculations associated with those constraints when solving an associated LPP.

Many researchers [1 - 15] have proposed different methods for solving and identifying redundant constraints in LPP.

LPP represents a mathematical model for solving numerous practical problems and it consists of objective function and a set of constraints.

The general form of LPP is

Optimize \( Z = C^T X \)

Subject to the constraints,

\( AX \leq b \)

\( X \geq 0 \)

Where \( X \) represents the vector of variables (to be determined) while \( C \) and \( b \) are vectors of (known) coefficients and \( A=[a_{ij}] \) is a (known) matrix of coefficients. If the technological coefficients \( a_{ij} \)'s \( \geq 0 \) and \( b_i > 0 \), the LPP is said to be “Non-negative linear programming problem (NNLPP)”.

In solving an LPP, we tend to include all possible constraints that will increase the number of iterations and computational effort. It is well known that for most of the large scale LP problems, only a relatively small percentage of constraints are binding at the optimal solutions. The purpose of this paper is to propose a new heuristic approach which obtains solutions to the LP problems with the use of some of the constraints.

This paper proceeds as follows: Section 2 gives the definitions of some of the terms and some of the earlier methods for selecting the constraints to solve the
LPP. Section 3 presents a proposed approach to identify the constraints from a set of constraints and to solve the problem with some numerical examples. Section 4 gives the computational results. Section 5 concludes the paper.

2. Some Preceding Methods

In this section, the definitions of the terms such as binding and redundant constraints and some of the earlier methods developed for selecting the constraints are explained.

2.1 Definitions

**Binding Constraints**

These are the constraints which define the feasible region and they are involved in finding the optimal solution. They are also called active/operative constraints.

**Redundant Constraints**

A redundant constraint is a constraint that can be removed from a system of linear constraints without changing the feasible region.

Consider the following system of m non-negative linear inequality constraints and n variables (m > n).

\[ AX \leq b, \quad X \geq 0 \]  \hspace{1cm} (1)

where \( A \in \mathbb{R}^{m \times n} \), \( b \in \mathbb{R}^m \), \( X \in \mathbb{R}^n \) and \( 0 \in \mathbb{R}^n \).

Let \( A_i X \leq b_i \) be the \( i^{th} \) constraint of the system (1) and let \( S = \{ X : R^n / A_i X \leq b_i, X \geq 0 \} \) be the feasible region associated with system (1).

Let \( S_k = \{ X : R^n / A_k X \leq b_k, X \geq 0, i \neq k \} \) be the feasible region associated with the system of equations \( A_i X \leq b_i \), \( i = 1, 2, m, i \neq k \). The \( k^{th} \) constraint \( A_k X \leq b_k \) (1 ≤ k ≤ m) is redundant for the system (1) if and only if \( S = S_k \).

2.2 Largest Summation Rule

In this approach, each constraint is divided by the corresponding right-hand side value and denoted by \( \sum_{i=1}^{m} \frac{A_i}{b_i} \). Compute \( S_i = \sum_{j=1}^{n} \frac{a_{ij}}{b_{ij}} \) and optimize with the constraints corresponding to the first two maximum values of \( S_i \). If it is not optimal, append the violated constraints with the existing problem.
2.3 Cosine Simplex Algorithm

For a given problem it begins by solving a relaxed problem consisting of the original objective function subject to a constraint corresponding to the index

\[ r = \arg \max_i \cos \theta_i, (i = 1, 2, 3, \ldots, m) \]

Where \( \cos \theta_i = \frac{a_i^T c}{|a_i||c|} \), yielding a non-empty bounded feasible region. At each subsequent iteration of the algorithm, the most parallel constraint to the objective function among those constraints violated by the solution to the current relaxed problem is appended to it. When no constraints are violated, the solution of the current relaxed problem is optimal to the original problem.

2.4 Constraints Optimal Selection Technique

They have taken NNLPP and these two factors are taken into consideration (i) Angle of its normal column vector \( a_i^T \) with \( c \) of the objective function. (ii) The depth of the cut that \( a_i x \leq b_i \) removes as a violated inoperative constraint from the feasible region.

Define \( \text{RAD} (a_i, b_i, c) = \frac{|a_i| \cos(a_i, c)}{b_i} \)

Take the constraint corresponding to \( \max RAD (a_i, b_i, c) \) and set the initial relaxed problem. On solving we get a solution. Check for inoperative constraints in the decreasing order. Take the first one violated by the solution and append the constraint to relaxed problem. Continue this till no inoperative constraint exists. The resultant solution is the optimal solution.

2.5 Limitations

The cosine criterion [3] plays a vital role more in the small scale problem than in the simplex but not in the large scale problem. The number of constraints selected is more in the large scale problem. The COSTs is more efficient than cosine simplex but when tie appears it undergoes cycling. Moreover, COSTs is for NNLPP whereas this new proposed method withstands all these limitations.

3. Proposed Method

This Proposed method identifies the relevant set of constraints which finds out the optimal solution for the following LPP.

Maximize \( Z = C^T X \)

Subject to the constraints

\( AX \leq b \)

\( X \geq 0, A \in \mathbb{R}^{m \times n}, b > 0, X, C \in \mathbb{R}^n \).
3.1 Algorithm

The steps of the algorithm are as follow:

1. Construct a matrix of intercepts of all the decision variables formed by each of the resource constraints along the respective coordinate axis
   \[ j_i = \frac{b_i}{a_{ij}}, \quad a_{ij} > 0, \quad i = 1,2,...m \text{ and } j = 1,2,...,n. \]
   Find the \( \min \{ j_i \} = \beta_j \) in each row \( j \).

2. Set \( L = \{i : \min \{ j_i \} < \max \{ j_i \}, \text{for all } j\} \), \( L \) is the index of the selected set of constraints and assumes it contains \( p \) number of constraints, \( p \leq m \).

3. Set \( J = \{m + j, j = 1,2,3,...,n\} \)

4. Find \( \alpha_K = \frac{a_{KC}}{||a_{KC}||}, K \in L \cup J \).

5. Set \( q = 1 \).
   Solve the relaxed problem
   \[
   \text{Maximize } Z = C^T X \\
   \text{Subject to the constraints} \\
   a_r x \leq b_r, \\
   a_s x \leq b_s, \\
   x \geq 0,
   \]
   where the \( r^{th} \) and \( s^{th} \) constraints are selected by \( r = \arg(\max_{K} \alpha_K) \) and \( s = \arg(\min_{K} \alpha_K) \), provided \( \alpha_K > 0 \). Let the solution obtained, be \( x(q) \).
   Check for violated constraints. If none is found, stop. Since \( x(q) \) is the optimal solution. Otherwise

6. Take \( l = \arg(\max \alpha_k) \) of the violated constraints if \( q \) is odd or Take \( l = \arg(\min \alpha_k) \) of the violated constraints if \( q \) is even. Set \( q = q + 1 \). Append the violated constraint corresponding to index \( l \) to the relaxed problem and go to step 5.

3.2 Illustration

The proposed method is illustrated with the following numerical examples.

**Example 1:**
Max \( Z = 20x_1 + 10x_2 + x_3 \)
Subject to the constraints
\[
3x_1 - 3x_2 + 5x_3 \leq 50 \\
1x_1 + 0x_2 + 1x_3 \leq 10 \\
1x_1 - 1x_2 + 4x_3 \leq 20 \\
2x_1 + 1x_2 + 3x_3 \leq 30
\]
\[2x_1 + 2x_2 + x_3 \leq 66\]
\[x_1 + 4x_2 + 3x_3 \leq 75\]
\[2x_1 + 2x_2 + 3x_3 \leq 80\]
\[x_1, x_2, x_3 \geq 0\]

**Solution:**

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
<th>(S_4)</th>
<th>(S_5)</th>
<th>(S_6)</th>
<th>(S_7)</th>
<th>(\beta_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>16.67</td>
<td>10</td>
<td>20</td>
<td>15</td>
<td>33</td>
<td>75</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>(x_2)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>30</td>
<td>33</td>
<td>18.75</td>
<td>40</td>
<td>18.75</td>
</tr>
<tr>
<td>(x_3)</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>66</td>
<td>25</td>
<td>26.67</td>
<td>5</td>
</tr>
</tbody>
</table>

Here the set \(L = \{1, 2, 3, 4, 6\}\) are the selected constraints.

\[\alpha_1 = 5.337 \quad \alpha_2 = 14.849 \quad \alpha_3 = 3.3 \quad \alpha_4 = 14.165 \quad \alpha_6 = 12.355\]

\[\alpha_8 = -20 \quad \alpha_9 = -10 \quad \alpha_{10} = -1\]

Here we select \(\alpha_i > 0\) only.

- **q=1**
  - Optimise the objective function subject to 2\(^{nd}\) and 3\(^{rd}\) constraints; here the solution is not optimal.

- **q=2**
  - Select 2\(^{nd}\), 3\(^{rd}\) and 4\(^{th}\) constraints, on solving we obtain the optimal solution to be \(x_1 = 10, x_2 = 10, x_3 = 0; Z = 300;\)

**Example 2:**

Max \(Z = 5x_1 + 6x_2 + 3x_3\)

Subject to the constraints
\[5x_1 + 5x_2 + 3x_3 \leq 50\]
\[2x_1 + 2x_2 + 0x_3 \leq 40\]
\[7x_1 + 6x_2 - 9x_3 \leq 30\]
\[5x_1 + 5x_2 + 5x_3 \leq 35\]
\[12x_1 + 6x_2 - 1x_3 \leq 90\]
\[-1x_1 + 1x_2 - 2x_3 \leq 20\]
\[x_1, x_2, x_3 \geq 0\]

**Solution:**

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
<th>(S_4)</th>
<th>(S_5)</th>
<th>(S_6)</th>
<th>(\beta_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>10</td>
<td>20</td>
<td>4.29</td>
<td>7</td>
<td>7.5</td>
<td>-</td>
<td>4.29</td>
</tr>
<tr>
<td>(x_2)</td>
<td>10</td>
<td>20</td>
<td>5</td>
<td>7</td>
<td>15</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>(x_3)</td>
<td>16.67</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>7</td>
</tr>
</tbody>
</table>
Here the set $L=\{3,4\}$ are the selected constraints.

$$\alpha_3= 3.415 \quad \alpha_4 = 8.083 \quad \alpha_5 = -0.4082 \quad \alpha_4 = -3 \quad \alpha_5 = -2 \quad \alpha_6 = -5$$

Here we select $\alpha_i > 0$ only.

$q=1$

Optimise the objective function subject to $3^{rd}$ and $4^{th}$ constraints. We obtain the optimal solution to be $x_1=0, \ x_2=6.2, \ x_3=0.8; Z=3.96;$

**Example 3:**

Max $Z = x_1 + x_2 + x_3$

Subject to the constraints

- $12x_1 + 12x_2 + 13x_3 \leq 312$
- $-11x_1 + 14x_2 + 15x_3 \leq 1155$
- $8x_1 + 7x_2 + 20x_3 \leq 200$
- $-4x_1 + 3x_2 + 6x_3 \leq 900$
- $2x_1 + 3x_2 + 7x_3 \leq 70$
- $9x_1 + 15x_2 + 2x_3 \leq 120$
- $-1x_1 + 3x_2 + 8x_3 \leq 70$
- $5x_1 - 2x_2 + 7x_3 \leq 75$
- $4x_1 + 6x_2 - 3x_3 \leq 50$
- $x_1, x_2, x_3 \geq 0$

**Solution:**

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
<th>$S_8$</th>
<th>$S_9$</th>
<th>$\beta_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>26</td>
<td>-</td>
<td>25</td>
<td>--</td>
<td>35</td>
<td>13.3</td>
<td>--</td>
<td>15</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>$x_2$</td>
<td>26</td>
<td>82.5</td>
<td>28.6</td>
<td>300</td>
<td>23.3</td>
<td>8</td>
<td>23.3</td>
<td>--</td>
<td>8.3</td>
<td>8</td>
</tr>
<tr>
<td>$x_3$</td>
<td>24</td>
<td>77</td>
<td>10</td>
<td>150</td>
<td>10</td>
<td>60</td>
<td>8.75</td>
<td>10.7</td>
<td>--</td>
<td>8.75</td>
</tr>
</tbody>
</table>

Here the set $L=\{3,5,6,7,8,9\}$ are the selected constraints.

$$\alpha_3= 1.545 \quad \alpha_5= 1.524 \quad \alpha_6=1.477 \quad \alpha_7=1.162 \quad \alpha_8=1.132 \quad \alpha_9=0.896$$

Here we select $\alpha_i > 0$ only.

$q=1$

Optimise the objective function subject to $3^{rd}$ and $9^{th}$ constraints, solution is $x_1=15.38, \ x_2=0, \ x_3=3.85; Z=19.23;$ But it is not optimal. Here $6^{th}$ & $8^{th}$ Constraints are In-operative.
\[ q = 2 \]
Select 3\(^{rd}\), 9\(^{th}\) and 6\(^{th}\) constraints, on solving solution is \(x_1 = 12.2, x_2 = 0, x_3 = 5.12; Z = 17.32\); But it is not optimal. Here 8\(^{th}\) Constraint alone is In-operative.

\[ q = 3 \]
Select 3\(^{rd}\), 9\(^{th}\), 6\(^{th}\), 8\(^{th}\) constraints, on solving we obtain the optimal solution to be \(x_1 = 7.6, x_2 = 2.63, x_3 = 6.04; Z = 16.27\);

4. Computational Results

The efficiency of the algorithm is tested by solving LPP for small scale problems using the Cosine Simplex method [3] and the proposed method. The following table 1 shows a comparison between the numbers of constraints selected for two methods and it shows that the proposed method is very useful to solve the LPP with a minimum amount of computational efforts and time. The comparative results of the two methods are clearly shown in the respective graph followed by the table 1. Table 2 and 3 shows the number of constraints selected and time in seconds for the proposed method only. The Cosine simplex method is applicable for small-scale problems only.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Size of the problem No. of Constraints (m)</th>
<th>No. of Selected Constraints, No. of Variables (n)</th>
<th>No. of Selected Constraints,</th>
<th>No. of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Simplex Method</td>
<td>Cosine Simplex Method</td>
<td>Proposed Method</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 2. Comparison of the Computational Efforts in Solving LPP: Large – Scale Problems

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>File name</th>
<th>No. of Constraints (m)</th>
<th>No. of Variables (n)</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No. of Selected Constrains</td>
</tr>
<tr>
<td>1</td>
<td>Scpe1</td>
<td>50</td>
<td>500</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>Scpe2</td>
<td>50</td>
<td>500</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>Scpe3</td>
<td>50</td>
<td>500</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>Scpe4</td>
<td>50</td>
<td>500</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>Scpe5</td>
<td>50</td>
<td>500</td>
<td>40</td>
</tr>
</tbody>
</table>
Table 3. Comparison of the Computational Efforts in Solving LPP: Netlib Problems

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>File name</th>
<th>No. of Constraints (m)</th>
<th>No. of Variables (n)</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No. of Selected Constrains</td>
</tr>
<tr>
<td>1</td>
<td>afiro</td>
<td>20</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>fit1d</td>
<td>24</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>fit2d</td>
<td>25</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>sc50b</td>
<td>28</td>
<td>28</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>sc50a</td>
<td>29</td>
<td>29</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>kb2</td>
<td>39</td>
<td>39</td>
<td>38</td>
</tr>
<tr>
<td>7</td>
<td>vtpbase</td>
<td>51</td>
<td>51</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>bore3d</td>
<td>52</td>
<td>52</td>
<td>51</td>
</tr>
</tbody>
</table>

5. Conclusion

A new algorithm using an intercept and pivot index values has been proposed for the selection of constraints to solve the linear programming problems. The resulting model is solved in order to establish the validity of the proposed method. A significant reduction in the computational effort is achieved.
REFERENCES

2. J. C. G. Boot, On trivial and binding constraints in programming problems, Management Science, 8 (1962), 419-441.

Received: February 1, 2014