

## *Research Article*

# **A Comparative Study of Redundant Constraints Identification Methods in Linear Programming Problems**

**Paulraj S.<sup>1</sup> and Sumathi P.<sup>2</sup>**

<sup>1</sup> *Department of Mathematics, Madras Institute of Technology Campus, Anna University, Chromepet, Chennai, Tamil Nadu 600 044, India*

<sup>2</sup> *Anna University, Chennai, Tamil Nadu, India*

Correspondence should be addressed to Paulraj S., paulraj@annauniv.edu

Received 7 January 2010; Revised 15 April 2010; Accepted 21 September 2010

Academic Editor: Joaquim J. Júdice

Copyright © 2010 Paulraj S. and Sumathi P. 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.

The objective function and the constraints can be formulated as linear functions of independent variables in most of the real-world optimization problems. Linear Programming (LP) is the process of optimizing a linear function subject to a finite number of linear equality and inequality constraints. 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. Quite often large-scale LP problems may contain many constraints which are redundant or cause infeasibility on account of inefficient formulation or some errors in data input. The presence of redundant constraints does not alter the optimal solutions(s). Nevertheless, they may consume extra computational effort. Many researchers have proposed different approaches for identifying the redundant constraints in linear programming problems. This paper compares five of such methods and discusses the efficiency of each method by solving various size LP problems and netlib problems. The algorithms of each method are coded by using a computer programming language C. The computational results are presented and analyzed in this paper.

## **1. Introduction**

Many researchers [1–17] have proposed different algorithms to identify the redundancies and removed them to get a reduced model for linear programming. In 1965, Zionts [17] suggested some improvements upon the implementation of Boot method, but not to the point where it achieved practical value. In addition, a number of other methods were developed that deal with redundancy, among which the geometric vertex enumeration method is the most well known. In geometric vertex enumeration method, the essential characteristic

is the establishment of a number of situations in which redundancy can be recognized immediately without further computations.

In 1971, Lisy [12] used the rules given by Ziont to identify all redundant constraints in systems of linear constraints. Gal [7] enlarged this approach by adding rules for situations in which constraints can be identified immediately as being nonredundant. Gal proposed another method to classify constraints as redundant or necessary. They produce results that are unconditionally correct; they perform iterations of an active set linear programming algorithm. Later Caron et al. [6] appended the above methods by adding rules to deal with degeneracy.

Brearily et al. [4] proposed a simple method to identify redundant constraints from a system of linear constraints. This method involves the lower and upper bounds of the variables. Telgan [15] proposed a deterministic method to identify redundant constraints by using minimum ratio criteria as in simplex method. Stojković and Stanimirović [13] proposed a method to identify redundant constraints by applying the maximum and minimum principle. Paulraj et al. [14] proposed a heuristic method to identify redundant constraints by using the intercept matrix of constraints of a linear programming problem. Gutman and Ioslovich [8] described a new approach to preprocess nonnegative large-scale problems so as to reduce the dimensions considerably by defining and removing redundant constraints and variables. This test is applicable to all nonnegative large-scale linear programming problem with group constraints. Group constraints only contain zeros and ones coefficients. Constraints and variables are removed by primal and dual tests. This method is applicable to constraints of knapsack problems.

A brief introduction to the redundant constraints of linear programming problems is presented in Section 2. Section 3 discusses the methods for identifying redundant constraints in linear problems. Section 4 deals with the computational results of the methods, and Section 5 concludes the paper.

## 2. 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  $m$  nonnegative linear inequality constraints and  $n$  variables ( $m \geq n$ ):

$$AX \leq b, \quad X \geq 0, \quad (2.1)$$

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

Let  $A_i X \leq b_i$  be the  $i$ th constraint of the system (2.1) and let  $S = \{X \in R^n / A_i X \leq b_i, X \geq 0\}$  be the feasible region associated with system (2.1).

Let  $S_k = \{X \in R^n / A_i X \leq b_i, X \geq 0, i \neq k\}$  be the feasible region associated with the system of equations  $A_i X \leq b_i, i = 1, 2, \dots, m, i \neq k$ . The  $k$ th constraint  $A_k X \leq b_k (1 \leq k \leq m)$  is redundant for the system (2.1) if and only if  $S = S_k$ .

*Definition 2.1.* Redundant constraints can be classified as weakly and strongly redundant constraints.

*Weakly Redundant Constraints*

The constraint  $A_i X \leq b_i$  is weakly redundant if it is redundant and  $A_i X = b_i$  for some  $X \in S$ .

*Strongly Redundant Constraints*

The constraint  $A_i X \leq b_i$  is strongly redundant if it is redundant and  $A_i X < b_i$  for all  $X \in S$ .

*Binding Constraint*

Binding constraint is the one which passes through the optimal solution point. It is also called a relevant constraint.

*Nonbinding Constraint*

Nonbinding constraint is the one which does not pass through the optimal solution point. But it can determine the boundary of the feasible region.

*Example 2.2.* Consider the following linear inequality constraints:

- (1)  $2x_1 + 1x_2 \leq 8$ ,
- (2)  $4x_1 + 0x_2 \leq 15$ ,
- (3)  $1x_1 + 3x_2 \leq 9$ ,
- (4)  $1x_1 + 2x_2 \leq 14$ ,
- (5)  $1x_2 \leq 4$ ,
- (6)  $1x_1 + 1x_2 \leq 5$ ,

where  $x_1 + x_2 \leq 0$ .

In Figure 1, the region OABCD is the feasible region and the vertex C is the optimal point. The constraints (1), (3), and (6) are binding, (4) and (5) are strictly redundant. The 2nd constraint is non-binding. Among the binding constraints, (6) is weakly redundant.

### 3. Methods for Identification of Redundant Constraints

Many methods are available in the literature to identify the redundant constraints in linear programming problems. In this paper, the following five methods are discussed and compared

- (1) bounds method [4]
- (2) linear programming method [6]
- (3) deterministic method [15]
- (4) stojković and Stanimirović method [13]
- (5) heuristic method [14].

#### 3.1. Bounds Method

Brearly et al. [4] proposed a simple method for identifying redundant constraints for a Linear Programming Problem (LPP) with bounded variables. This method involves the lower and upper bounds of the variables. The upper and lower bounds of each constraint are computed and compared with the right-hand side of that constraint to decide if it is a redundant constraint or not.

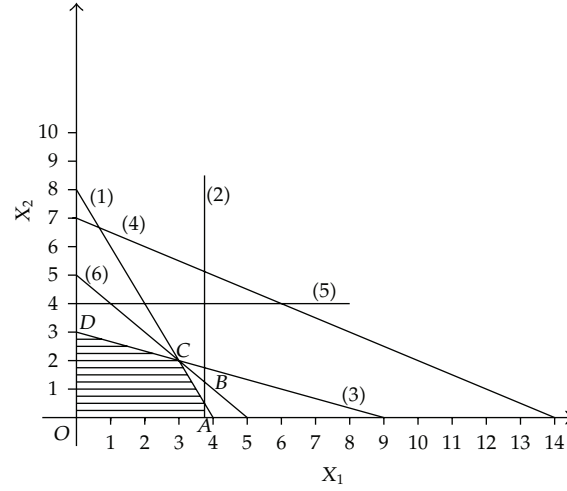


Figure 1

*Procedure of the Method.*

The general form of an LPP with bounded variables is

$$\begin{aligned} \max \quad & z = \sum_{j=1}^n c_j x_j, \\ \text{Subject to} \quad & \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = 1, 2, \dots, m), \\ & l_j \leq x_j \leq u_j \quad (j = 1, 2, \dots, n). \end{aligned} \quad (3.1)$$

*Step 1.* Compute upper and lower bounds for each constraint by

$$\begin{aligned} U_i &= \sum_{j \in P_i} a_{ij} u_j + \sum_{j \in N_i} a_{ij} l_j, \\ L_i &= \sum_{j \in P_i} a_{ij} l_j + \sum_{j \in N_i} a_{ij} u_j, \end{aligned} \quad (3.2)$$

where  $P_i = \{j; a_{ij} > 0\}$ , and  $N_i = \{j; a_{ij} < 0\}$ ,  $L_i$  may be 0, and  $U_i$  may be  $+\infty$ .

*Step 2.* Test whether  $U_i \leq b_i$ ,  $i = 1, 2, \dots, m$ . The  $i$ th constraint  $\sum_{j=1}^n a_{ij} x_j \leq b_i$  is redundant if  $U_i \leq b_i$ .

*Example 3.1.* Consider the following LPP:

$$\begin{aligned}
 \max \quad & z = 4x_1 + 2x_2 + x_3, \\
 \text{subject to} \quad & 2x_1 + x_2 + x_3 \leq 30, \\
 & 3x_1 + x_2 + x_3 \leq 26, \\
 & x_2 + x_3 \leq 13, \\
 & x_1 + 2x_2 + x_3 \leq 45, \\
 & 0 \leq x_1 \leq 8.67, \\
 & 0 \leq x_2 \leq 13, \\
 & 0 \leq x_3 \leq 13.
 \end{aligned} \tag{3.3}$$

*Solution 1.*

*Step 1.* Define

$$\begin{aligned}
 P_1 &= \{1, 2, 3\}, & N_1 &= \varphi, \\
 P_2 &= \{1, 2, 3\}, & N_2 &= \varphi, \\
 P_3 &= \{2, 3\}, & N_3 &= \varphi, \\
 P_4 &= \{1, 2, 3\}, & N_4 &= \varphi,
 \end{aligned} \tag{3.4}$$

where  $\varphi$  is the empty set.

*Step 2.* Compute

$$\begin{aligned}
 L_1 &= 0, & U_1 &= 17.34 + 13 + 13 = 43.34, \\
 L_2 &= 0, & U_2 &= 26.01 + 13 + 13 = 52.01, \\
 L_3 &= 0, & U_3 &= 13 + 13 = 26, \\
 L_4 &= 0, & U_4 &= 8.67 + 26 + 13 = 47.67.
 \end{aligned} \tag{3.5}$$

Since all  $U_i \geq b_i$ ,  $i = 1, 2, 3, 4$ , there is no redundant constraint found by this method.

### **3.2. Linear Programming Method**

Caron et al. [6] developed an algorithm for identifying redundant constraints. This method will take more computational effort to identify the redundant constraints. To identify the redundant constraints, the left-hand side of each constraint is optimized subject to the remaining constraints. The optimal objective functional value is compared with the right-hand side value of corresponding constraints to decide if it is redundant or not. In this method, the objective function of the original LPP is not considered.

Let  $\mathbf{P}$  denote the given linear programming problem.

Let  $P_i$  denote the LP Problem without the  $i$ th constraint  $\sum_{j=1}^n a_{ij}x_j \leq b_i$  of  $P$  and let the objective function of LP problem  $P_i$  be  $\max z_i = \sum_{j=1}^n a_{ij}x_j$ .

*Step 1.* Find the optimal objective function value to the problem  $P_i$ ,  $i = 1, 2, \dots, m$ , by using the simplex method. Let  $z_i$  be the optimal objective function value of problem  $P_i$ .

*Step 2.* Check whether  $z_i \leq b_i$ . The  $i$ th constraint  $\sum_{j=1}^n a_{ij}x_j \leq b_i$  is redundant if  $z_i \leq b_i$ . Otherwise, it is not redundant.

*Example 3.2.* Consider the Example 3.1 presented in Section 3.1.

*Solution 2.* By solving the above Example 3.2, we get  $z = 43.33$ ,  $x_1 = 4.33$ ,  $x_2 = 13.00$ , and  $x_3 = 0$ .

*Step 1.* For  $i = 1, 2, 3, 4$ , consider the problem  $P_i$  as follows:

$$\begin{aligned}
 \max \quad & z_i = \text{left-hand side of constraint } i, \\
 \text{subject to} \quad & 3x_1 + x_2 + x_3 \leq 26, \\
 & x_2 + x_3 \leq 13, \\
 & x_1 + 2x_2 + x_3 \leq 45, \\
 & x_1, x_2, x_3 \geq 0.
 \end{aligned} \tag{3.6}$$

Using the simplex method,

the solution of problem  $P_1$  is  $z_1 = 21.67$ ,  $x_1 = 4.33$ ,  $x_2 = 13.00$ , and  $x_3 = 0.00$ .

the solution of problem  $P_2$  is  $z_2 = 45$ ,  $x_1 = 15$ ,  $x_2 = 0.00$ , and  $x_3 = 0.00$ .

the solution of problem  $P_3$  is  $z_3 = 26$ ,  $x_1 = 0.0$ ,  $x_2 = 19.00$ , and  $x_3 = 7.00$ .

the solution of problem  $P_4$  is  $z_4 = 30.33$ ,  $x_1 = 4.33$ ,  $x_2 = 13.0$ , and  $x_3 = 0.00$ .

*Step 2.* Here  $z_1 < b_1$ ,  $z_2 > b_2$ ,  $z_3 > b_3$ , and  $z_4 < b_4$ . Therefore, the constraints (1) and (4) are redundant.

### 3.3. Deterministic Method

Telgan [15] developed an algorithm for identifying redundant constraints and implicit equalities in system of linear constraints using minimum ratio criteria as in the simplex method.

#### *Procedure of the Method*

Assume that a basic feasible solution is given, and the corresponding contracted simplex tableau is set up. Let  $y_{ij}$  be the entries of this tableau with  $y_{i0}$  the right-hand side coefficients, and let  $x_j^B$  and  $x_j^N$  represent the basic and nonbasic variables, respectively. Let  $H$  be the set of all indices of constraints to be identified as either redundant or nonredundant.

Table 1

	$x_1$	$x_2$	$x_3$	RHS
$u_1$	2	1	1	30
$u_2$	3	1	1	26
$u_3$	0	1	1	13
$u_4$	1	2	1	45

Here  $H = \{1, 2, 3, 4\}$ .

Let  $u_k$  be the slack variable corresponding to the  $k$ th constraint.

*Step 1.* If the solution is nondegenerate, all  $u_k = x_j^N$  correspond to nonredundant inequalities, remove  $k$  from  $H$  and continue with Step 3.

*Step 2.* In a degenerate solution check all nonbasic variables  $u_k = x_p^N$  with  $k \in H$ . Check the property  $y_{ip} \geq 0$  for all  $i$  with  $y_{i0} = 0$ . If this holds, then the constraint  $A_k x \leq b_k$  is not redundant, and remove  $k$  from  $H$ .

*Step 3.* Check all basic variables  $u_k = x_r^B$  with  $k \in H$  for the property  $y_{ij} \leq 0$  for all  $j$ . If this holds, then the constraint  $A_k x \leq b_k$  is redundant, and remove  $k$  from  $H$ .

*Step 4.* Check all basic variables  $u_k = x_i^B$  with  $k \in H$  for the property  $y_{r0}/y_{rs} = \min\{y_{i0}/y_{is}, y_{is} > 0\}$  is attained at a unique  $r$  for some  $s$ . If this holds, the constraint  $A_k x \leq b_k$  is not redundant, and remove  $k$  from  $H$ .

*Step 5.* If  $H = \varnothing$ , then stop. Else, go to Step 6.

*Step 6.* If there is no basic variable  $u_k = x_i^B$  with  $k \in H$ , then introduce a nonbasic variable  $u_k = x_j^N$  with  $k \in H$  (e.g., the one with the smallest index  $k$ ) into the basis and continue with Step 1.

*Step 7.* Select a basic variable  $u_k = x_j^B$  with  $k \in H$  (e.g., the one with the smallest index  $k$ ) and perform a feasible pivot step in column  $p$  with  $y_{ip} = \max y_{ij}$ . Continue with Step 1.

*Example 3.3.* Consider Example 3.1 presented in Section 3.1.

*Solution 3.*

*Iteration 1.* Contracted simplex table, see Table 1.

*Step 1.*  $x_1 \geq 0$ ,  $x_2 \geq 0$ , and  $x_3 \geq 0$  are not redundant.

*Step 4.* Now, divide the RHS values by the first column and take the minimum of it  $\min\{15, 8.66, -, 45\} = 8.66$ , which corresponds to the 2nd row. Therefore, constraint (2) is not redundant. Now,  $H = \{1, 3, 4\}$ .

*Step 7.* Select  $u_1$  and find the maximum  $\{2, 1, 1\} = 2$ , which corresponds to the 1st column. Therefore, pivoting on  $y_{21}$ , we obtain, see what is Table 2.

*Iteration 2.* Divide the RHS values by the 2nd column and take minimum of it.  $\min\{38, 26, 13, 21.8\} = 13$ , which corresponds to the 3rd row. Therefore, constraint (3) is not redundant. Now,  $H = \{1, 4\}$ .

Table 2

	$u_2$	$x_2$	$x_3$	RHS
$u_1$	$-2/3$	$1/3$	$1/3$	$38/3$
$x_1$	$1/3$	$1/3$	$1/3$	$26/3$
$u_3$	$0$	$1$	$1$	$13$
$u_4$	$-1/3$	$5/3$	$2/3$	$109/3$

Step 4. Selecting  $u_1$ ,  $u_3$  we cannot pivot. So select  $u_4$  and the maximum  $\{-1/3, 5/3, 2/3\} = 5/3$  corresponds to the 2nd column. Therefore, pivoting on  $y_{32}$ , we obtain, see what is Table 3.

Step 3. In the first and last row, all the coefficients are  $\leq 0$ .

Step 5.  $H = \varphi$ . Therefore, Constraints (1) and (4) are redundant.

### 3.4. Stojković and Stanimirović Method

This method is proposed by Stojković and Stanimirović [13]. It is a simple method. It verifies the existence of the saddle point of payoff matrix for the game problem by applying the maxmin and minmax principles.

#### Procedure of the Method

Step 1. Compute  $d_{ij} = a_{ij}/b_{ic_j}$ ,  $i = 1, 2, \dots, m$ , and  $j = 1, 2, \dots, n$ .

Step 2. If  $\max_{1 \leq i \leq m} \min_{1 \leq j \leq n} d_{ij} = \min_{1 \leq i \leq n} \max_{1 \leq i \leq m} d_{ij}$ , then there are no redundant constraints. Stop.

Else, if there exist  $k$  and  $l$  such that  $d_{kj} \leq d_{lj}$ , for all  $j = 1, 2, \dots$ , then the  $k$ th constraint is redundant.

Example 3.4. Consider the Example 3.1 presented in Section 3.1.

Solution 4.

Step 1.

$$\begin{aligned}
 d_{ij} &= \begin{bmatrix} \frac{2}{120} & \frac{1}{60} & \frac{1}{30} \\ \frac{3}{104} & \frac{1}{52} & \frac{1}{26} \\ 0 & \frac{1}{52} & \frac{1}{26} \\ \frac{1}{180} & \frac{2}{90} & \frac{1}{45} \end{bmatrix} \\
 &= \begin{matrix} \min \\ 0.01666 & 0.01666 & 0.0333 \\ 0.02885 & 0.01923 & 0.0385 \\ 0.0 & 0.0385 & 0.0769 \\ 0.0055 & 0.0222 & 0.0222 \end{matrix} \\
 &\max \begin{matrix} 0.01666 \\ 0.01923 \\ 0.0 \\ 0.0055 \end{matrix} \\
 &\begin{matrix} 0.02885 & 0.0385 & 0.0769 \end{matrix}
 \end{aligned} \tag{3.7}$$



Table 3

	$u_2$	$u_3$	$x_3$	RHS
$u_1$	$-2/3$	$-1/3$	0	$25/3$
$x_1$	$1/3$	$-1/3$	0	$13/3$
$x_2$	0	1	1	13
$u_4$	$-1/3$	$-5/3$	-1	$44/3$

Step 2.

$$\max_{1 \leq i \leq 4} \min_{1 \leq j \leq 3} d_{ij} = 0.01923,$$

$$\min_{1 \leq j \leq 3} \max_{1 \leq i \leq 4} d_{ij} = 0.02855, \quad (3.8)$$

$$\max_{1 \leq i \leq 3} \min_{1 \leq j \leq 4} d_{ij} \neq \min_{1 \leq j \leq 4} \max_{1 \leq i \leq 4} d_{ij}.$$

There must be at least one redundant constraint in the above problem.

Here  $d_{1j} \leq d_{2j}$ ,  $j = 1, 2, 3$ , hence the constraint (1) is redundant.

### 3.5. Heuristic Method

Paulraj et al. [14] proposed a heuristic method to identify redundant constraint by using the intercept matrix of constraints of a linear programming problem.

#### Procedure of the Method

Step 1. Let  $I$  be the set of subscripts associated with the initial basic variables (slack variables). Initially let that set be  $I = \{1, 2, \dots, m\}$ .

Let  $J$  be the set of subscripts associated with the initial decision variables. Initially let that set be  $J = \{1, 2, \dots, n\}$ .

Step 2. Construct an intercept matrix " $\theta$ " using the following relationship

$$\theta_{ji} = \frac{b_i}{a_{ij}}, \quad a_{ij} > 0, \quad \text{for } j \in J, i \in I. \quad (3.9)$$

Step 3. Determine the entering variables making use of the following steps.

(i) Calculate  $z_j - c_j = C_B B^{-1} a_j - c_j$  for all nonbasic variables  $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$ .

(ii) Let  $\beta_j = \min\{\theta_{ji}\}$ , for  $j \in J$ ,  $i \in I$ .

(iii) Compute  $z'_j - c'_j = \beta_j(z_j - c_j)$  for  $j \in J$ .

Step 4. (i) Let  $z'_k - c'_k = \min_{j \in J} \{z'_j - c'_j\}$ .

(ii) If  $z'_k - c'_k \geq 0$ , then the problem has no redundant constraints and stop.

(iii) Otherwise, take away the element  $k$  from the set  $J$ , that is,  $J = J - \{k\}$ .

(iv) Let  $\theta_{kl} = \min_{i \in I} \{\theta_{ki}\} = \beta_k$ .

Table 4

Decision variables	Basic variables				$z_j - c_j$	$\beta_j$	$z'_j - c'_j$
	$S_1$	$S_2$	$S_3$	$S_4$			
$X_1$	15	8.67	—	45	-4	8.67	-34.68
$X_2$	30	26	13	22.5	-2	13	-26
$X_3$	30	26	13	45	-1	13	-13

Table 5

Iteration number	$k$	$J$	$l$	$I$	$J$
1	1	{2,3}	2	{1,3,4}	{2,3}
2	2	{3}	3	{1,4}	{3}
3	3	$\Phi$	3	{1,4}	$\varphi$

(v) Take away the element  $l$  from the set  $I$ , that is,  $I = I - \{l\}$ .

(vi) Find  $p$  such that  $\min \{\theta_{pl}\} = \beta_p$  for  $p \in J$ . If so, take away such  $p$  elements from the set  $J$ , that is,  $J = J - \{p\}$ .

Step 5. If  $J = \varphi$ , then go to Step 6. Otherwise, go to Step 4.

Step 6. If  $I = \varphi$ , then the problem has no redundant constraints and stop. Otherwise, all the constraints  $i \in I$  whose intercepts on coordinate axis  $j$  satisfy the condition  $\theta_{ji} \geq \max\{\beta_j\}$  where  $i \in I$  and  $j = 1, 2, \dots, n$  are redundant. Stop.

Example 3.5. Consider the Example 3.1 presented in Section 3.1

Solution 5.

Step 1. (i)  $I = \{1, 2, 3, 4\}$ , (ii)  $J = \{1, 2, 3\}$ .

Step 2. [see Table 4].

Step 3. [see Table 4].

Step 4. [see Table 5].

Step 5.  $J = \varphi$ . Constraints (1) and (4) are redundant constraints.

#### 4. Numerical Results

The comparative results of the five methods are presented in the following tables. Table 6 shows the comparison results of small-scale problems, Table 7 shows the comparison results of medium-scale problems, and Table 8 shows the comparison results of netlib problems [18]. Comparison results of large size problems from OR library [19] are presented in Table 9.

In Figures 2, 3, 4 and 5, the comparative results of small scale problems, medium scale problems, netlib

problems and large size problems are shown graphically. Figure 2 shows that the heuristic and linear programming methods identify more redundant constraints than the other three methods. Figure 3 indicates that the Stojkovic and stanimirovic method identifies

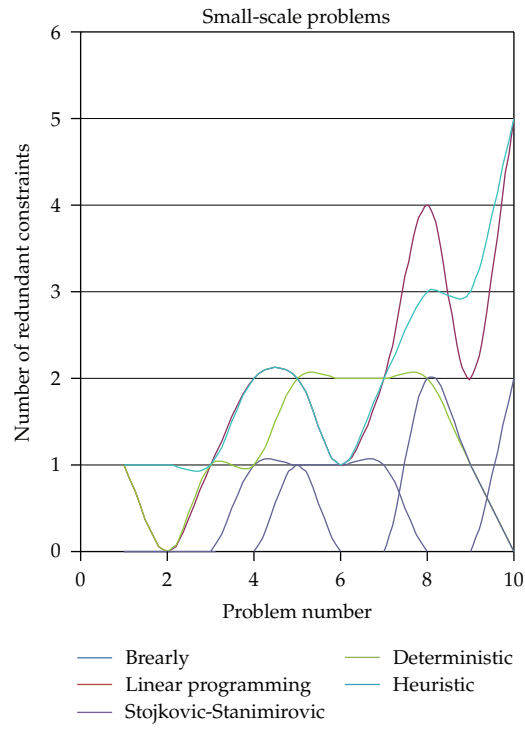


Figure 2

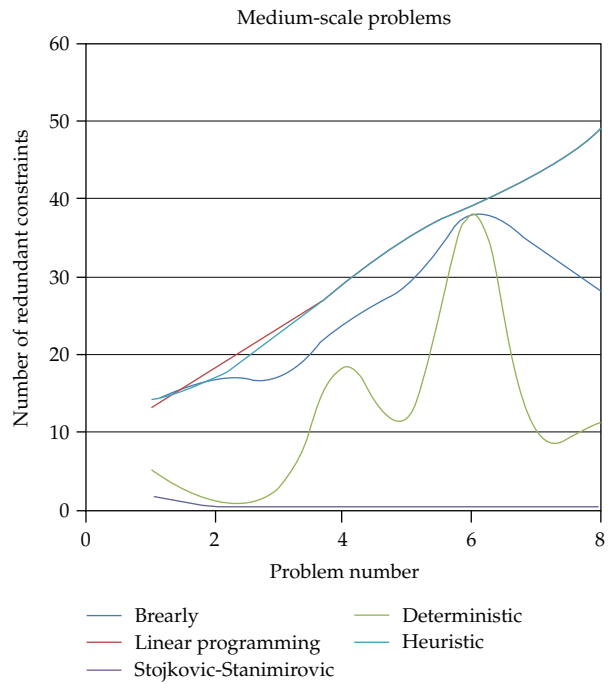


Figure 3

Table 6: Comparison of five methods: small-scale problems.

S. no.	1	2	3	4	5	6	7	8	9	10
No. of constraints	3	3	3	4	4	3	4	5	5	7
No. of variables	2	2	2	3	3	3	5	2	4	10
Brearly's method (constraint no.)	0	0	0	1 (4)	1 (4)	1 (3)	1 (4)	0	0	2 (3,7)
With redundant	93	93	93	311	311	336	336	505	535	1894
Without redundant	93	93	93	167	167	186	186	505	535	883
Linear programming method (constraint no.)	1 (3)	0	1 (3)	2 (3,4)	2 (1,4)	1 (3)	2 (3,4)	4 (2,3,4,5)	2 (2,4)	5 (2,3,4,6,7)
With redundant	93	93	93	311	311	336	336	505	535	1894
Without redundant	42	93	42	76	75	97	88	70	50	161
Deterministic method (constraint no.)	1 (3)	0	1 (3)	1 (3)	2 (1,4)	2 (2,3)	2 (3,4)	2 (3,5)	1 (2)	0
With redundant	93	93	93	311	311	336	336	505	535	1894
Without redundant	42	93	42	168	75	88	88	159	50	1894
Stojković-Stanimirović method (constraint no.)	0	0	0	0	1 (1)	0	0	2 (2,3)	1 (2)	0
With redundant	93	93	93	311	311	336	336	505	535	1894
Without redundant	93	93	93	311	167	178	336	159	167	1894
Heuristic method (constraint no.)	1 (3)	1 (1)	1 (3)	2 (3,4)	2 (1,4)	1 (3)	2 (3,4)	3 (3,4,5)	3 (2,3,4)	5 (2,3,4,6,7)
With redundant	93	93	93	311	311	336	336	505	535	1894
Without redundant	42	42	42	76	75	97	88	70	50	161

**Table 7:** Comparison of five methods: medium-scale problems.

S. no.	Size of the problem		Number of redundant constraints identified by				
	no. of constraints	no. of variables	Brearly's method	linear programming method	deterministic method	Stojković-Stanimirović method	heuristic method
1	16	6	14	13	5	1	14
2	20	5	17	18	1	0	17
3	25	6	17	23	3	0	23
4	30	3	24	29	18	0	29
5	37	5	29	35	12	0	35
6	40	2	38	39	38	0	39
7	45	3	34	43	10	0	43
8	50	5	28	49	11	0	49

**Table 8:** Comparison of five methods: netlib problems.

S. no. and File name	Size of the problem		Number of redundant constraints identified by				
	No. of constraints	No. of variables	Brearly's method	Linear programming method	Deterministic method	Stojković-Stanimirović method	Heuristic method
(1) afiro	20	20	9	3	0	0	4
(2) fit1d	24	24	2	10	0	0	13
(3) fit2d	25	25	0	19	0	7	19
(4) sc50b	28	28	0	7	0	0	10
(5) sc50a	29	29	1	11	0	2	11
(6) kb2	39	39	3	13	0	14	14
(7) vtpbase	51	51	1	21	0	4	30
(8) bore3d	52	52	42	17	0	22	18

no redundant constraints where as deterministic method identifies very low redundant constraints compared with other methods. Figure 4 indicates that the deterministic method identifies nothing; brearly's, Stojkovic and stanimirovic methods identify very low redundant constraints and the remaining methods more or less coincide. Figure 5 shows that the Stojkovic and stanimirovic method identifies very less redundant constraints where as heuristic method identify more redundant constraints than the others.

The tables deal with the identification of the number of redundant constraints in linear programming problems by using the five methods. It is very easy to identify quickly the best method in finding redundant constraints of LP problems. Heuristic method seems to be less time consuming, and it requires less computational effort. It also finds more redundant constraints when compared with the other four methods. So this method would be easy and reliable method for identifying redundant constraints. Even though the LP method identifies more redundant constraints, it needs more computational work and takes more time. Brearly's method identifies less redundant constraints with less computational effort than heuristic and LP methods. Deterministic methods identify more redundant constraints with more computational effort. So time consumption is bigger when compared with heuristic method. Stojković and Stanimirović identified a smaller number of redundant constraints than the others.

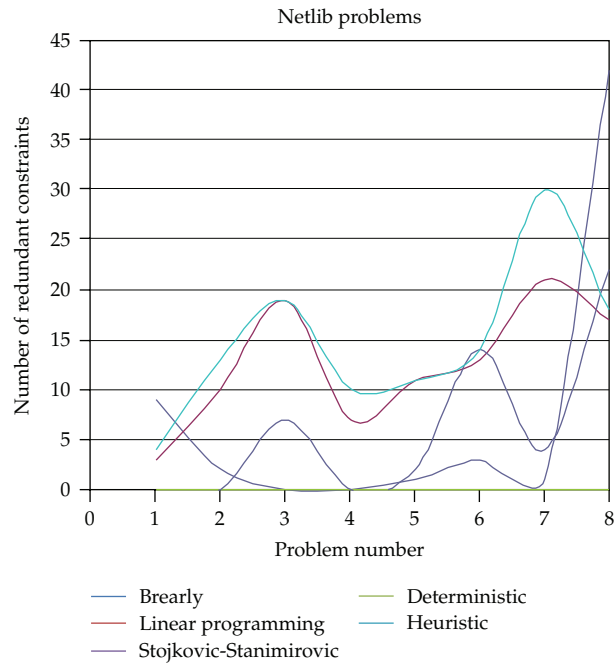


Figure 4

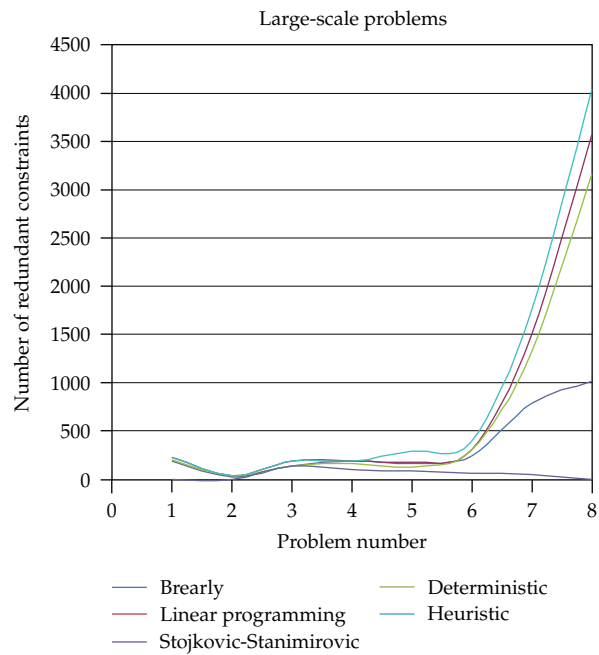


Figure 5

**Table 9:** Comparison of five methods: Large Size Problems.

S. no. and File name	Size of the problems		Number of redundant constraints identified by				
	No. of constraints	No. of variables	Brearly method	Linear programming method	Deterministic method	Stojković Stanimirović method	Heuristic method
(1) scpcyc06	240	192	197	235	201	0	236
(2) scpe2	50	500	31	40	38	0	43
(3) scp43	200	1000	142	195	136	143	196
(4) scp52	200	2000	187	197	163	98	198
(5) scpa3	300	3000	165	181	123	93	293
(6) scpd3	400	4000	243	305	315	64	395
(7) scpcyc08	1792	1024	800	1512	1328	54	1780
(8) scpc1r13	4095	715	1023	3608	3204	0	4083

The efficiency of the algorithms was also tested by solving the first set of Linear Programming Problems mentioned before and after removing the redundant constraints, identified by each method. Table 6 gives the computational results.

## 5. Conclusions

In this paper, the heuristic approach [14] for identifying redundant constraints has been compared with other four methods. Each method has its own role in viewing computational effort and time factor. Linear programming method [6] and deterministic method [15] more or less coincide with heuristic method in most of the problems. Brearly et al. [4] method depends on the upper and lower bounds of the decision variables for identifying the redundant constraints. Hence, the heuristic method is more useful than the other methods.

## References

- [1] E. D. Andersen and K. D. Andersen, "Presolving in linear programming," *Mathematical Programming. Series B*, vol. 71, no. 2, pp. 221–245, 1995.
- [2] M. L. Balinski, "An algorithm for finding all vertices of convex polyhedral sets," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 1, pp. 72–88, 1961.
- [3] J. C. G. Boot, "On trivial and binding constraints in programming problems," *Management Science*, vol. 8, no. 4, pp. 419–441, 1962.
- [4] A. L. Brearley, G. Mitra, and H. P. Williams, "Analysis of mathematical programming problems prior to applying the simplex algorithm," *Mathematical Programming*, vol. 8, pp. 54–83, 1975.
- [5] A. Boneh, S. Boneh, and R. J. Caron, "Constraint classification in mathematical programming," *Mathematical Programming*, vol. 61, no. 1, pp. 61–73, 1993.
- [6] R. J. Caron, J. F. McDonald, and C. M. Ponic, "A degenerate extreme point strategy for the classification of linear constraints as redundant or necessary," *Journal of Optimization Theory and Applications*, vol. 62, no. 2, pp. 225–237, 1989.
- [7] T. Gal, "Weakly redundant constraints and their impact on post optimal analysis," *European Journal of Operational Research*, vol. 60, pp. 315–326, 1979.
- [8] P. O. Gutman and I. Isolovich, "Robust redundancy determination and evaluation of the dual variables of linear programming problems in the presence of uncertainty, on the generalized wolf problem: preprocessing of nonnegative large scale linear programming problems with group constraints," *Technion-Israel Institute of Technology*, vol. 68, no. 8, pp. 1401–1409, 2007.
- [9] H. J. Greenberg, "Consistency, redundancy, and implied equalities in linear systems," *Mathematics and Artificial Intelligence*, vol. 17, pp. 37–83, 1996.

- [10] M. H. Karwan, V. Lotfi, J. Telgen, and S. Zionts, Eds., *Redundancy in Mathematical Programming: A State-of-the-Art Survey*, Springer, Berlin, Germany, 1983.
- [11] T. H. Mattheis, "An algorithm for determining irrelevant constraints and all vertices in systems of linear inequalities," *Operations Research*, vol. 21, pp. 247–260, 1973.
- [12] J. Lisy, "Metody pro nalezeni redundantnich omezeni v ulohach linerniho programovani. Ekonomicko MatematicS," *Obzor*, vol. 7, no. 3, pp. 285–298, 1971.
- [13] N. V. Stojković and P. S. Stanimirović, "Two direct methods in linear programming," *European Journal of Operational Research*, vol. 131, no. 2, pp. 417–439, 2001.
- [14] S. Paulraj, C. Chellappan, and T. R. Natesan, "A heuristic approach for identification of redundant constraints in linear programming models," *International Journal of Computer Mathematics*, vol. 83, no. 8-9, pp. 675–683, 2006.
- [15] J. Telgen, "Identifying redundant constraints and implicit equalities in system of linear constraints," *Management Science*, vol. 29, no. 10, pp. 1209–1222, 1983.
- [16] G. L. Thompson, F. M. Tonge, and S. Zionts, "Techniques for removing nonbinding constraints and extraneous variables from linear programming problems," *Management Science*, vol. 12, no. 7, pp. 588–608, 1996.
- [17] S. Zionts, *Size of reduction techniques of linear programming and their applications*, Dissertation, Carnegie Institute of Technology, 1965.
- [18] "Netlib problems," <http://ftp.zib.de/pub/mp-testdata/madlib/node2.html>.
- [19] J. E. Beasley, "Distributing test problems by electronic mail," *Journal of Operational Research Society*, vol. 41, pp. 1069–1107, 1990.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

