

Hybrid Method with ANN application for Linear Programming Problem

L. R. Aravind Babu*

Dept. of Information Technology, CSE Wing, DDE Annamalai University, Annamalainagar, Tamilnadu, India

ABSTRACT: This paper presents a new hybrid approach of Neural Network (NN) to solve Linear Programming Problem. When formulating an LP model, systems analysts and researchers often include all possible constraints although some of them may not be binding at the optimal solution. The presence of redundant data consumes great computational effort. Various redundant identification methods are used to reduce the computational effort but the accuracy of the system is not concerned. For which I had developed the neural network based structure which concern both accuracy and computational effort. The training and learning of the data are done here before apply the formal methodology

KEYWORDS: Linear Programming, Back-propagation, heuristic, neural network, redundant constraints, optimization.

I. INTRODUCTION

The most important techniques used in modelling and solving practical optimization problems that arise in industry, commerce, and management is Linear Programming. This is a mathematical models used to represent real life situations in the form of linear objective function and constraint. There are different methods are available to solve Linear Programming Problems. All methods are concentrates only on computational effort but the accuracy concern is very less. When formulating a Linear Programming (LP) model, researchers often include all possible constraints although some of them may not be binding at the optimal solution. These redundant constraints will not change the optimum solution but will increase the computational effort. Many researchers have proposed several algorithms [5, 6, and 12] for identifying the redundant constraints in Linear Programming (LP) models. The widely used methodology for Linear Programming (LP) problems is Revised Simplex Method [3], a unilabiate search technique. This method suffers the drawback of slow convergence with the tendency of variable popping in and out of the basis matrix. These results more number of iterations and increase the computational effort and time. This results the development of a new algorithm, called Complex Algorithm. The bounded complex algorithm [2] is found to be more effective than revised simplex method on comparison based on the number of iterations; computational time and number of multiply/divide operations. This leads to achieve optimality in accuracy and also in computational effort, the new algorithm/method has been developed called Hybrid Algorithm. The developed hybrid algorithm/method trains the constraint and parameter with the help of ANN before applying the formal methodology.

II. MATERIALS AND METHODS

Complex Algorithm for Bounded Variables

The Complex Algorithm for Bounded Variables [2], a univariate search technique is adopted, but the entering variables are arranged not only based on the maximum contribution to the objective function but also on the type of the constraints. In the Complex bounded algorithm maximum change criterion as well as the type of constraints are used to arrange the variables in the order in which it has to enter the basis. The leaving variable is selected based on the bound of the variables. This led to the reduction in the number of iterations and hence in the saving of computational effort. The presence of redundant constraints does not alter the optimal solution but they may consume extra computational effort. The identification of redundant constraint in Linear Programming Problem using various methods [5] have been studied and discussed about the efficiency of each method.

Algorithm to Detect Redundant Constraints

A simple heuristic algorithm [6] has been developed to identify the redundant constraint in a class a linear programming problem. This reduces the original model to a lesser dimension before solving them. The heuristic algorithm performs very efficiently and has the additional advantage of ease of implementation. An available code to

solve the Linear Programming using Karmarkar's algorithm can be easily adopted with minimal effort. Computational efficiency of newly developed algorithm can be established only when they are compared with the existing algorithms.

Study of Redundant Constraints Identification Methods in Linear Programming Problems

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 Linear Programming (LP) problem. Quite often large-scale Linear Programming (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. Nevertheless, they may consume extra computational effort. Many researchers have proposed different approaches for identifying the redundant constraints [4] in linear programming problems. This work compares five of such methods and discusses the efficiency of each method by solving various size Linear Programming (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 work. Many methods are available in the literature to identify the redundant constraints in linear programming problems. In this work, the following five methods are discussed and compared

- 1) Bounds method
- 2) Linear Programming method
- 3) Deterministic method
- 4) Stojkovic and Stanimirovic method
- 5) Heuristic method.

In this work, the Heuristic approach 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 and Deterministic method are more or less coincides with Heuristic method in most of the problems. The Conventional method [8, 9] 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.

Heuristic Approach to Deduct the Redundant Constraints in Linear Programming Model

Linear programming is one of the most important techniques adapted to model, to allocate resources and solve practical optimization problems. Whenever Systems Analysts model, particularly large-scale linear programming problems, it is quite possible that some redundant constraints may creep in due to inadvertency. These unidentified redundant constraints when present in the model will waste computational effort and lower the efficiency of computation. Karmarkar's Algorithm is a polynomial time algorithm to solve large-scale linear programming problems. This approach has a great potential for solving large to very large scale linear programming problems which are sometimes far beyond the reach of the Simplex method. Karmarkar's algorithm transforms the original linear programming model into another standard model named as Karmarkar's model. This gives overview of a simple heuristic algorithm to identify the redundant constraints a priori in the Karmarkar's model so as to minimize the dimension of the problems. While formulating a linear programming model, systems analysts and researchers often tend to include inadvertently all possible constraints although some of them may not be binding at the optimal solution. There is a possibility of including redundant constraints. Presence of redundancies will waste computational efforts. If these are identified and eliminated from the original problem, the number of iterations needed to reach the optimal solution can be reduced. This process of eliminating redundancies from a problem is called as model reduction. An attempt is made in this work to detect the redundant constraint of the Karmarkar's model a priori using gradient matrix of the constraints. The heuristic approach to detect the redundant constraint using the gradient method is discussed.

Although radically differs from the Simplex method, Karmarkar's algorithm does share a flow of the same characteristics. Karmarkar's algorithm generates a sequence of points in the feasible region whose costs approach the best cost. In the end, it jumps to a vertex of no greater cost, which is the optimal. By using the proposed heuristic algorithm optimal solution is reached faster. A simple heuristic approach to detect the redundant constraints in the Karmarkar's model has been developed. The applicability of the gradient method to detect the redundancies has been explained. This works mainly focus on the development of model reduction algorithm on Karmarkar's model and compares the efficiency when it is applied. When the model reduction algorithm is applied on Karmarkar's model it speeds up the computation. It detects the redundancies compared with the actual.

Theoretical Background of Artificial Neural Network

Neural networks can be thought of as "black box" devices that accept inputs and produces outputs. Figure.1 shows a typical neural network structure consisting of three layers:

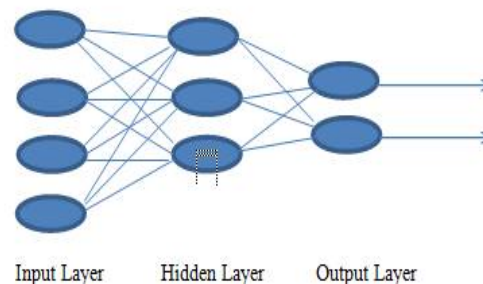


Figure. 1 Structure of Neural Network

Input Layer: A layer of neurons that receives information from external sources, and passes this information to the network for processing. These may be either sensory inputs or signals from other systems outside the one being modeled. **Hidden Layer:** A layer of neurons that receives information from the input layer and processes them in a hidden way. It has no direct connections to the outside world (inputs or outputs). All connections from the hidden layer are to other layers within the system. **Output Layer:** A layer of neurons that receives processed information and sends output signals out of the system. The number of input neurons corresponds to the number of input variables into the neural network, and the number of output neurons is the same as the number of desired output variables. The number of neurons in the hidden layer(s) depends on the application of the network. As inputs enter the input layer from an external source, the input layer becomes “activated” and emits signals to its neighbors (hidden layer) without any modification. Neurons in the input layer act as distribution nodes and transfer input signals to neurons in the hidden layer. The neighbors receive excitation from the input layer, and in turn emit an output to their neighbors (second hidden layer or output layer). Each input connection is associated to a quantity, called “a weight factor” or “aconnection strength”. The strength of a connection between two neurons determines the relative effect that one neuron can have on another. The weight is positive if the associated connection is excitatory and negative if the connection is inhibitory.

Learning or Training of Neural Network

The property that is of primary significance for a neural network is the ability of the network to learn from its environment, and to improve its performance through learning. The improvement in performance takes place over time in accordance with some prescribed measure. A neural network learns about its environment through an interactive process of adjustments applied to its synaptic weights and bias levels. Ideally, the network becomes more knowledgeable about its environment after each iteration of the learning process. Therefore, Learning is defined as a process by which the free parameters of a neural network are adapted through a process of simulation by the environment in which the network is embedded.

III. PROPOSED WORK - HYBRID METHOD

A most excellent algorithm must be optimal in both computational effort and accuracy. All the above methodologies concentrated only on achieving best computational effort. Whereas accuracy of these systems may goes down due to reducing number of constraints and number of iteration. To achieve optimality in computational effort and also in accuracy, we proposed a hybrid algorithmic structure. If we reduce number of iterations, the time complexity become optimal, but the accuracy of the system has reduced. To improve the accuracy, we suggested training and learning of parameters and constraints. This training is possible in real world application by applying artificial neural network (ANN).The ANN has applied for various applications to predict forecasting of parameters, constraints and also to obtain optimality in real world parameters. The following figure proposes the Hybrid Structure with ANN to achieve the accuracy for any process model. The Complex Algorithm/ Redundant constraints is used to achieve optimal in computational effort.

Proposed Hybrid Structure - Complex algorithm / redundant constraints is used to achieve optimal in computational effort whereas ANN is applied to achieve accuracy in any process model.

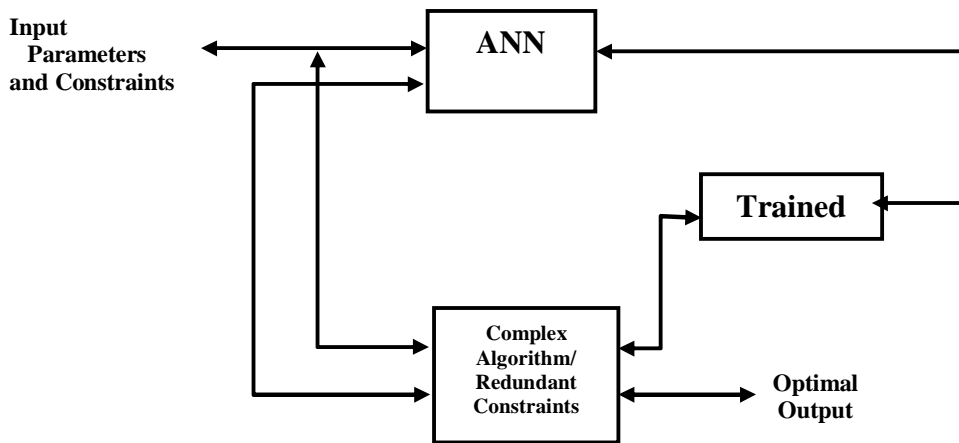


Fig. 2 Proposed Hybrid Model with ANN

This training is possible in real world application by applying Artificial Neural Network (ANN). The ANN has been applied for various applications to predict forecasting of parameters, constraints and also to obtain optimality in real world parameters.

Proposed Algorithm:

- Step1:** If the predicted day / time are belonging to class-1 environment, go to step 4 otherwise, it is class-2 environment.
- Step2:** Construct input information using the input parameters and constraints using any formal or conventional methods like complex algorithm and or identifying redundant constraints before the predicted day or time.
- Step3:** Forecast the appropriate parameters.
- Step4:** In case that the predicted day or time belongs to a class-1 situation, the value of parameters may vary.
- Step5:** Forecast the appropriate parameters with the physical sensitivities calculated in step 4.

After taking the above steps, the normalized value of parameters is calculated from the data obtained. Thereafter, the parameters for any class of situation are forecasted from the normalized value.

IV. RESULTS

Comparison of Revised Simplex Method with Complex Bounded Algorithm

The bounded complex algorithm a search technique is adopted but the entering variables are arranged not only based on the maximum contribution if that variables to the objective function but also on the type of the constraints.

Table 1: Findings out the time difference between the two algorithms

| S.No | Revised Simplex Algorithm | | Complex Algorithm (Bounded) | | Difference in time |
|------|---------------------------|------------------|-----------------------------|------------------|--------------------|
| | Time in sec | No of iterations | Time in sec | No of iterations | |
| 1. | 0.1649 | 3 | 0.0467 | 2 | 0.1182 |
| 2. | 0.1845 | 4 | 0.0549 | 2 | 0.1300 |
| 3. | 0.1945 | 4 | 0.1090 | 3 | 0.0855 |
| 4. | 0.2048 | 4 | 0.1195 | 2 | 0.0853 |
| 5. | 0.3425 | 5 | 0.1645 | 4 | 0.1780 |
| 6. | 0.1347 | 3 | 0.0346 | 2 | 0.1001 |

Table 2: Number of Variables with respect two algorithm

Identifying Redundant Constraints

| S.No | Revised Simplex Algorithm (Variables) | Complex Bounded Algorithm (Variables) |
|------|---------------------------------------|---------------------------------------|
| 1. | 250 | 191 |
| 2. | 159 | 111 |
| 3. | 251 | 167 |
| 4. | 404 | 283 |
| 5. | 530 | 342 |
| 6. | 190 | 100 |
| 7. | 187 | 113 |

While formulating a linear programming model, system analysts and researchers often tend to include all possible constraints although some of them may not be binding at the optimal solution. The presence of redundant constraints does not alter the optimal solution, nevertheless they may consume extra computational effort. Here we have compared the deterministic method with the other three methods. In most of the problems Deterministic method more or less coincides with Heuristic method. The second (Brearley et al.,1975), the identification of redundant constraints depend on the upper and lower bounds of the decision variables. If the upper and lower bounds for the decision variables are given nearer to the solution then we can identify the redundant constraints to some extent. Stojkovic and Stanimirovic method does not identifies redundant constraint properly in some cases. While comparing the Deterministic and Heuristic methods, in many problems Heuristic method finds more number of redundant constraints than the Deterministic method. This is because the Deterministic method involves no objection function.

Table 3: Comparison of four methods: Large size Problems

| S. No | Size of the problems | | Number of Redundant Constraints Identified by | | | |
|-------|----------------------|------------------|---|----------------------|------------------------------|------------------|
| | No. of Constraints | No. of Variables | Brearily Method | Deterministic Method | StojkovicStanimirovic Method | Heuristic Method |
| 1 | 240 | 192 | 197 | 201 | 0 | 236 |
| 2 | 50 | 500 | 31 | 38 | 0 | 43 |
| 3 | 200 | 1000 | 142 | 136 | 143 | 196 |
| 4 | 200 | 2000 | 187 | 163 | 98 | 198 |
| 5 | 300 | 3000 | 165 | 123 | 93 | 293 |
| 6 | 400 | 4000 | 243 | 315 | 64 | 395 |
| 7 | 1792 | 1024 | 800 | 1328 | 54 | 1780 |
| 8 | 4095 | 715 | 1023 | 3204 | 0 | 4083 |

The New Developed Hybrid Method – Proposed Work Result:

The Hybrid Algorithmic Structure that improves the accuracy of bounded variables in Linear Programming Problem model by suggesting the training and learning of parameters and constraints. The Training is possible in applications by applying Artificial Neural Network. I believe, the proposed algorithm shows increased performance in accuracy. The Hybrid Algorithm must be optimal in both computational effort and accuracy. All the above methodologies concentrated only on achieving best computational effort. Whereas, accuracy of these systems may go down due to reducing number of constraints and number of iteration. To achieve optimality in computational effort and also in accuracy, I proposed a Hybrid Algorithmic Method.

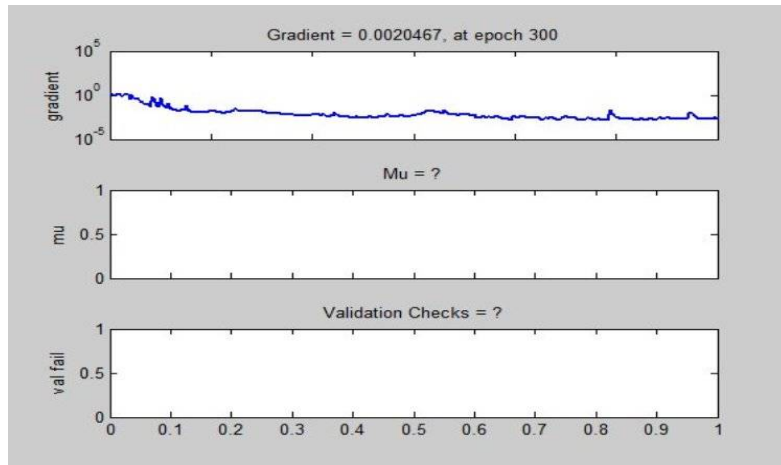


Figure: 3 Shows the training state of the network

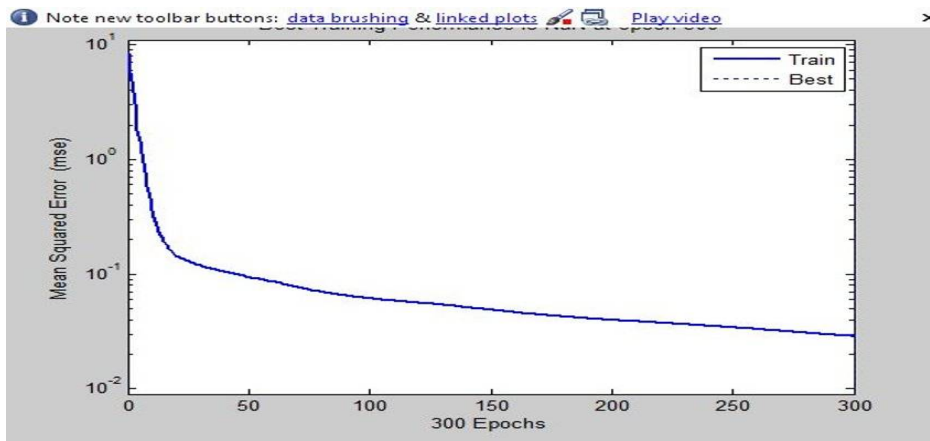


Figure: 4 Plot Performance of the Training State

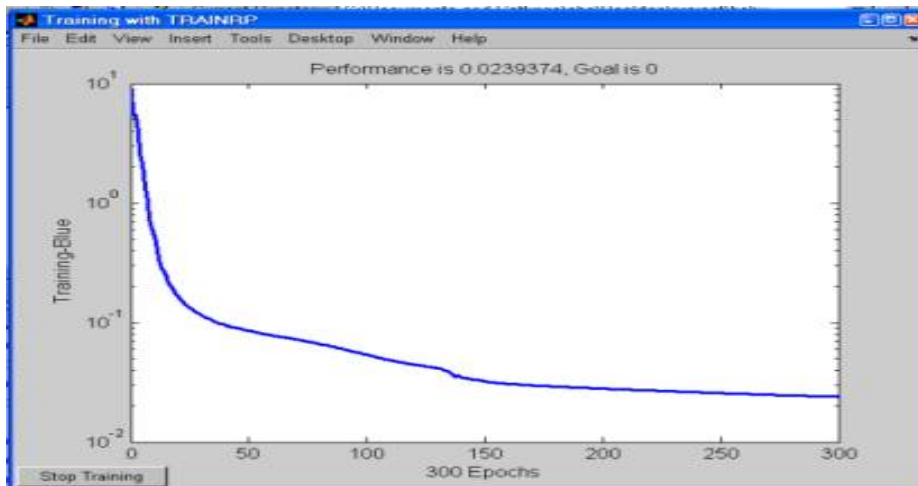


Figure: 5 Results of Back propagation in bounded variables and redundant constraints

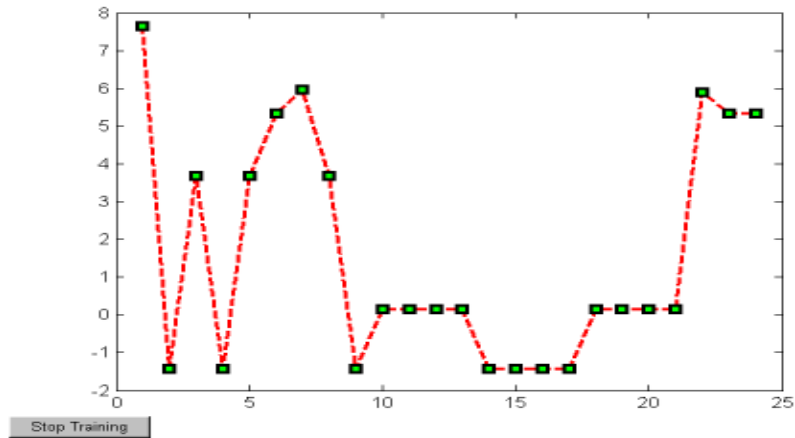


Figure: 6 Error rate in Linear Programming Problem Model

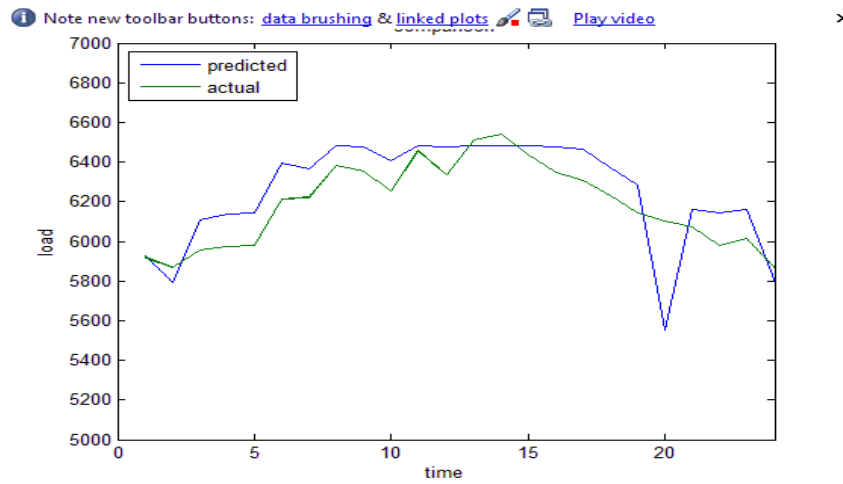


Figure: 7 Testing Phase with load versus time

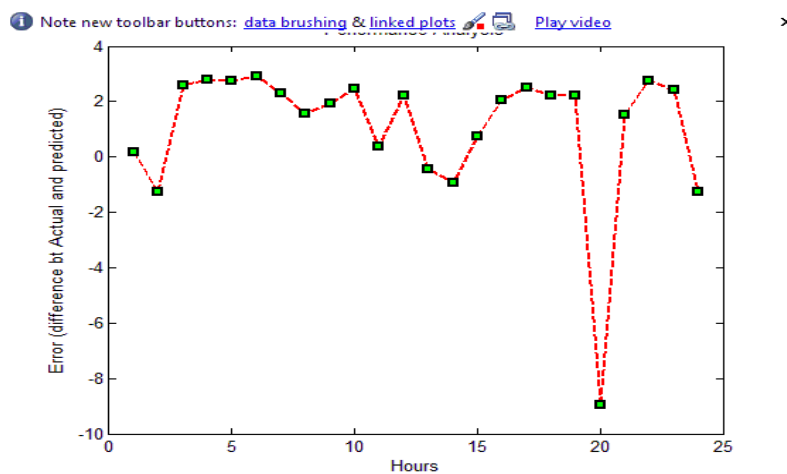


Figure: 8 Error between actual and predicted

V. CONCLUSION

The Hybrid Method that improves the accuracy of bounded variables in Linear Programming Problem model by suggesting the training and learning of parameters and constraints. The Training is possible in applications by applying Artificial Neural Network. I believe, the proposed algorithm shows increased performance in accuracy. The Hybrid method must be optimal in both computational effort and accuracy. All the above methodologies concentrated only on achieving best computational effort and reduce time whereas; accuracy of these systems may go down due to reducing number of constraints and number of iteration. To achieve optimality in computational effort and also in accuracy, I proposed a structure called Hybrid method with training model for optimization of parameters in a real time Linear Programming Problem.

REFERENCES

1. Anderson E.D. and Andersen K.D., 1995, "Pre-solving in linear programming", Mathematical Programming Series, Vol.2, No.7, pp.221-245.
2. ArvindBabu L.R., Samueeullah A.M. and Palaniappan B., 2007, "Complex Algorithm for bounded variables", ActaCienciaIndica, Vol. XXXIII M, No.2, pp.633-636.
3. Bharambhe M.T., 1996, "A New Rule for Entering a variable in the basis in Revised Simplex Method", Mathematical Programming, Vol.6, pp.278-283.
4. Chen Chuan-bo and LiTao, 2005, "A Hybrid Neural Network System for Prediction and recognition of promoter regions in Human Genome", Journal of Zheijiang University Science, 6B(5), pp.401-407.
5. ArvindBabu L.R. and Palaniappan B., 2009, "Identification of Redundant constraints in LPP Using Various methods: A Comparative Study", ACCST Research Journal, Vol. VII, No.2, pp.57-65.
6. Paulraj S., Chellappan C., and Natesan T.R., 2006, "A heuristic approach for identification of redundant constraints in linear programming models", International Journal of Computer Mathematics, Vol.83: 8, pp.675-683.
7. Rahman S. and Bhatnagar R., 1988, "An expert system based algorithm for short-term load forecast", IEEE Transaction Power System, Vol.3, No.2, pp.392-399.
8. Senjyu T., Takara H., and Fundabashi T., 2002, "One-hour-ahead load forecasting using neural network", IEEE Transaction Power System, Vol.17, No.1, pp.113-118.
9. Song K.B., Baek Y.S., Hong D.H., and Jang G.S., 2005, "Short-term load forecasting for the holidays using fuzzy linear regression method", IEEE Transaction Power System, Vol.20, No.1, pp.96-101.
10. Stojkovic N.V and Stanimirovic P.S., 2001, "Two direct methods in linear programming", European Journal of Operational Research, Vol.131, pp.417-439.
11. Tarafdar, Haque and Kashtiban, 2005, "Applications of Neural Networks in power systems: A Review", Proceedings of world Academy of Science, Engineering & Technology, Vol.6, pp.1346-1345.
12. Telgan J., 1983, "Identifying redundant constraints and implicit equalities in system of linear constraints", Management Science, Vol.29, No.10, pp.1209-1222.
13. Trudnowski D.J., 2001, "Real-time very short-term load prediction for power-system automatic generation control", IEEE Trans. Control System Technology, Vol.9, No.2, pp.254-260.
14. HasanGhasabi-Osokei, NizamMahdavi-Amiri, 2006, "An Efficient Simplified Neural Network for Solving linear and quadratic programming problems", Elsevier, Vol.9, pp.452-4