RESEARCH ARTICLE OPEN ACCESS

Implementation of Hybrid Neural Network Model for Linear Programming Problem

L.R.Arvind Babu
Assistant Professor
Department of Computer and Information Science
Annamalai University
Annamalainagar – 608002
Tamil Nadu – India

ABSTRACT

Linear programming problems arise in real-life economic situations where profits are to be maximized or costs to be minimized. The presence of redundant constraints does not alter the optimum solution(s), but may consume extra computational effort. Advantage of using neural networks to solve problems includes powerful computation and less time required. This paper shows the advantages of neural networks for solving diverse linear programming problems. 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 method, neural network, redundant constraints, optimization.*

I. INTRODUCTION

The most important techniques used in modeling 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 (5,6,12) 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 univariate 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. MATERIAL AND METHODS

Complex Algorithm

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.

Deduction of 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 case 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.

Identification of Redundant Constraints

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.

A Heuristic Method to deduct the redundant constraints

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 was 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 apriori 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 actuals. Linear programming (LP) is the process of taking various linear inequalities together and find out the "best" optimal solution.This linear programming problem is used in various real-life applications like economic problem to find out the profit to be maximized or cost to be minimized. Linear programming problem is define as, consider the objective function

Min P(x) = cTx;
Subject to: $Ax \geq b$
And x>=0 (= g j (x) ≥ b j, j=1, 2... m) Where x is the vector of variables that is to be determined, c and b are vectors of known coefficients, A is a known matrix of coefficients, and (.)T is the transpose matrix. The expression to be maximized or minimized is called the objective function (cTx in this case). The inequalities $Ax \leq b$ is the constraints which specify a convex polytype over which the objective function is to be optimized. These linear programming problems can be solved by various methods i.e. polynomial time, interior point method, simplex method etc. These linear programming problems can also be solved by using neural network. Neural network is software or hardware simulation of a biological neuron. This neural network is used to learn to recognize the patterns in the data. Once this neural network has been trained on the samples of the data, it can make predictions by detecting similar patterns in future data. The typical structure of neural network is shown in below diagram.

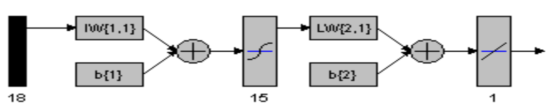

Figure.1 Shows Neural Network for Load Forecasting

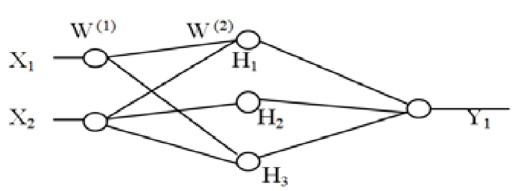

Figure.2 Shows a Typical Neural Network

The typical neural network comprises of inputs of figure (2) X1, X2, corresponding to independent variables. A hidden layer as the first layer, and second layer is the output layer whose output unit is Y1 corresponding to dependent variables. In between there is hidden layer H1, H2………corresponding to intermediate variables.
These interact by means of weight variables i.e. W(1), W(2).The activation function in neural network model is defined as

F (x) =P(x) + j+(x))2
This paper presents the linear programming problem which will be solved by using neural network. The linear programming problem can be solved by various algorithms of neural network i.e. feed forward algorithm, hybrid algorithm, Hopfield neural network etc. But this paper presents the linear programming problem which will be solved by using back propagation algorithm. Use of back propagation is rear in linear programming problem solution. It has its own advantages and disadvantages. Based on the detail literature survey and literature Gap following work is proposed. It contains defining linear programming problem and solution with neural network approach. Back propagation algorithm is used to train the network. Main Idea is to distribute the error function across the hidden layers, corresponding to their effect on the output. Process is divided into three modules
1.Implementation of neural network model.
2.Implementation of linear programming problem.
3. Solving linear programming problem by using neural network.

Step 1:Implementation of neural network model. The neural network model will be implemented by using back propagation algorithm. Back propagation algorithm is used to train the network. In first phase neural network model will be established which will be checked for working of back propagation algorithm on any simple problem. The simulation technique is used to implement the network model.
Step2: Implementation of linear programming problem. In this step the linear programming problem will be solved in MATLAB. Simple procedure will be used like Simplex Algorithm to solve the linear programming problem. It will be run for some problems to prepare test cases which will be used in training stage in final module.
Step 3: Solving linear programming problem by using neural network. The last module is designed to solve LPP by neural network. Working of the model is explained as follows.
Consider a LP problem in the following standard form:
Find x that
Maximizes: bTx
Subject to the constraints: A x<= c, x >=0 (1)
Where x and b ε Rn, A ε Rm x n and c ε Rm.
The dual problem of (1) is:
Find y that
Minimizes: c T y
Subject to the constraints: A T y >=b, y >= 0 (2)
Mathematically, the outputs of the above primal and dual neurons

can be described by the following nonlinear dynamical system:

dx /dt = b – AT (y + k (dy / dt)) , x >=0 (3)

dy/dt = -c + A (x + k (dx / dt)) , y >= 0 (4)

It can be seen that (3) and (4) are equivalent to a system of second order differential equations. The Euler method is used to solve differential equations (3) and (4) which will be solved in neural network processing.

Algorithm:

Input:

1. Pair of Linear Programming Problems with solution
2. Minimum error value

Output:

Training Stage:

1. Initialize the weights in the network (often randomly)
2. Do

For each example data (e) in the training set

i. O= neural-net-output (network, e);

ii. T = output for e

iii. Calculate error (T - O) at the output units

iv. Compute delta_wi for all weights from Hidden layer to output layer;

v. Backward pass Compute delta_ wi for all weights from input layer to hidden layer;

vi. Backward pass continued

vii. Update the weights in the network until dataset classified

III. IMPLEMENTATION OF THE PROPOSED METHOD

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.

Existing Load Forecasting Algorithm:

Step1: If the predicted day is belonging to a summer season, go to step 4. Otherwise, it is the day on spring, fall, or winter seasons.

Step2: Construct input information using the load data during three days (which are subject to Monday through Friday) before the predicted day.

Step3: Forecast the maximum load using the exponential smoothing method.

Step4: In case that the predicted day belongs to a summer season, the temperature sensitivities are computed using the variations of the load and temperature between the predicted day and its one previous day.

Step5: Forecast the maximum load with the temperature sensitivities calculated in step. After taking the above steps, the normalized value of the 24 hourly loads is calculated from the data obtained from the load during the previous three weeks of the predicted day. Thereafter, the 24 hourly loads of the day are forecasted from the normalized value.

Proposed Hybrid Algorithm:

Step 1: If the predicted day / time are belonging to class-1 environment, go to step 4 otherwise, it is class-2 environment.

Step 2: 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.

Step 3: Forecast the appropriate parameters.

Step 4: In case that the predicted day or time belongs to a class-1 situation, the value of parameters may vary.

Step 5: 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. RESULT

The Implementation of the hybrid model 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. The Hybrid Algorithm must be optimal in both computational effort and accuracy. All the above methodologies concentrated only on achieving best computational effort.

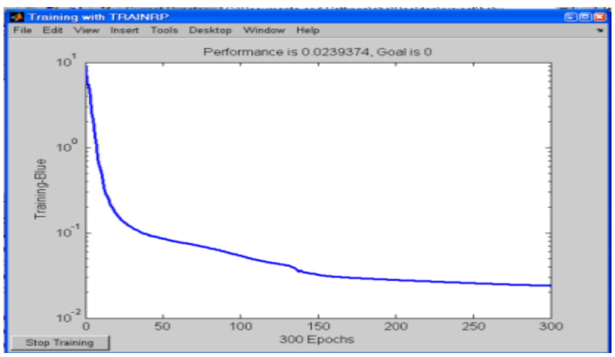

Figure: 3 Results of Back propagation

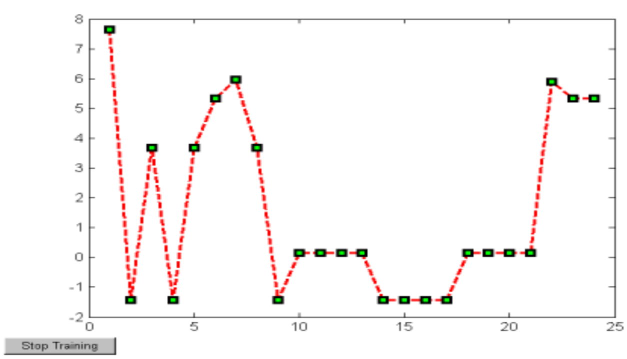

Figure: 4 Error rate in Linear Programming Problem Model

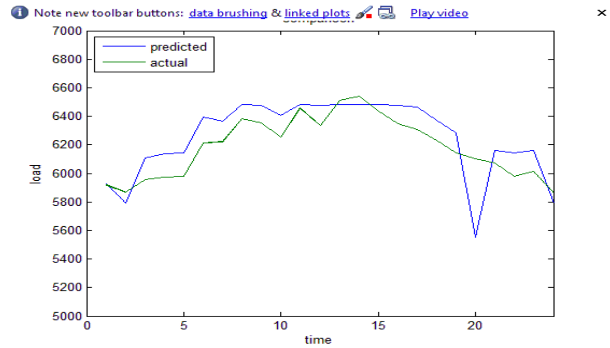

Figure: 5 Testing Phase between load vs time

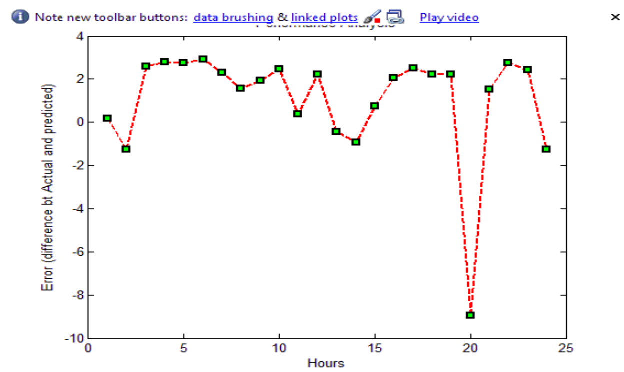

Figure: 6 Error between actual and predicted

V. CONCLUSION

The implementation of Hybrid Neural Network 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] Arvind Babu L.R., Samueeullah A.M. and Palaniappan B., 2007, “Complex Algorithm for bounded variables”, Acta Ciencia Indica, 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] Arvind Babu 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] Hasan Ghasabi-Osokei, Nizam Mahdavi-Amiri, 2006,”An Efficient Simplified Neural Network for Solving linear and quadratic programming problems”, Elsevier, Vol.9, pp.452-464.