NEURAL NETWORK BASED APPROACH FOR OPTIMIZATION

A DISSERTATION

Submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
CHEMICAL ENGINEERING
(With Specialization in Computer Aided Process Plant Design)

By

BHARI BHUJAN SINGH

UNIVERSITY OF ROORKEE CENTRAL LIBRARY
Acc. No. G10,065
Date 19-2-2000
ROORKEE

DEPARTMENT OF CHEMICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

MARCH, 2000

Gratis

CANDIDATE'S DECLARATION

I hereby certify that the work presented in this dissertation entitled **'NEURAL NETWORK BASED APPROACH FOR OPTIMIZATION'** in partial fulfillment of the requirements for the award of the degree of **MASTER OF ENGINEERING in CHEMICAL ENGINEERING with specialization in COMPUTER AIDED PROCESS PLANT DESIGN** of the University of Roorkee, Roorkee, is authentic record of my own work carried out during the period from July 1999 to March 2000 under the guidance of **Dr. Bikash Mohanty**, Associate Professor, Department of Chemical Engineering, University of Roorkee, Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other University.

Singh

Dated: 30th March, 2000

(BHARI BHUJAN SINGH)
Signature of the candidate

This is to certify that above statement made by the candidate is correct to the best of my knowledge.

Bikash Mohanty

Dated: 30·3·2000

(Bikash Mohanty)
Associate Professor,
Department of Chemical Engineering,
University of Roorkee,
Roorkee- 247 667

ABSTRACT

Optimization concerns the minimization or maximization of functions. These objective functions may be formulated from technical and/or economic viewpoint. In the optimization problem many a times independent variables should satisfy various equality and/or inequality constraints. In chemical engineering, processes are highly non-linear with constraints arising from safety considerations, process limitations etc., which leads to non-linear constrained optimization problem.

There are number of methods for solving non-linear constrained optimization problems. These methods can be divided into two broad categories: Classical methods like gradient search and direct search; and Non classical methods like genetic algorithms and ANN etc.

Although ANN is being used for optimization in other engineering fields like Electronics/Electrical etc. and to be more specific in analog VLSI technologies and electrooptics. Relatively a few studies available in optimization, which is specifically concerned with ANN based approach for optimization in Chemical engineering field.

Neural networks have now becomes focus of attention, largely because it can easily handle complex and non-linear problems which contains imprecise or noisy data. For model development, it does not require the prior knowledge of process and physics associated with it thus poorly understood systems can also be modeled with ease. The only requirement is considerable number of input/output data sets.

To study the applicability of ANN based approach for non-linear constrained optimization a number of mathematical as well as Chemical engineering problems has been selected through literature survey. In all six problems, two mathematical and four chemical engineering were chosen. MATLAB environment with its Neural network and Optimization toolbox, is used for ANN modeling and then optimization of selected problems respectively. To compare the efficiency of ANN based approach, the same problems were solved by conventional approach. Function "constr", which is nonlinear constrained optimizer of MATLAB optimization toolbox is used for optimization in conventional mathematical model based approach. For ANN based approach, first the mathematical objective function is replaced by its equivalent ANN model then this ANN model is used in a modified nonlinear constrained optimizer of MATLAB optimization toolbox named function "opti" to optimize the problem.

This study successfully demonstrated the use of ANN based approach for optimizing mathematical as well as Chemical engineering problems. ANN based approach has shown edge over conventional approach both in terms of time required and number of function evaluation needed for optimization of Chemical engineering problems. It has been noted in this present work that number of function evaluation needed to converge at optimum depends upon choice of initial guess of decision variables. Further it is also observed that ANN model takes less time in computing objective functions than that required in conventional approach. The limitation of this present work is that MATLAB optimizer may some time give local solutions, which obviously can be overcome by running the optimizer with several initial guesses of decision variables.

ACKNOWLEDGEMENTS

It is with great reverence that I express my due gratitude to my dissertation guide Dr. Bikash Mohanty, Associate Professor, Department of Chemical Engineering, University of Roorkee, Roorkee for his precious guidance and critical examination of various aspects of the study, which enabled me to complete this dissertation to the desired level. It was a matter of great pleasure and satisfaction to work under his supervision.

Thanks are due to Dr. I. M. Mishra, Professor & Head, for providing all facilities and to show keen interest in my work. I am also thankful to Dr. Surendra Kumar, Professor & ex-Head, Department of Chemical Engineering, University of Roorkee, Roorkee, for providing various facilities during this dissertation.

I owe my sincere thanks to Dr. S. C. Gupta, Professor, O/C., HTRL, Dr. V. K. Agarwal, Assistant Prof., O/C., PIL, Dr. I. D. Mall, Assistant Prof., Mr. Ravindra Bhargava, Lecturer, Department of Chemical Engineering, University of Roorkee, Roorkee, for their useful guidance and constant encouragement for completing this work.

I would like to thank Mr. Anuj Jain, Mr. P.B. Gangawati, Mr. Abdal Kareem, Mr. Yisaye Kindie Adem and Mr. Swapan Bhaumic, Research scholars, Department of Chemical Engineering, University of Roorkee, Roorkee for their support, advice and interest shown in my work.

I am thankful to Mr. Suresh Sharma and Mr. Rajkumar, members of technical staff of HTRL for their help and co-operation.

I also take this opportunity to thank all of my friends for their kind co-operation and support during the course of work.

Special and very sincere thanks are also due to my family members for their constant encouragement and moral support throughout this work.

All may not have been mentioned but none is forgotten.

(Bhari Bhujan Singh)

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE AND ABBREVIATIONS

NOMENCLATURE

b	Bias Vector
d	Search Direction
E	Squared Error
f	Objective Function
f(.)	Non-linear Function
g	Constraint
H	Hessian
P	Input Vector for Artificial Neural Network
R^2	Coefficient of Determination
T	Target Vector
W	Weight Vector/ Matrix
w_{il}	Weight for i^{th} neuron in l^{th} layer
x_i	Input Signal to the i^{th} Neuron
Y	Output Vector for Artificial Neural Network

GREEK LETTERS

α	Momentum factor
α_k	Step length parameter
η	Learning Rate
δ	Error gradient
θ	Bias Vector
λ	Lagrange Multiplier

ABBREVIATIONS

ANN	Artificial Neural Network
ARDS	Adaptive Randomly Directed Search
ARSM	Adaptive Random Search Method
BP	Back-Propagation
CM	Complex Method
GDR	Generalised Delta Rule
GP	General Problem
KT	Kuhn-Tucker
l_r	Learning Rate
MM	Method of Multipliers
NN	Neural Network
QP	Quadratic Programming
RST	Random Search Technique
SQP	Sequential Quadratic Programming
SSE	Sum Squared Error

CONTENTS

CHAPTER 1

INTRODUCTION

Optimization is an important aspect of Chemical engineering. It involves the minimization (or maximization) of an objective function, that can be established from a technical and/or economic viewpoint. In general, the decision variables are subjected to constraints such as valid range (max. and min. limits) as well as constraints related to safety considerations and those that arises from the process model equations. In engineering design and operation one is frequently confronted with the problem of static non-linear optimization. In design, one wants to find the bottlenecks and many a times redesign certain aspects of the system to maximize some function, for example, return on investment. In operation, one wants to know what flow rates, recycle, reactor temperature, catalyst activity, etc are required to maximize daily profit. Since, the relationships between many variables could be non-linear and objective function may also be non-linear, one confronts with non-linear programming problem. A general optimization problem is stated as:

Minimize or Maximize $f(x)$

Subject to constraints:

$$g_i(x) = 0 \qquad i=1\ldots m_e$$

$$g_i(x) \leq 0 \qquad i=m_{e+1}\ldots m$$

$$x_{min} \leq x \leq x_{max}$$

The non-linear constrained optimization problems, in question, can be solved by classical methods using the mathematical model of the problem. With the emergence of neural network as a reliable model building tool, researchers started exploring the possibility of replacement of mathematical model by its equivalent neural network model. A discussion regarding the classical methods and motivation for neural network based approach for tackling the above problem is included in the subsequent sections.

1.1 Classical methods

Usually in chemical engineering problems, both the objective function and the constraints are found to be non-linear. Computational methods of non-linear programming with constraints usually have to cope with problems such as numerical evaluation of derivatives (Jacobian, Hessian) and feasibility issues. On the other hand, derivative free (topology) methods, also called direct search methods, are usually less efficient and more time consuming as these usually require a higher number of iterations. The lower efficiency, of direct search methods, results from the necessity of solving the non-linear model equations in each iteration. The other difficulty faced in the optimization is the presence of multiple (local) optima and trapping of the optimizer in the local optima rather in global optima. This is usually overcome by running the program several times with different initial guesses of the decision variables.

1.2 Motivation for neural networks based approach

Neural nets have now become the focus of much attention, largely because of their wide range of applicability and the ease with which they can handle complex and non linear problems. Neural net technology is well suited to solving problems in chemical industry. Neural nets can handle problems involving data that are imprecise or "noisy" as well as those that are highly nonlinear and complex. While applying these networks fundamental understanding of processes or phenomena being modeled is not a prior condition. However, for building a efficient neural network based model one should be careful in selecting number of hidden neurons and layers, method of training and size of training data set. Selection of these parameters, many a times, based on experience. These features make neural nets excellent candidate for creating efficient model for wide range of problems in the Chemical industry. Examples include classifying mutlicomponent feed stocks, chemical composition analysis and pattern recognition, characterizing and modeling reactor processes and units, modeling poorly understood phenomena (such as turbulent flows), and so on.

Based on the neural network models non-linear constrained optimization strategies can be formulated and thus is the motivation of the present work.

1.3 Scope of this present investigation

The present investigation has been planned to carry out following objectives:

1. To develop a suitable ANN model for selected problems using feed forward architecture with back propagation training algorithm by using neural network toolbox of MATLAB.
2. To formulate strategy for the use of ANN Based model in place of mathematical model for optimization of selected problems and to apply the above developed strategy using MATLAB's Optimization toolbox.
3. To compare the efficiency of neural network model based optimizer, the same problems taken for point 1 & 2, be solved by conventional mathematical model based optimizer by using MATLAB's Optimization toolbox.

In order to test reliability of proposed neural network based method of optimization, a few problems were picked up from the literature survey. To test the robustness of proposed method, two mathematical problems were selected. To test the proposed optimization procedure with relatively realistic chemical process models, four examples extensively studied in literature were chosen.

LITERATURE REVIEW

Literature review was carried out for Neural Network Based approach for optimization. This includes the work done in the field of neural network based approach for optimization in chemical engineering field from 1988 to 1999. This literature review also embodies survey on classical non-linear optimisers and problems as these were used for comparison with neural network based approach. As in the present investigation, MATLAB environment was used for optimization using its Optimization toolbox, a discussion pertaining to this aspect is also included in the literature review.

As a first step for neural network based approach for optimization, modelling of problem is done with the help of neural network. *Himmelblau and Hoskins (1988), Bhagat (1990), Savkovic-Stevanovic (1993/1994), Huggat (1999), Nascimento and Giudici(1998), Bhat and Mc Avoy (1990), Chen and Weigand (1994), Thompson and Kramer (1996), Van Can et al. (1996)* have illustrated that ANNs appears to be suitable for modelling complex non-linear processes. Most of the time, multilayer feed forward neural network are used to approximate complex functions. A robust learning heuristic for multilayered feed forward ANN, called back propagation was successfully used by *Himmelblau and Hoskins (1988), Bhagat (1990), Savkovic-Stevanovic (1993/1994), Nascimento and Giudici(1998)*. The detailed overview of neural network is given in Appendix A. Literature review for neural network based approach for optimization is discussed below.

2.1 NEURAL NETWORK BASED OPTIMIZATION

Optimization is becoming an important feature of all fields, including the Chemical engineering field. Relatively few studies available in optimization are specifically concerned with the neural network based approach for optimisation in Chemical Engineering field.

Himmelblau and Hoskins (1988) published first paper that relates Neural Networks and Chemical Engineering. In this paper, they demonstrated via a simple

Chemical engineering process (Three continuous stirred tank reactors in series), how an artificial neural network could learn and discriminate successfully among faults.

Bhagat (1990) discussed the application of neural networks by taking two examples, the first involves two CSTR's in series and second example involves identifying the degree of mixing in a reactor or vessel. He also discussed about the potential application of neural networks in Chemical Engineering.

Savkovic-Stevanovic (1993) published first paper in which he discussed about neural network model for analysis and optimisation of process. ANNs based on a feed forward architecture and trained by back propagation technique were applied to analysis and improvement of a separation.

Savkovic-Stevanovic (1994) published paper in which author discussed about the neural networks' application for process analysis and optimisation. The author had undertaken two case studies first case study consists of a distillation column and a liquid-liquid separator (heterogeneous azeotrope). Second case study he considered, the process of azeotropic distillation of butylacetate-butylalcohol-water system in interlinked industrial distillation column. For the first case five input neurodes and three output neurodes and for second case, study eight input neurodes and two output neurodes were taken. Various numbers of hidden units and training method were explored and compared. The training results obtained by GDR algorithm are compared with those obtained by the Powell method for minimizing the sum of squares of errors. GDR algorithm gives less training errors, than Powell method for same number hidden units, results obtained by him illustrate the feasibility of using neural network as a data analyser and as an optimisation tool.

Nascimento and Giudici(1998) presented a very good paper on neural network based optimisation, they used NN for modelling the system and then use this trained NN to carry out a grid search, mapping all the regions of interest. They applied this in a twin-screw extrude reactor. This corresponds to the finishing stage of an industrial polymerisation plant. A qualitative optimisation procedure is used, taking in account safe operation conditions, wear and tear of the equipment, product quality and energy consumption. The chosen operation variables are then checked with the phenomenological model. This approach provides more comprehensive information for the engineer's analysis than the conventional non-linear programming procedure.

Hugget et al (1999) recently published a paper in which a novel and global strategy involving ANN and a genetic algorithm is presented and validated for an industrial convective drier. To begin with, a method to represent a drying model using ANN is defined. This method is tested and the results are compared with those obtained with classic numerical methods. This approach drastically reduces times and maintains good accuracy and generalisation properties. Second, the associated optimal design problem is considered, this optimisation appears as a complex combinatorial problem with 42 million potential solutions and a multiobjective function that is not continuous, not differentiable, and not explicit. A stochastic method, such as genetic algorithm appeared well suited to this kind of problem. Find results illustrate the efficiency of this global approach.

2.2 CLASSICAL METHODS OF OPTIMIZATION

There is growing number of optimization methods, mainly proposed by applied mathematics. These methods can be conveniently divided into two broad categories: direct search methods and gradient search methods; as the name imply, the latter category uses derivatives of various functions involved in the optimization problem whereas the former category does not. It is possible to group direct search methods into two classes depending on the methodology: pattern search methods and random search methods. Direct search methods are simple to understand and use, but gradient search methods, being based on sound mathematical principles, are often computationally efficient. However, unavailability of analytical derivatives and discontinuous functions may, in some cases, make direct search methods more attractive than gradient search methods. (*Nascimento and Giudici(1998)*

2.2.1 Problems identified for optimisation (solved by classical methods)

Extensive literature survey was done to select the problems for optimisation. Because this is Chemical engineering thesis, more emphasis was given to chemical engineering related problems but few mathematical problems have also been selected to test the robustness of proposed method. Selected problems were solved by several investigators by different optimisation techniques. These problems were used to test and compare the different optimisation methods by previous investigators.

Two mathematical problems, selected here, had been used to test the reliability of algorithms. One mathematics problem has 18 local maxima, this problem is used to test whether proposed method can identify all local maxima and gives global maxima.

In order to test the optimisation procedure with relatively realistic chemical process models, four examples extensively studied in literature were chosen. Fuel allocation problem is to find minimum fuel cost to produce desired power. The objective function is non-linear with four independent variables, one equality constraint, and a non-linear inequality constraint. Drying process problem has non-linear objective function and is optimised under the non-linear inequality constraint functions. In this problem, we have to find the airflow rate and bed thickness, which will maximise the production rate. This non-linear example has only two independent variables but three inequality constraints. Two Alkylation process problems are chosen; first one has ten independent variables, seven of which can be eliminated by equality constraints. Bracken and McCormick (1968) formulated a different optimisation problem for the alkylation process by assuming that the four equality constraints arising from the regression analysis of experimental data need not be satisfied exactly, it is sufficient if values of these equality constraints lie between two limits. So in the new problem, four equality constraints are transformed into eight inequality constraints.

2.2.1.1 Mathematical problems

2.2.1.1.1 Four Variable problem

Rosen & Suzuki (1965) originally proposed this problem as a test for non-linear programming algorithms. *Gould (1971)* solved this by applying a modified sequential unconstrained technique to achieve the optimum at – 44.03. *Luus and Jaakola (1973)* and *Heuckroth and Gaddy (1976)* also solved the problem with random search procedure and gives optimum at – 44.0 and – 43.999 respectively. *Martin and Gaddy (1982)* solved this problem by Adaptive Randomly Directed search procedure (ARDS), gives optimum value as – 44.0. *Gade Pandu Rangaiah (1985)* solved this problem by complex method and compared the results with ARSM method, reported optimum at – 44.0.

2.2.1.1.2 Hesse's function

Hesse (1973) originally presented this problem. *Luus and Wang (1978)* optimised the function using new random search technique. *Martin and Gaddy (1982),* by using ARDS algorithm, solved this problem. *R. Salcedo, et al, (1990)* used SGA algorithm to solve this problem and compared their results with other methods. *V. Visweswaran, et al, (1990)* solved this concave function by using Global optimisation algorithm. All these investigators reported optimum at -310.

2.2.1.2 Chemical Engineering Problems

2.2.1.2.1 Fuel allocation in power plants

Hovanessian and stout (1963) solved this problem by linearisation and the simplex method and reported optimum value of function at 3.17. Luus and Jaakola (1973), and Heuckroth and Gaddy (1976) also optimised the process using random search techniques. G. P. Rangaiah (1985) used MM, CM and ARSM for solving this problem. R. Salcedo, *et al*, (1990) solved this problem by SGA algorithm. All these investigators reported optimum value at 3.05.

2.2.1.2.2 Drying process problem

This problem involves the maximization of drying rate for a through-circulation dryer. *Chung (1972)* who proposed this problem, solved the system analytically using a differential algorithm. He gives optima at 172.5. *Luus and Jaakola (1973)* and *Heuckroth and Gaddy (1976)* also optimised the process using random search techniques and reported optimum at 172.49. *Martin and Gaddy (1982)* used ARDS to solve this problem, gives optima at 172.5. *R. Salcedo, et al, (1990)* solved this problem by SGA algorithm, gives optima at 172.49. *G. P. Rangaiah (1985)* used MM, CM and ARSM for solving this problem and gives optima at 172.487.

2.2.1.2.3 Alkylation process problem (a)

This problem, an alkylation process, was proposed by *Payne (1958)* and optimised by *Sauer, et al , (1964)* utilizing a linear programming technique with automatic approximation of the non-linear model with small linear segments.

Keefer (1973) also solved this problem with simpat method. *Luus and Jaakola (1973)* and *Heuckroth and Gaddy (1976). Martin and Gaddy (1982)* solved this problem by ARDS algorithm. *R. Salcedo, et al, (1990)* solved this problem by SGA algorithm. *G. P. Rangaiah (1985)* solved this problem by CM and ARSM. All these investigators reported a minimum value of -1162.

2.2.1.2.4 Alkylation process problem (b)

Bracken and McCormick formulated a different optimisation problem for the alkylation process. *Berna, et al, (1980)* studied this problem by using a Newton-Rapson type method and quadratic programming, reported optimum at –1.765. *Westerberg & Debrosse (1973)* also studied this problem using an optimising algorithm for structured design systems shows –1.715 as optimum value. *Carlos Vinante, et al, (1985)* solved this problem by application of the method of multipliers and given –1.767 as optimum value. *G. P. Rangaiah (1985)* also solved this problem by method of multipliers shows this problem has optimum value of -1.769.

2.3 MATLAB OPTIMISER

For optimising the chosen problems, MATLAB optimisation Toolbox is used. Function "constr", which is a MATLAB optimiser, used for optimisation with the help of mathematical model equations of problems and function "opti", which is modified version of function "constr", used for optimisation with model equations, are replaced by equivalent neural network model. The general approach of MATLAB optimiser is discussed below.

2.3.1 Constrained Optimization

In constrained optimization, the general aim is to transform the problem into an easier sub problem that can then be solved and used as the basis of an iterative process. A characteristic of a large class of early methods, is the translation of the constrained problem to a basic unconstrained problem by using a penalty function for constraints, which are near or beyond the constraint boundary. In this way, the constrained problem is solved using a sequence of parameterized unconstrained optimizations, which in the limit (of the sequence) converge to the constrained problem. These methods are now considered relatively inefficient and have been replaced by methods that have focused on the solution of the Kuhn-Tucker (KT) equations. The KT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, that is, f (x) and, g_i(x), i = 1,.....,m, are convex functions, then the KT equations are both necessary and sufficient for a global solution point.

A General Problem (GP) description is stated as

GP

$$\underset{x \in \Re^n}{\text{minimize}} \ f(x)$$

$$\text{subject to:} \quad g_i(x) = 0, \quad i = 1,\ldots\ldots,m_e$$

$$g_i(x) \leq 0, \quad i = m_e+1,\ldots\ldots,m$$

$$x_l \leq x \leq x_u \tag{2.1}$$

Referring to GP (Eq. 2.1), the Kuhn-Tucker equations can be stated as

KT

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0$$

$$\lambda_i^* g_i(x^*) = 0 \quad i = 1,\ldots\ldots,m_e$$

$$\lambda_i^* \geq 0 \quad i = m_e+ 1, \ldots\ldots,m \tag{2.2}$$

The first equation describes a canceling of the gradients between the objective function and the active constraints at the solution point. For the gradients to be canceled, Lagrange Multipliers (λ_i,i = 1,...,m) are necessary to balance the deviations in magnitude of the objective function and constraint gradients. Since only active constraints are included in this canceling operation, constraints that are not active, must not be included in this operation and so are given Lagrange multipliers equal to zero. This is stated implicitly in the last two equations of Eq. 2.2.

The solution of the KT equations forms the basis to many nonlinear programming algorithms. These algorithms attempt to compute directly the Lagrange multipliers. Constrained quasi-Newton methods guarantee superlinear convergence by accumulating second order information regarding the KT equations using a quasi-Newton updating procedure. These methods are commonly referred to as Sequential Quadratic Programming (SQP) methods since a QP sub-problem is solved at each major iteration (also known as Iterative Quadratic Programming, Recursive Quadratic Programming, and Constrained Variable Metric methods).

2.3.1.1 Sequential Quadratic Programming (SQP)

SQP methods represent state-of-the-art in nonlinear programming methods. *Schittowski* (1985), for example, had implemented and tested a version that out performs every other tested method in terms of efficiency, accuracy, and percentage of successful solutions, over a large number of test problems.

Based on the work of *Biggs* (1975), *Han* (1977), and *Powell* (1978), the method allows one to closely mimic Newton's method for constrained optimization just as is done for unconstrained optimization. At each major iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a QP sub-problem whose solution is used to form a search direction for a line search procedure. An overview of SQP is found in *Fletcher* (1980), *Gill et al.* (1981), *Powell* (1983), and *Schittowski* (1983). The general method, however, is stated here.

Given the problem description in GP (Eq. 2.1), the principal idea is the formulation of a QP sub-problem based on a quadratic approximation of the Lagrangian function.

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \qquad (2.3)$$

Here Eq. 2.1 is simplified by assuming that bound constraints have been expressed as inequality constraints. The QP sub-problem is obtained by linearizing the nonlinear constraints.

QP Sub-problem

$$\text{minimize } \frac{1}{2}d^T H_k d + \nabla f(x_k)^T d$$

$$d \in R^n$$

$$\nabla g_i(x)^T d + g_i(x) = 0 \qquad i = 1,\ldots\ldots,m_e$$

$$\nabla g_i(x)^T d + g_i(x) \leq 0 \qquad i = m_{e+1},\ldots\ldots,m \qquad (2.4)$$

This sub-problem can be solved using any QP algorithm. The solution is used to form a new iterate

$$x_{k+1} = x_k + \alpha_k d_k$$

The step length parameter α_k is determined by an appropriate line search procedure so that a sufficient decrease in a merit function is obtained. The matrix H_k is a positive definite approximation of the Hessian matrix of the Lagrangian function (Eq. 2.3). H_k can be updated by any of the quasi-Newton methods, although the BFGS method appears to be the most popular.

A nonlinearly constrained problem can often be solved in less iteration than an unconstrained problem using SQP. One of the reasons for this is that, because of limits on the feasible area, the optimizer can make well-informed decisions regarding directions of search and step length.

2.3.1.2 SQP Implementation

The MATLAB SQP implementation consists of three main stages, which are discussed briefly in the following subsections:

- Updating of the Hessian matrix of the Lagrangian function
- Quadratic programming problem solution
- Line search and merit function calculation

2.3.1.2.1 Updating the Hessian Matrix

At each major iteration, *H*, a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, is calculated using the BFGS method where λ_i (i = 1,...,m) is an estimate of the Lagrange multipliers.

Hessian Update (BFGS)

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T H_k}{s_k^T H_k s_k}$$

$$\text{where } s_k := x_{k+1} - x_k$$

$$q_k := \nabla f(x_{k+1}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x)_{k+1} - (\nabla f(x_k) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x)_k) \qquad (2.5)$$

Powell (1978) recommends keeping the Hessian positive definite even though it may be positive indefinite at the solution point. A positive definite Hessian is maintained providing $\mathbf{q}_k^T\mathbf{s}_k$ is positive at each update and that ***H*** is initialized with a positive definite matrix. When is $\mathbf{q}_k^T\mathbf{s}_k$ not positive, $\mathbf{q}_k$ is modified on an element by element basis so that $\mathbf{q}_k^T\mathbf{s}_k>0$. The general aim of this modification is to distort the elements of $\mathbf{q}_k$, which contribute to a positive definite update, as little as possible. Therefore, in the initial phase of the modification, the most negative element of $\mathbf{q}_k\mathbf{s}_k^T$ is repeatedly halved. This procedure is continued until the minimum diagonal element of $\mathbf{q}_k^T\mathbf{s}_k \geq -1e\text{-}5$ or if $\mathbf{q}_k^T\mathbf{s}_k$ becomes positive. If after this procedure, $\mathbf{q}_k^T\mathbf{s}_k$ is still not positive, $\mathbf{q}_k$ is modified by using a vector v multiplied by a constant scalar *w*, so that

$$\mathbf{q}_k = \mathbf{q}_k + w.v \qquad (2.6)$$

where

$$v_i = \begin{cases} \nabla g_i(x)_{k+1} g_i(x)_{k+1} - \nabla g_i(x)_k \, g_i(x)_k \, , if q_{ki} w_i < 0 and q_{ki} s_{ki} < 0 \\ 0 otherwise \; (i=1,.....m) \end{cases}$$

and *w* is systematically increased until $\mathbf{q}_k^T\mathbf{s}_k$ becomes positive.

Function "constr" and function "opti" both use SQP. If options(1) is set to 1, then various informations are given such as function values and the maximum constraint violation. When the Hessian has to be modified using the first phase of the procedure described above to keep it positive definite, then mod Hess is displayed. If the Hessian has to be modified again using the second phase of the approach described above, then mod Hess (2) is displayed. When the QP sub-problem is infeasible, then infeasible will be displayed. Such displays are usually not a cause for concern but indicate that the problem is highly nonlinear and that convergence may take longer than usual. Sometimes the message no update is displayed indicating that $q_k s_k$ is nearly zero. This can be an indication that the problem setup is wrong or you are trying to minimize a noncontinuous function.

2.3.1.2.2 Quadratic Programming Solution

At each major iteration of the SQP method a QP problem is solved of the form where **A**$_i$ refers to the i-th row of the $m \times n$ matrix A.

QP

$$\text{minimize } \frac{1}{2}x^T Hx + c^T x$$

$$\mathbf{x} \in \Re$$

$$\boldsymbol{A_i x = b} \qquad i = 1,\ldots,m_e$$

$$\boldsymbol{A_i x \le b} \qquad i = m_e,\ldots,m \tag{2.7}$$

The method used in the Optimization Toolbox is an active set strategy (also known as a projection method) similar to that of *Gill et al.*, described in (1991). It has been modified for both LP and QP problems.

The solution procedure involves two phases: the first phase involves the calculation of a feasible point (if one exists), the second phase involves the generation of an iterative sequence of feasible points that converge to the solution. In this method an active set is maintained, $\overline{A}_k$, which is an estimate of the active constraints (i.e., which are on the constraint boundaries) at the solution point. Virtually all QP algorithms are active set methods. This point is emphasized because there exist many different methods that are very similar in structure but that are described in widely different terms.

$\overline{A}_k$ is updated at each iteration, k, and this is used to form a basis for a search direction, **d**$_k$ Equality constraints always remain in the active set, $\overline{A}_k$. The notation for the nonitalicised variables, $\mathbf{d}_k$ and k, is used here in order to distinguish them from d_k and k in the major iterations of the SQP method. The search direction is calculated and minimizes the objective function while remaining on any active constraint boundaries. The feasible subspace for $\mathbf{d}_k$ is formed from a basis, $\mathbf{Z}_K$, whose columns are orthogonal to the estimate of the active set, $\overline{A}_k$ (i.e., $\overline{A}_k \mathbf{Z}_k\mathbf{=0}$). Thus a search direction, which is formed from a linear summation of any combination of the columns of $\mathbf{Z}_k$, is guaranteed to remain on the boundaries of the active constraints.

The matrix $\mathbf{Z}_K$ is formed from the last *m-l* columns of the QR decomposition of the matrix $\overline{A}_k$, where *l* is the number of active constraints and l < m. $\mathbf{Z}_k$, is given by

$$Q\overline{A}_k^T = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$\text{and } \mathbf{Z}_{kij} = \boldsymbol{Q}_{ij}. \quad i = 1{:}n \quad j = m\text{-}l{:}m. \qquad (2.8)$$

Having found $\mathbf{Z}_k$ a new iterate $\boldsymbol{x}_{k+1}$ is sought of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \mathbf{d}_k \qquad (2.9)$$

where $\mathbf{d}_k$ is the search direction formed from a linear combination of the columns of $\mathbf{Z}_k$, i.e., $\mathbf{d}_k = \boldsymbol{p}^T\mathbf{Z}_k$, where $\boldsymbol{p}$ is a vector of constants. The value of the objective function at the next iterate, k+1, can thus be given as

$$f(p)_{k+1} = \frac{1}{2}(x_k + pZ_k)^T H(x_k + pZ) + c^T(x_k + pZ_k) \qquad (2.10)$$

Differentiating this with respect to $\boldsymbol{p}$ yields

$$\nabla f(p)_{k+1} = Z_k^T H Z_k p + Z_k(Hx_k + c) \qquad (2.11)$$

$\nabla f(p)_{k+1}$ is referred to as the projected gradient of the objective function since it is the gradient projected in the subspace defined by $\mathbf{Z}_k$. The term, $\mathbf{Z}_k^T\ \boldsymbol{H}\mathbf{Z}_k$, is called the projected Hessian. Assuming the Hessian matrix, $\boldsymbol{H}$, is positive definite (which is the case in this implementation of SQP), then the minimum of the function, $f(\text{p})_{k+1}$, in the subspace defined by $\mathbf{Z}_k$, occurs when $\nabla f(\text{p})_{k+1} = 0$, which is the solution of the system of linear equations

$$Z_k^T HZ_k p = -Z_k(Hx_k + c) \qquad (2.12)$$

A step is then taken of the form

$$x_{k+1} = x_k + \alpha \mathbf{d}_k \quad \text{where} \quad \mathbf{d}_{k+1} = p^T Z_k \qquad (2.13)$$

At each iteration, because of the quadratic nature of the objective function, there are only two choices of step length. A step of unity along $\mathbf{d}_k$ is the exact step to the minimum of the function restricted to the null space of $\overline{A}$. If such a step can be taken, without violation of the constraints, then this is the solution to **QP** (2.7). Otherwise, the step along $\mathbf{d}_k$ to the nearest constraint is less than unity and a new constraint is included in the active set at the next iterate. The distance to the constraint boundaries in any direction, $\mathbf{d}_k$, is given by

$$\alpha_i = \frac{-(A_i x_k - b)}{A_i d_k}, i = 1, \ldots\ldots, m \qquad (2.14)$$

which is defined for constraints not in the active set, and where the direction, $\mathbf{d}_k$, is towards the constraint boundary, i.e., $A_i \mathbf{d}_k > 0$, $i = 1, \ldots\ldots, m$.

When n independent constraints are included in the active set, without location of the minimum, Lagrange multipliers, λ_k are calculated which satisfy the nonsingular set of linear equations

$$\overline{A}\,\lambda_k = Hx_k + c \qquad (2.15)$$

If all elements of λ_k are positive, $\mathbf{x}_k$ is the optimal solution of **QP**. However, if any component of λ_k is negative, and it does not correspond to an equality constraint, then the corresponding element is deleted from the active set and a new iterate is sought.

2.3.1.2.3 Line Search and Merit Function

The solution to the QP sub problem produces a vector d_k, which is used to form a new, iterate.

$$x_{k+1} = x_k + \alpha_k \mathbf{d}_k \qquad (2.16)$$

The step length parameter, α_k, is determined in order to produce a sufficient decrease in a merit function. The merit function used by Powell (1983) of the form given below

Merit Function

$$\Psi(x) = f(x) + \sum_{i=1}^{m_e} r_i g_i(x) + \sum_{i=m_e}^{m} r_i \max\{0, g_i(x) \qquad (2.17)$$

has been used in this implementation. Powell recommends setting the penalty parameter

$$r_{ki} = \max\left\{\lambda_i, \frac{1}{2}\left(r_{(k-1)i} + \lambda_i\right)\right\}, \qquad i = 1,....m \tag{2.18}$$

This allows positive contribution form constraints that are inactive in the **QP** solution but were recently active. In this implementation, initially the penalty parameter r_i is set to

$$r_i = \frac{\|\nabla f(x)\|}{\|\nabla g_i(x)\|} \qquad i = 1,......,m \tag{2.19}$$

where $\|\cdot\|$ represents the Euclidean norm.

This ensures larger contributions to the penalty parameter from constraints with smaller gradients, which would be the case for active constraints at the solution point.

2.3.2 ALGORITHM FOR MATLAB NON-LINEAR CONSTRAINED OPTIMIZATION FUNCTION

The algorithm for MATLAB non-linear constrained optimization function is given below:

Step(i) Take initial guess of independent variables **x**=[x_1, x_2,..x_n] and calculate function and constraint value at this point.

Step(ii) Set parameters like step length, maximum number of iterations, options(1)>0 for displaying of intermediate results etc.

Step(iii) Initialise Hessian with symmetric, positive definite matrix like identity matrix.

Step(iv) Evaluate function gradient(finite difference gradient)

Step(v) Make sure *H* is positive definite.

Step(vi) Check $q_k^T s_k > 0$.

Step(vii) If in step(vi), $q_k^T s_k$ is less than 0 then q_k, is modified. The most negative diagonal element of $q_k s_k^T$ is repeatedly halved until the minimum diagonal element of $q_k^T s_k \geq$-1e-5 or if $q_k^T s_k > 0$.

Step(viii) If step(vii) do not give $q_k^T s_k > 0$, than q_k is modified by using eqn. 2.6 until $q_k^T s_k$ become positive.

Step(ix) Perform BFGS update when $q_k^T s_k$ is positive using equation 2.5. When the Hessian has to be modified using step(vii) then mod Hess is displayed and If the Hessian has to be modified again using step(viii) then mod Hess(2) is displayed under "Step procedure" in intermediate results. This updated Hessian is used in QP solution.

Step(x) The solution to QP sub problem by using eqn. 2.7 to 2.15 produces a vector d_k, which is used to form a new iterate.

Step(xi) Display intermediate results:

f-count function value max{g} Step procedure

Step(xii) The step length parameter α_k is determined in order to produce a sufficient decrease in a merit function (eqn. 2.27). This merit function looks for improvement in either the constraints or the objective function.

Step length starts with 1 and then new iterate $\mathbf{x}_{k+1}=\mathbf{x}_k+\alpha_k\mathbf{d}_k$ is formed, then calculate function and constraint value and see whether sufficient decrease in merit function or not. If not then step length is modified (halved until step length <1e-4 and then change direction i.e. step length =-step length) again new iterate is formed this repeats until required decrease in merit function is achieved and number of iterations are less than maximum allowed iterations.

This gives new x and new function value.

Step(xiii) Check for the termination tolerance for x and on function value and on constraint violation, if satisfied then program is terminated, printing last results and displaying 'Optimization Terminated successfully' and printing active constraints.

Otherwise back to Step(iv).

PROBLEMS IDENTIFIED FOR OPTIMIZATION

The algorithm was tested with 6 sever functions published in literature. All 6-test functions employed for the analysis of performance were proposed by independent authors and cover a good range of difficult optimization problems. The main characteristics of these functions are listed in Table 3.1. These represent optimization problems of two classes, viz. constrained pure mathematical problems and constrained Chemical engineering problems.

Table 3.1: Characteristics of test Problems considered for optimization

No.	Short title	Type of problem	Type of optimization	Number of parameters	Reference
1	Four variable problem	MF	CMIN	4	Rosen & Suzuki (1965)
2	Hesse's function 18 local maxima: tests reliability	MF	CMAX	6	Hesse (1973)
3	Fuel allocation in power plants; local minima	CEP	CMIN	3	Luus & Jaakola (1973)
4	Drying process	CEP	CMAX	2	Chung (1972)
5(a)	Alkylation process	CEP	CMAX	3	Luus & Jaakola (1973)
5(b)	Alkylation process	CEP	CMAX	7	Bracken & McCormick (1968)

Type of problem: MF = Mathematical function
CEP = Chemical engineering problem
Type of optimization: CMAX = constrained maximization
CMIN = constrained minimisation

3.1 Mathematical Problems

3.1.1 4 variable problem

This test problem was first introduced by *Rosen and Suzuki (1965)* and was subsequently used by *Gould (1971)* and many other investigators, to test the effectiveness of their optimisers. The problem is purely mathematical one and does not have any physical interpretation but has a unique optimum. The problem is to minimise:

$$Min.y = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \tag{3.1}$$

Subject to:

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0 \tag{3.2}$$

$$x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0 \tag{3.3}$$

$$2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \le 0 \tag{3.4}$$

3.1.2 Hesse's function

This problem is taken from *Hesse (1973)*. It involves the minimisation of a concave function subjected to linear and quadratic (non-convex) constraints. This function has 18 local minima with a global minimisation of –310 at (5,1,5,0,5,10).

This problem is to minimise:

$$y = 25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2 + (x_6 - 4)^2 \tag{3.5}$$

Subject to:

$$2 \le x_1 + x_2 \le 6 \tag{3.6}$$

$$-x_1 + x_2 \le 2 \tag{3.7}$$

$$x_1 - 3x_2 \le 2 \tag{3.8}$$

$$(x_3 - 3)^2 + x_4 \ge 4 \tag{3.9}$$

$$(x_5 - 3)^2 + x_6 \ge 4 \tag{3.10}$$

$$0 \le x_1 \tag{3.11}$$

$$0 \le x_2 \tag{3.12}$$

$$1 \le x_3 \le 5 \tag{3.13}$$

$$0 \le x_4 \le 6 \tag{3.14}$$

$$1 \le x_5 \le 5 \tag{3.15}$$

$$0 \le x_6 \le 10 \tag{3.16}$$

3.2 Chemical Engineering Problems

3.2.1 Optimum Fuel Allocation in Power Plants

The problem is formulated for power plants consisting of a number of boiler-turbine generator combinations each with specified efficiency-load fuel characteristics. The total electrical output of power plant is also specified. The objective is to find the operating level and fuel mixture ratio of boiler-turbine generator combination to minimise the fuel cost.

This problem is related to the minimization of purchase of fuel oil when it is desired to produce an output of 50 MW from a two-boiler-turbine generator combination which can use fuel oil or blast furnace gas (BFG) or any combination of these. The maximum BFG that is available is specified. For generator 1 *Luss and Jakola (1973)* have reported fuel requirements for fuel oil in tons per hour as under:

$$f_1=1.4609+0.15186x_1+0.00145x_1^2 \tag{3.17}$$

and for BFG in fuel units per hour

$$f_2 = 1.5742+ 0.1631x_1+0.001358x_1^2 \tag{3.18}$$

where x_1 is the output in MW of generator 1. The range of operation of the generator is

$$18 \leq x_1 \leq 30 \tag{3.19}$$

Similarly, for generator 2 the requirement for fuel oil in tons per hour is;

$$g_1= 0.8008+ 0.2031x_2+0.000916x_2^2 \tag{3.20}$$

And for BFG,

$$g_2= 0.7266+ 0.2256x_2+0.000778x_2^2 \tag{3.21}$$

where x_2 is the output in Mw of generator 2. The range of operation of the second generator is

$$14 \leq x_2 \leq 25 \tag{3.22}$$

It is assumed that only 10.0 fuel units of BFG are available each hour and that each generator may use any combination of fuel oil or BFG. It is further assumed that when a combination of fuel oil and BFG is used, the effects are additive. That is, if in generator 1 one uses fuel oil and BFG in 1:3 ratio to produce x_1 MW, then the total fuel consumption consists of 0.25 f_1 tons of fuel oil per hour and 0.75 f_2 fuel units of BFG per hour.

It is planned to produce 50MW from the two generators in such a way that the amount of fuel oil consumed is minimum. Mathematically, the formulation of the problem is as follows:

Minimize

$$C= x_3 f_1 + x_4 g_1 \qquad (3.23)$$

where f_1 and g_2 are given by Equations (1) and (4) subjected to the following constraints:

(a) Operating range for the generator 1

$$18 \leq x_1 \leq 30 \qquad (3.24)$$

(b) Requirement of 50MW of power

$$x_2 = 50 - x_1 \qquad (3.25)$$

(c) Operating range of generator 2

$$14 \leq x_1 \leq 25 \qquad (3.26)$$

(d) Fraction of fuel oil used in generator 1

$$0 \leq x_3 \leq 1 \qquad (3.27)$$

(e) Fraction of fuel oil used in generator 2

$$0 \leq x_4 \leq 1 \qquad (3.28)$$

(f) Availability of blast furnace gas (BFG)

$$BFG = (1-x_3)f_2 + (1-x_4)g_2 \leq 10.0 \qquad (3.29)$$

where f_2 and g_2 are given by Equations (2) and (3).

By using Equation (9), x_2 is eliminated and the problem is to choose the variables x_1, x_3, and x_4, so that C as given by equation (7) is minimized. There are 9 inequality constraints embodied in Equation (8) and in Equations (10) to (13). Note that there is no lower restriction on Equation (13) since computationally BFG cannot become negative.

3.2.2 Drying Problem

This problem involves the minimisation of drying rate for a through-circulation dryer. Chung(1972) proposed the mathematical model of a through-circulation dryer. This model approximates the drying time of constant rate drying and falling rate drying processes. The drying production rate, in terms of the independent operating variables, is a non-linear objective function, and is optimised under the non-linear inequality constraint functions by different algorithms (details given in Chapter 2 section 2.3.1.2.2).

The optimization problem of a drying process for a through-circulation dryer is to find the air flow rate and the bed thickness which will maximise the production rate under certain constraints for a given material of known particle characteristics drying at a given temperature. An increase in airflow rate increases the drying production rate at the expense of increased pressure drop and power consumption. An increase in the bed thickness also increases the pressure drop and the power consumption. Though an increase in the bed thickness decreases the dying rate, net result may be an increase in the production rate (per pound of dry solid). The limitation of the pressure drop depends on the construction of dryer. High pressure drop causes serious air leakage and blowing of particles out through the feed end and/or the delivery end of a continuous through-circulation dryer. The power consumption is limited by the maximum horsepower of the motor of the circulating fans used.

The objective function, production rate Y, in mass unit of dry solid per unit time per unit area of bed is given by

Maximize y

$$y = \frac{0.033x_1}{H} \tag{3.30}$$

where:

$$H = 0.036/F + 0.095 - 9.27 \times 10^{-4}/E(\ln(G/F)) \tag{3.31}$$

$$G = 1 - \exp(-5.39E) \tag{3.32}$$

$$F = 1 - \exp(-107.9E) \tag{3.33}$$

$$E = x_2 / x_1^{0.41} \tag{3.34}$$

x_1 = 1st independent variable, mass flow rate, mass unit/ (area)time.

x_2 = 2nd independent variable, bed thickness, length unit.

Subject to:

Power constraint function

$$0.2 - 4.62 \times 10^{-10} x_1^{2.85} x_2 - 1.055 \times 10^{-4} x_1 \geq 0 \tag{3.35}$$

the pressure drop constraint function

$$4/12 - 8.2\times10^{-7} x_1^{1.85} x_2 - 2.25/12 \geq 0 \qquad (3.36)$$

one possible additional constraint function is the ratio of the drying time required to reach the final (average) moisture content of the bed to drying time required for the surface of the bed to reach the same final moisture content. The higher is the ratio, the thicker is the bed. Thus, to restrain the time ratio implies to restrain the bed thickness so that the variation of the final moisture content between the top and bottom layers of the bed will be within the required limit

$$2 - 109.6(E)(H) > 0 \qquad (3.37)$$

3.2.3 Alkylation Process Optimization

3.2.3.1 Problem(a)

Alkylation process is important in upgrading gasoline. The process is illustrated schematically in Figure 3.1. Fresh olefins and isobutane are added to reactor along with a large recycle stream, which contains unreacted isobutane. Fresh sulphuric acid is added to the reactor to catalyse the reaction, the waste acid is removed. The reactor effluent is sent to a fractionater, where the alkyalate product is separated from the unreacted isobutane. This problem is to determine the best operating conditions in the alkylation process described by *Payne (1958)* and is used for optimisation by *Sauer et al (1964)*.

Define:

x_1 = olefin feed, barrels/day

x_2 = isobutane recycle, barrels/day

x_3 = acid addition rate, thousand pounds/day

x_4 = alkylate yield, barrels/day

x_5 = isobutane makeup, barrels/day

x_6 = acid strength, weight %

x_7 = motor octane number

x_8 = external isobutane to olefin ratio

x_9 = acid dilution factor

x_{10}= F-4 performance number

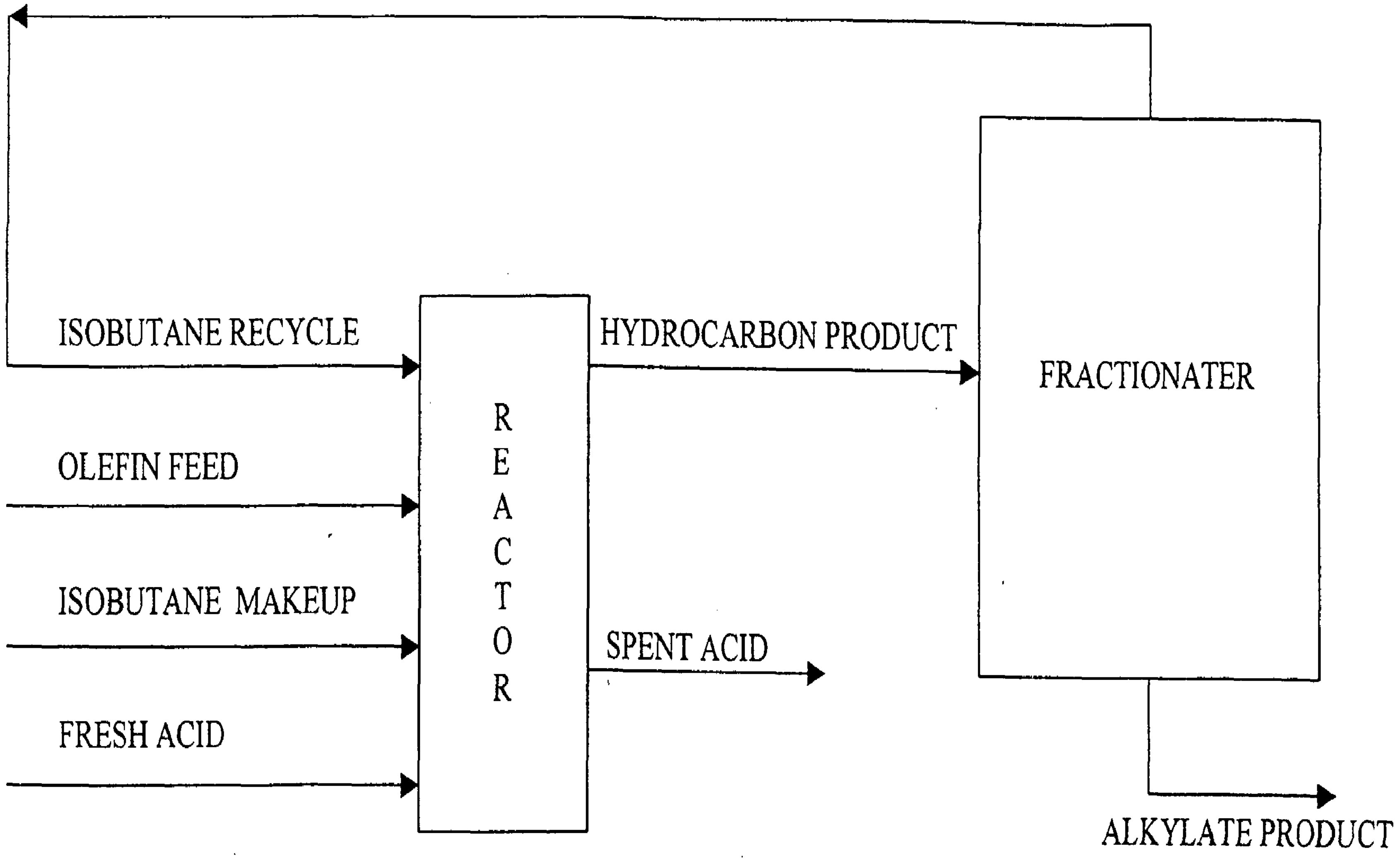


Fig. 3.1: Simplified alkylation process flow diagram

The objective of this problem is

Maximise:

$$Y = 0.063x_4x_7 - 5.04x_1 - 0.035x_2 - 10x_3 - 3.36x_5 \qquad (3.38)$$

Subject to the inequality constraints:

$$0.01 \le x_1 \le 2000 \qquad (3.39)$$

$$0.01 \le x_2 \le 16{,}000 \qquad (3.40)$$

$$0.01 \le x_3 \le 120 \qquad (3.41)$$

$$0.01 \le x_4 \le 5000 \qquad (3.42)$$

$$0.01 \le x_5 \le 2000 \qquad (3.43)$$

$$85 \le x_6 \le 93 \qquad (3.44)$$

$$90 \le x_7 \le 95 \qquad (3.45)$$

$$3 \le x_8 \le 12 \qquad (3.46)$$

$$1.2 \le x_9 \le 4 \qquad (3.47)$$

$$145 \le x_{10} \le 162 \qquad (3.48)$$

And equality constraints:

$$x_2 = x_1x_8 - x_5 \qquad (3.49)$$

$$x_3 = 0.001x_4x_6x_9/(98 - x_6) \qquad (3.50)$$

$$x_4 = x_1(1.12 + 0.13167x_8 - 0.00667x_8^2) \qquad (3.51)$$

$$x_5 = 1.22x_4 - x_1 \qquad (3.52)$$

$$x_6 = 89 + (x_7 - (86.35 + 1.098x_8 - 0.038x_8^2))/0.325 \qquad (3.53)$$

$$x_9 = 35.82 - 0.222x_{10} \qquad (3.54)$$

$$x_{10} = -133 + 3x_7 \qquad (3.55)$$

In this problem there are 7 equality constraints. Therefore, there are only 3 independent variables. The variables x_1,x_7,x_8 are chosen as independent variables. From these x_4, x_5, x_2, x_6, x_{10}, x_9 and x_3 can be calculated.

3.2.3.2 Problem(b)

Bracken and McCormick (1968) formulated a different optimization problem for the alkylation process by assuming that the four equality constraints arising from the regression analysis of experimental data need not to be satisfied exactly; it is sufficient if values of these equality constraints lie between two limits. So in new problem, four equality constraints are transformed into eight inequality constraints.

Define:

x_1 = olefin feed, barrels/day

x_2 = isobutane recycle, barrels/day

x_3 = acid addition rate, thousand pounds/day

x_4 = alkylate yield, barrels/day

x_5 = isobutane makeup, barrels/day

x_6 = acid strength, weight %

x_7 = motor octane number

x_8 = external isobutane to olefin ratio

x_9 = acid dilution factor

x_{10}= F-4 performance number

The objective of this problem is

Maximise:

$$y = -(6.3x_4x_7 - 5.04x_1 - 0.35x_2 - x_3 - 3.36x_5) \tag{3.56}$$

Subject to the inequality constraints:

$$x_1(1.22 + 1.12167x_8 - 0.0066x_8^2) - 0.99x_4 \geq 0 \tag{3.57}$$

$$-x_1(1.22 + 1.2167x_8 - 0.0066x_8^2) + x_4/0.99 \geq 0 \tag{3.58}$$

$$0.8635 + (1.098x_8 - 0.038x_8^2)/100 - 0.325(x_6 - 0.89) - 0.99x_7 \geq 0 \tag{3.59}$$

$$-0.8635 - (1.098x_8 - 0.038x_8^2)/100 + 0.325(x_6 - 0.89) + x_7/0.99 \geq 0 \tag{3.60}$$

$$35.82 - 22.2x_{10} - 0.9x_9 \geq 0 \tag{3.61}$$

$$-35.82 + 22.2x_{10} + x_9/0.99 \geq 0 \tag{3.62}$$

$$-1.33 + 3.x_7 - 0.99x_{10} \geq 0 \tag{3.63}$$

$$0 \leq x_1 \leq 2 \tag{3.64}$$

$$0 \leq x_2 \leq 1.6 \tag{3.65}$$

$$0 \leq x_3 \leq 1.2 \tag{3.66}$$

$$0 \leq x_4 \leq 5 \tag{3.67}$$

$$0 \leq x_5 \leq 2 \tag{3.68}$$

$$0.85 \leq x_6 \leq 0.93 \tag{3.69}$$

$$0.90 \leq x_7 \leq 0.95 \tag{3.70}$$

$$3 \leq x_8 \leq 12 \tag{3.71}$$

$$1.2 \leq x_9 \leq 4 \tag{3.72}$$

$$1.45 \leq x_{10} \leq 1.62 \tag{3.73}$$

And equality constraints:

$$1.22x_4 - x_1 - x_5 = 0 \tag{3.74}$$

$$0.98x_3 - x_6(x_4x_9/100 + x_3) = 0 \tag{3.75}$$

$$10.0x_2 + x_5 - x_1x_8 = 0 \tag{3.76}$$

CHAPTER 4

STRATEGY FOR OPTIMIZATION BY NEURAL NETWORK

Optimization is an important feature of all fields, including Chemical Engineering. In the past, optimization for chemical engineering processes has been done employing different methods, by many investigators. The classical optimisers use process model equations and constraints to find out optimum solution. These optimizers many a times give inaccurate results because of the assumptions and approximations taken during the development of process model equations, which invariantly leads to improper representation of process. Artificial neural networks based on a feed forward architecture and trained by the back propagation technique, can represent system behavior more accurately than its counter parts. The replacement of the phenomenological model by an equivalent NN model in the optimization step will prove to be efficient as it can take advantage of the high speed processing available with personal computers. It is a fact that simulation with a NN is much easier as it involves only a few non-iterative algebraic calculations.

The use of optimiser based on Neural Network is relatively recent. The potential of ANN's to solve several problems, which were difficult and troublesome for traditional methods, has been recognised by different authors. (*Savkovic-stevannovic (1993); Nascimento et al (1998), Hugget et al (1999)* etc).

Hugget et al (1999) *used ANN to model Dryer and applied Genetic Algorithm to optimise the problem.* Nascimento et al (1998) *replaced model equations of industrial nylon-6, 6 polymerisation process by ANN then applied grid search to find the optimal solution.* Savkovic-stevannovic (1993) *used backward simulation to find out the optimal operating conditions for two different distillation systems.*

For solving problems with the help of neural network based optimiser one should follow these three steps:

1. Formulation of the problem.
2. Development of ANN model
3. Use of optimiser on ANN model.

4.1 Formulation of problem

The optimisation of a large chemical process is a difficult problem of high dimensionality, often with both real and integer independent variables, and with numerous constraints, many of which are implicit. Due to complexity of such problems, most studies have used simplified cases. Part of processes, such as reactors, absorbers strippers and distillation column-condenser systems have been optimised using different optimisation algorithms.

For formulating the optimisation problem one has to choose, whether one wants to optimise complete plant or only part of the plant and according to that input and output variables are selected. Optimisation parameters should be chosen such that it represents all the equipment involved and operating conditions. The objective function mostly comprises of total cost, product flow, quality, operating conditions etc.

General optimization problem:

$$\text{Max. or Min. } f(x)$$

Subject to constraints:

$$g_i = 0 \qquad i = 1 \ldots m_e$$

$$g_i \leq 0 \qquad i = m_{e+1} \ldots m$$

After identification of input/ output variables next step is to create data set for training of neural network.

4.1.1 Creating data set

The first and usually longest step in development, and also generally the most critical to eventual success, is the creation of data set so that the network can learn efficiently. Data, in any case, is paramount for a neural network. How much data is enough is a complex issue, and often is affected by practical concerns such as cost of gathering data. In general, the training set must provide a representative sample of data; the network will process in the finished application. Large training sets reduces the risk of under sampling the underlying function. As in fitting a curve to a set of points, the more numerous the points, the better is the estimate. Provide too small, noisy or skewed training set, the network will learn it perfectly but will fail in final application. In practice, the sufficiency of data depends on factors such as network size, input and target distribution. The size of network matters most, a big one needs more training data than small one. Although neural network can handle data that are imprecise or 'noisy', if reconciliation of raw industrial plant data is done then with the help of reconciled plant data, the training is generally quicker and easier.

In this present study, four chemical engineering problems and two mathematical problems were solved. For these problems, model equations are used to generate data for training of the neural network.

4.2 Development of ANN model

To model the problem by ANN, a back propagation (BP) training algorithm is used (the details of this algorithm is given in Appendix A). The final development step in building ANN model is training of the network. During training, each pass through the training data is called epoch, and neural networks learns through the overall change in weights accumulating over many epochs. Training continues until the values of the weights cause the network to map input patterns to appropriate results.

Trying different configurations is a matter of changing parameters that control network structure and behaviour. For back propagation, these parameters are size of the update applied to weights, the number of hidden layers, the number of hidden nodes etc.

The values assigned to these parameters can have a large impact on the performance of the system. As there is seldom any way of deducing the best values, training requires experimentation. Training, therefore is an interactive process; the trainer tries a configuration, evaluate results, makes the necessary changes, tries it again, and repeats so until satisfied. The parameters controlling network size have a large impact on generalisation. In network using BP training, the number of hidden nodes has, particularly, a large impact. These networks, with too many hidden nodes, tend to memorise the training data; those with too few cannot answer the problem. Choosing an appropriate number is a good illustration of training through experimentation.

A large network has many weights, which represent the details of a model. These large networks can be advantageous for modelling complex functions, provided sufficient training data are available. Otherwise, excessive weights can be a drawback, since neural network can use them to memorise the training data and only reproduce the training data correctly rather than predicting the general trend correctly. A smaller network with fewer weights forces it to learn the underlying model and permits generalisation beyond training data. However, too small a network can not learn the problem.

A critical goal during training is to find a network that is a large enough to learn the application, but small enough to generalise well. The best performer is the network with fewest weight needed to process data accurately. The number of input and output nodes depends on the data and is fixed during training. As a result, choosing the number of weights is same as choosing the number of hidden nodes.

A reasonable strategy is to start with a few hidden nodes and increase the number while monitoring generalisation. The most common index of generalisation for back propagation, is mean squared error, calculated by squaring each error, summing the squares, then averaging the sum by the number of outputs and data patterns. A good technique for preventing over training is to stop when mean square error yielded stops improving much. Mean squared error fall rapidly at the beginning of training as the networks moves its weights away from their original random positions, in time curve become flatter. For the best generalisation, training should stop when mean squared error reached its lowest point and further learning offers no benefit.

4.3 Use of optimiser on ANN model

This is the final step for optimization of problem, function “opti”, which is the modified version of MATLAB optimization toolbox's constrained minimisation function “constr”, may be used for the optimization of the problem. function “opti” finds the constrained minimisation of several variables.

x = opti('fun1',x0) starts at the point x0 and finds a minimum of the function modelled by ANN. The constraints are placed in a M-file named ‘fun1.m’. The function ‘fun1’ returns a matrix of constraints, g.

[g] = fun1(x)

x = opti('fun1',x0,options) uses the parameter values in the vector options rather than the default option values. Of the 18 elements of options, the input options used by opti are: 1, 2, 3, 4, 9, 13, 14, 16, and 17. When options is an output parameter, the options used by opti to return values are: 8, 10, 11, 12, and 18.

- options (1) controls display. Setting this to a value of 1 produces a tabular display of intermediate results.
- options(2) controls the accuracy of x at the solution. (default 1e-4)
- options(3) controls the accuracy of f at the solution. (default 1e-4)
- options(4) sets the maximum constraint violation that is acceptable. (default 1e-7)
- options(8) Function value at the last evaluated point
- options(9) Gradient check (default 0)
- options(10) Function count
- options(11) Gradient count
- options(12) Constraint count
- options(13) Equality constraint (default 0)
- options(14) Maximum iterations (default 100*n, where n is number of independent variables)
- options(16) Minimum change in variables for finite difference gradient calculation (default 1e-8)
- options(17) Maximum change in variables for finite difference gradient calculation (default 0.1)
- options(18) Step size (default 1)

[x,options] = opti('fun1',x0) returns the parameters used in the optimization method.

[x,options,lambda] = opti('fun1',x0) returns the vector lambda of the Lagrange multipliers at the solution x.

[x,options,lambda,hess] = opti('fun1',x0) also returns the approximation to the Hessian at the final iteration.

Equality constraints, when present, are placed in the first elements of g. When using equality constraints, options(13) must be set to the number of equality constraints. Function "opti" uses a Sequential Quadratic Programming (SQP) method. In this method, a Quadratic Programming (QP) subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula. The QP subproblem is solved using an active set strategy. A line search is performed using a merit function.

CHAPTER 5

RESULTS AND DISCUSSIONS

In this chapter the salient results obtained from artificial neural network (ANN) modelling of selected problems and then optimization of these problems using the two different approaches, one based on ANN Model, which is basically an approximate model of the mathematical function and the other, based on the actual mathematical function are presented.

The neural network toolbox and optimisation toolbox of MATLAB software version 4.2c.1 has been used for all computational tasks. For data generation of selected problems, programs were written in FORTRAN-77. However, to carry out Neural Network modelling and optimisation of these selected problems, necessary codes were developed in MATLAB environment, the flow chart for the same is presented in Appendix C. All these programs were run using a Pentium PC 100 MHz with 8 MB RAM.

In MATLAB Optimisation toolbox, function "constr" is used for non-linear constrained optimisation. A copy of this function "constr" is modified and named "opti", so that it can be used for neural network based optimization technique used in the present investigation. It should be noted that the objective function evaluation method is different in function "constr" from that of function "opti". In function "constr" objective function of the problem is evaluated from mathematical model equations whereas in function "opti", objective function is evaluated from the ANN model, which is an approximation of the function under investigation. This is the only basic difference in these two optimization approaches otherwise the rest of the methodology is the same. In this present work, for conventional approach (that uses mathematical objective function), function "constr" and for neural network model based approach, function "opti" is used for non-linear constrained optimization.

The most common criteria used to evaluate the relative effectiveness of the optimization codes for both the approaches have been: (1) the number of functional evaluations required to obtain the optimal solution of a given test problem for a given degree of precision and/or (2) the computational time required to reach the solution of given problem. In this present work comparison between conventional approach and ANN based approach was mainly done on the above said basis.

5.1 MATHEMATICAL PROBLEMS

Two mathematical problems were selected to test the strength and weaknesses of ANN based approach for optimization.

5.1.1 Four variable problem

A 4 variable problem was selected to find out the suitability of an ANN based model to find optimum solution to above class of problem. The details of the problem are given in section 3.1.1 of Chapter 3. Many investigators [R2], [G3], [L1] etc, have solved this problem in past to test their optimisers as discussed in section 2.2.1.1.1 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.1. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the four variable problem is given below:

5.1.1.1 Development of an ANN model for 4 variable problem

To model four variable problem by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the ANN model are equal to the number of input/ output variables. There are 4 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 2 to 14 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 8 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 but found unsuitable as it causes error to increase. Therefore, learning rate of 0.002 is taken and found to be suitable for this network. Activation function used for this model is bipolar, soft continuous function

(TANSIG/ TANSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.1 and constraints 3.2 through 3.4 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.1. The training data set consists of three hundred forty four set of values of input/ output variables. The time taken for the generation of above data base of training data is 0.11 sec (input data is presented in table E.1 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.002 and 5000 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.1:

Table 5.1: SSE for various NN structure (for 4 variable problem)

	NN structure (IxHxO)*	SSE
For learning rate 0.002 and 5000 epochs	4 x 2 x 1	0.115399
	4 x 4 x 1	0.363576
	4 x 6 x 1	0.139182
	4 x 8 x 1	0.0542054
	4 x 10 x 1	0.123823
	4 x 12 x 1	0.110599
	4 x 14 x 1	0.120275

* Where I is input nodes; H is hidden nodes and O is Output nodes.

For 14 hidden neurons error first increases then decreases i.e. rippling starts after 14 hidden neurons. Thus the maximum number of hidden neuron tried are 14. From the table 5.1 it is clear that 8 hidden neurons are optimum for the present network. The weights of the network are obtained after 15000 epochs with a learning rate of 0.002. The weights and biases of the network are given in table D.1 through table D.4.

The result of training of the ANN model is shown in fig. 5.1. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes out to be 0.9969. The topology of the ANN model for a 4 variable problem is given below:

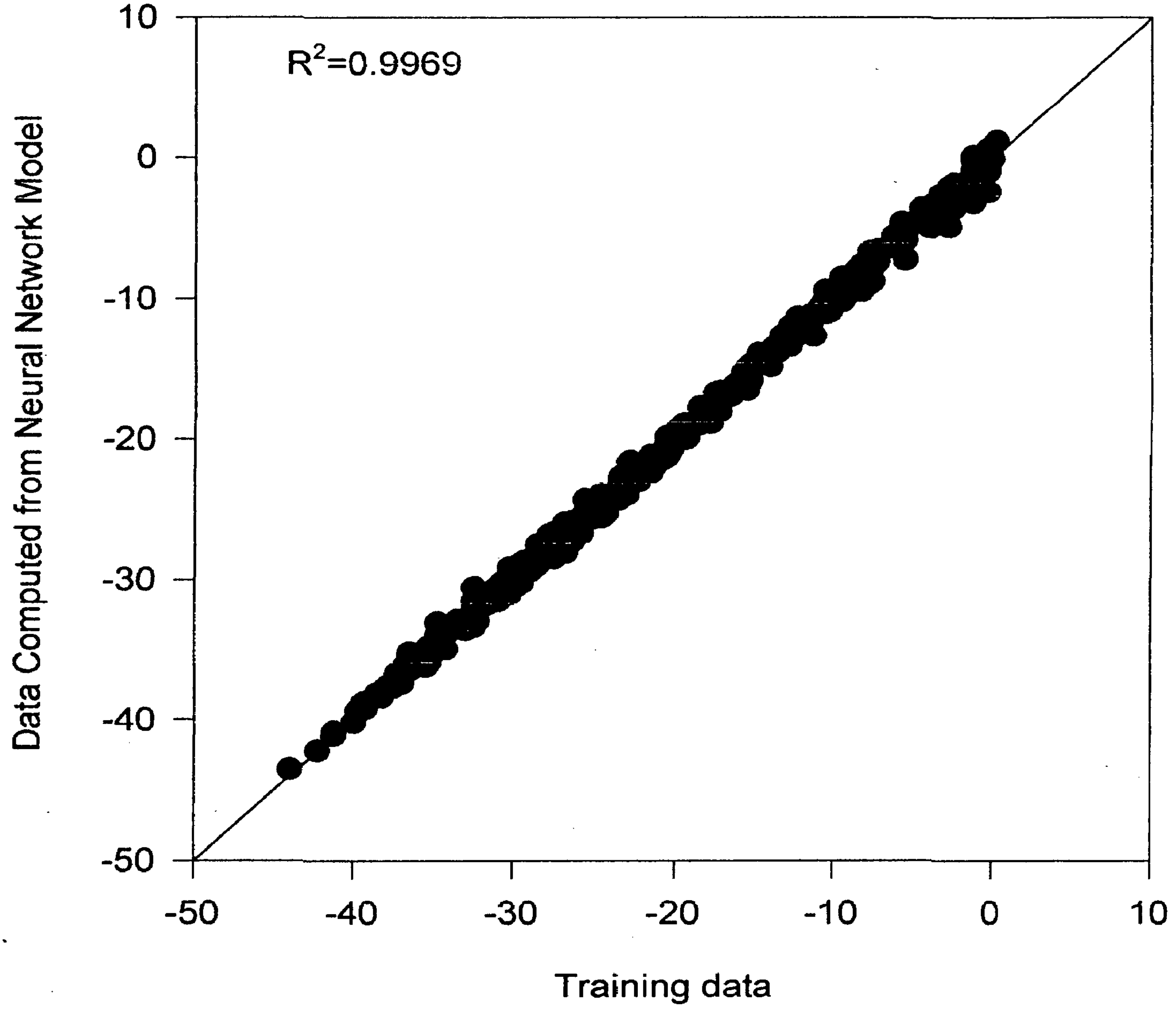


Fig. 5.1 Comparison between outputs of training data and the predicted value of output by neural network model for the 4 variable problem

Network Topology:

Input nodes	4
Output nodes	1
Hidden layer	1
Hidden nodes	8
Learning rate	0.002

After the neural network has been trained, it is able to predict the value of output corresponding to the each set of values of input variables, which may necessarily not be a part of the training data.

5.1.1.2 Optimization of 4 variable problem using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.1). When the initial guess of **x** =[0,1,2,0] is provided during the execution of program intermediate results of optimization are obtained which are shown in table 5.2.

Table 5.2: Intermediate results of optimization for initial guess x=[0,1,2,0] of the 4 variable problem by ANN based approach

f-COUNT	FUNCTION	MAX{g}	STEP Procedures
5	-37.743	-1	1
19	-40.8144	-0.949535	0.00195
30	-40.8299	-0.849712	0.0156
42	-40.8393	-0.801834	0.00781
47	-44.0501	0.746728	1
52	-44.4769	0.442839	1
57	-43.6834	0.0665958	1
62	-43.6432	0.0173814	1
67	-43.6099	0.000997241	1
72	-43.6077	1.0087e-006	1 mod Hess

73	-43.6077	5.31311e-009	1 mod Hess

After 0.82 sec of CPU time and 73 functions and constraint evaluations, the ANN based optimization approach gives optimum value of function equal to –43.6077 at x_1= -0.0284463, x_2= 0.93257, x_3= 2.03746 and x_4= -0.966902. Number of function gradient evaluation is 11 and active constraints at optimum conditions are g(1) and g(3).

5.1.1.3 Optimization of 4 variable problem by classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.1 along with the constraints eqn. 3.2-3.4 of the problem is used for optimization. MATLAB optimiser function “constr” finds the constrained minimisation of a function of several variables and is used in the present case, x=constr(‘fun’,x0) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.1). The function ‘fun’ should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that $g \leq 0$.

When the initial guess of **x** =[0,1,2,0] is provided during execution of program intermediate results of optimization obtained are shown in table 5.3.

Table 5.3: Intermediate results of optimization for initial guess x=[0,1,2,0] of the 4 variable problem by conventional approach

f-COUNT	FUNCTION	MAX{g}	STEP Procedures
5	-38	-1	1
18	-41.5615	0.528013	0.00391
31	-42.0031	0.918486	0.00391
37	-45.8861	2.83194	0.5
49	-47.4968	4.13223	0.00781
54	-51.4135	7.05316	1
59	-45.5837	0.716032	1

64	-44.0813	0.0301865	1
69	-44.0094	0.00352321	1
74	-44.0005	0.00018141	1
79	-44.0001	0.000043195	1 mod Hess
84	-44	6.84734e-006	1 mod Hess
85	-44	2.42011e-008	1 mod Hess

After 1.1 sec of CPU time and 85 functions and constraint evaluation, optimizer gives optimum value of function as –44.0 at x_1= 5.06299e-6, x_2= 0.99999, x_3=2.0, x_4=-1. Number of function gradient evaluation is 13 and active constraints at optimum conditions are g(1) and g(3).

5.1.1.4 4 variable problem optimised by conventional optimization techniques by different authors([G3],[L1],[H3],[M1],[R1]):

Gould (1971) solved this problem by applying a modified sequential unconstrained technique, work done by some other researchers are presented in a table 5.4.

Table 5.4: Results of previous investigators for 4 variable problem

	Minimum functions evaluations Required to reach 0.1 % of the Optimum	Results	Literature source
Initial guess x_i=0 for i=1,4	912	-44.03	Gould (1971)
	1759	-44.0	Luss & Jaakola (1973) by ARSM
	190	-43.999	Heuckroth & Gaddy (1976) by ARSM
	39	-44.0	Martin & Gaddy (1982) by ARDS
			Gade Pandu Rangaiah (1985)
	77	-44.0	By CM
	450	-44.0	By ARSM

From the above table it is clear that the number of function evaluations to arrive at optimum results depends on the type of conventional method selected and the minimum number of function evaluation varies from 39 to 1759.

5.1.1.5 Comparison of ANN based Optimiser with Model equation based optimiser for 4 variable problem with initial guess x=[0,0,0,0]

A comparison is carried out to check the pros and cons of the proposed Neural Network based approach vis-à-vis the conventional method for optimization. By starting with initial guess of $\mathbf{x}_i$=0, for i = 1,4. Results obtained by neural network based approach and conventional approach are presented in a table 5.5.

Table 5.5: Comparison between ANN based approach and conventional approach for initial guess x=[0,0,0,0] for 4 variable problem

Parameters	NN based approach	Conventional approach
Initial guess	x_1=0, x_2=0, x_3=0, x_4=0	x_1=0, x_2=0, x_3=0, x_4=0
Optimum points	x_1=-0.028444, x_2=0.932571, x_3=2.037461, x_4=-0.966904	x_1=-0.000005, x_2=1.000001, x_3=2.000003, x_4=-0.999996
Function Value	-43.6077	-44.0
CPU time required	1.76 sec	1.54 sec
Function evaluation	124	97
Function Gradient Evaluation	19	16

It is clear form the above table that neural network based approach was able to give results quite close to conventional approach, error in optimum value of function is only 0.89%. Although number of function evaluation needed is more than conventional approach, it is interesting to note that the time required per function evaluation for ANN model is less (for neural network, time required is 0.0142 sec per function evaluation whereas for conventional approach, it is 0.016 sec per function evaluation).

5.1.1.6 The effect of initial guess on function evaluation in a optimization process

To show the effect of initial guess on function evaluation in an optimization process, two initial guesses **x** =[0,1,2,0] and **x** =[0,0,0,0] were taken. For initial guess **x** =[0,1,2,0], results for ANN based approach and that of conventional approach are presented in table 5.2 and table 5.3 respectively. For initial guess **x** =[0,0,0,0], the results for ANN based approach as well as conventional approach is presented in table 5.5. For initial guess of **x** =[0,1,2,0], neural network based approach require only 73 function evaluation as compare to 85 function evaluation required by conventional approach whereas for initial guess of **x** =[0,0,0,0], neural network based approach needs 124 function evaluations as compared to 97 function evaluation required by conventional approach. This clearly shows that number of function evaluation depends on initial guess of input variables and this affects the effectiveness of the optimization approach.

5.1.2 Hesse's function problem

Hesse's function problem was selected because it has 18 local minima with one global minimum at –310. The details of the problem are given in section 3.1.2 of Chapter 3. Many investigators [H2], [L2], [M1], [S1] etc have solved this problem in past to test their optimiser as discussed in section 2.2.1.1.2 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.2. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the Hesse's function problem is given below:

5.1.2.1 Development of an ANN model for Hesse's function problem

To model Hesse's function problem by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the ANN model are equal to the number of input/ output variables. There are 6 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 2 to 14 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 10 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 but found unsuitable as it causes error to increase. Therefore, learning rate of 0.02 is taken and found to be suitable for this network. Activation function used for this model is unipolar, soft continuous function (LOGSIG/ LOGSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.5 and constraints 3.6 through 3.16 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.2. The training data set consists of eight hundred set of values of input/ output variables. The time taken for the generation of above data base of training data is 1.87 sec (input data is presented in table E.2 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.02 and 5000 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.6.

Table 5.6: SSE for various NN structure for Hesse's function problem

	NN structure (IxHxO)*	SSE
For learning rate 0.02 and 5000 epochs	6 x 2 x 1	1.37392
	6 x 4 x 1	0.933266
	6 x 6 x 1	1.07707
	6 x 8 x 1	0.936304
	6 x 10 x 1	0.97061
	6 x 12 x 1	1.1944
	6 x 14 x 1	2.94264

* Where I is input nodes, H is hidden nodes and O is Output nodes.

For 12 and 14 hidden neurons error first increases than decreases i.e. rippling starts after 12 hidden neurons thus maximum number of hidden neuron tried were 14. Although SSE is less for 8 hidden neurons from 10 hidden neurons for 5000 epochs, after 15000 epochs 10 hidden neuron structure give less SSE than 8 hidden neuron and is, therefore selected to model the problem. The weights of the network are obtained after 15000 epochs with a learning rate of 0.02. The weights and biases of the network are given in table D.5 through D.8.

The result of training of the ANN model is shown in fig. 5.2. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes

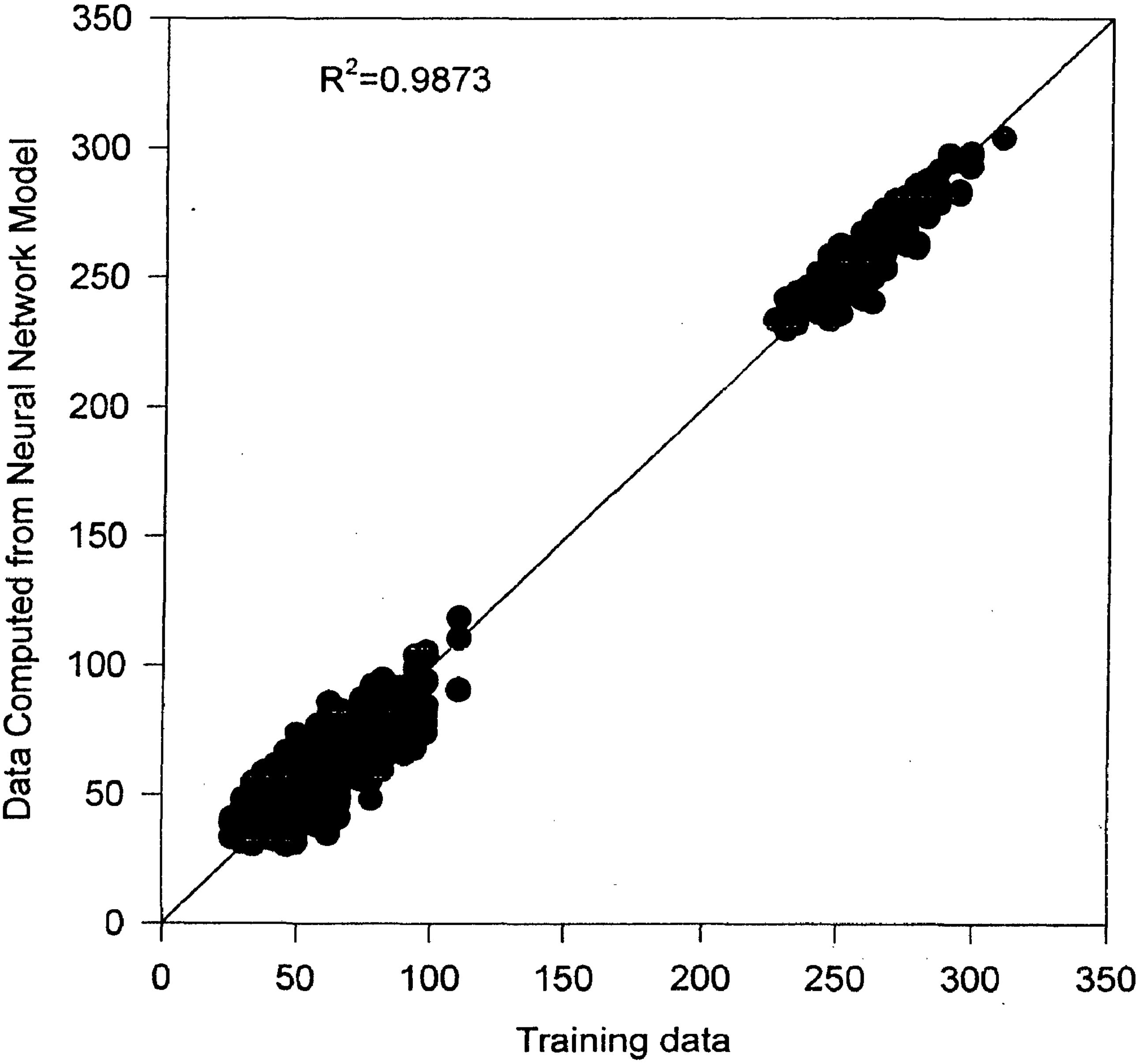


Fig. 5.2 Comparison between outputs of training data and the predicted value of output by neural network model for the Hesse's function problem

out to be 0.9873. The topology of the ANN model for a Hesse's function problem is given below:

Network Topology:

Input nodes	6
Output nodes	1
Hidden layer	1
Hidden nodes	10
Learning rate	0.02

After the neural network has been trained, it is able to predict the value of output corresponding to the each set of values of input variables, which may necessarily not be a part of the training data.

5.1.2.2 Optimization of Hesse's function problem using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.2). When the initial guess of **x** =[5,5,5,0,5,10] is provided during the execution of program intermediate results of optimization are obtained which are shown in table 5.7.

Table 5.7: Intermediate results of optimization for initial guess *x=[5,5,5,0,5,10] of Hesse's function problem by ANN based approach*

Initial Guess [5,5,5,0,5,10]	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
	7	-177.784	4	1
	14	-303.838	4.67444e-008	1 mod Hess(2)
	15	-303.838	0	1 mod Hess(2)

After 0.55 sec of CPU time and 15 functions and constraint evaluation, the ANN based optimization approach gives optimum value of function -303.838 at x_1= 5, x_2= 1, x_3= 5, x_4= 5, x_5= 0 and x_6= 10. Number of function Gradient Evaluation is 3 and active constraints at optimum conditions are g(1), g(4), g(9), g(11), g(14), g(16).

5.1.2.3 Optimization of Hesse's function by classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.5 along with the constraints eqn. 3.6-3.16 of the problem is used for optimization. MATLAB optimiser function "constr" finds the constrained minimisation of a function of several variables and is used in the present case, x=constr('fun',x0) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.2). The function 'fun' should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that g $\leq$ 0.

When the initial guess of **x** =[5,5,5,0,5,10] is provided during execution of program intermediate results of optimization obtained are shown in table 5.8.

Table 5.8: Intermediate results of optimization for initial guess x=[5,5,5,0,5,10] of Hesse's function problem by conventional approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [5,5,5,0,5,10]	7	-318	4	1
	14	-310	4.28201e-007	1 mod Hess(2)
	15	-310	0	1 mod Hess(2)

After 1.1 sec of CPU time and 15 functions and constraint evaluation, optimizer gives optimum value of function as –310.0 at x_1= 5, x_2= 1, x_3= 5, x_4= 5, x_5= 0 and x_6= 10. Number of function Gradient Evaluation is 3 and active constraints at optimum conditions are g(1), g(4), g(9), g(11), g(14), g(16).

5.1.2.4 Hesse's function problem optimised by conventional optimization techniques by different authors([L2],[M1])

Work done by some other researchers are presented in a table 5.9.

Table 5.9: Results of previous investigators for Hesse's function problem

Minimum functions evaluations Required to reach 0.1 % of the Optimum	Results	Literature source

1674	-310	Luss and Wang (1978) by RST with the method of Luss and Jaakola(1973)
1201	-310	Heuckroth, et al (1976)
1787	-310	Luss and Wang (1978)
407	-310	Martin & Gaddy (1982) By ARDS

From the above table it is clear that the number of function evaluations to arrive at optimum results depends on the type of conventional method selected and the minimum number of function evaluation varies from 407 to 1674.

5.1.2.5 Comparison of ANN based Optimiser with Model equation based optimiser

A comparison is carried out to check the pros and cons of the proposed Neural Network based approach vis-à-vis the conventional method for optimization. By starting with initial guess of **x**=[5,5,5,0,5,10] Results obtained by neural network based approach and conventional approach are presented in a table 5.10.

Table 5.10: Comparison between ANN based approach and conventional approach for initial guess x=[5,5,5,0,5,10]

Parameters	NN based approach	Conventional approach
Initial guess	x=[5,5,5,0,5,10]	x=[5,5,5,0,5,10]
Optimum points	x_1=5, x_2=1, x_3=5, x_4=0, x_5=5, x_6=10	x_1=5, x_2=1, x_3=5, x_4=0, x_5=5, x_6=10
Function Value	-303.838	-310
CPU time required	0.55 sec	1.1 sec
Function evaluation	15	15
Function Gradient Evaluation	3	3

G10,065.

It is clear form the above table that neural network based approach was able to give results quite close to conventional approach. Value of variables is exactly same as given by conventional approach while error in function is only 1.988%. Though number of function evaluation and function gradient evaluation are same in both approaches and time required by ANN based approach is half of the time required by conventional approach due to the fact that the function evaluation based on ANN model takes less time. This is evident by above discussion that when neural network models the system quite well it gives optimum solution quicker than conventional approach. Minimum number of function evaluation needed to reach at optimum solution is much higher than function evaluation need in present investigation, this shows that both of approaches i.e. conventional approach as well as ANN based approach have edge over the approaches used by previous investigators [L2],[M1].

5.2 CHEMICAL ENGINEERING PROBLEMS

In order to test the optimization procedure with relatively realistic Chemical process models, four examples extensively studied in literature were chosen.

5.2.1 Fuel allocation in power plants

The objective of this problem is to minimise the cost of fuel for a power plant. The objective function is non-linear, with four independent variables, one equality constraint, and a non-linear inequality constraint. The details of the problem are given in section 3.2.1 of Chapter 3. Many investigators [H5], [L1], [H3] etc, have solved this problem in past to test their optimisers as discussed in section 2.2.1.2.1 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.3. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the Fuel allocation in power plants problem is given below:

5.2.1.1 Development of an ANN model for Fuel allocation in power plants problem

To model Fuel allocation in power plants problem by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the ANN model are equal to the number of input/ output variables. There are 3 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 2 to 10 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 6 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 and found suitable for this problem. Activation function used for this model is unipolar, soft continuous function (LOGSIG/ LOGSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.23 and constraints 3.24 through 3.29 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.3. The training data set consists of one hundred and eighty set of values of input/ output variables. The time taken for the generation of above data base of training data is 0.11 sec (input data is presented in table E.3 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.1 and 500 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.11.

Table 5.11: SSE for various NN structure for Fuel allocation in power plants

	NN structure (IxHxO)*	SSE
For learning rate 0.1 and 500 epochs	3 x 2 x 1	0.521637
	3 x 4 x 1	0.228555
	3 x 6 x 1	0.0870553
	3 x 8 x 1	0.100717
	3 x 10 x 1	0.559419

* Where I is input nodes, H is hidden nodes and O is Output nodes.

For 10 hidden neurons error first increases than decreases i.e. rippling starts after 8 hidden neurons thus maximum number of hidden neuron tried were

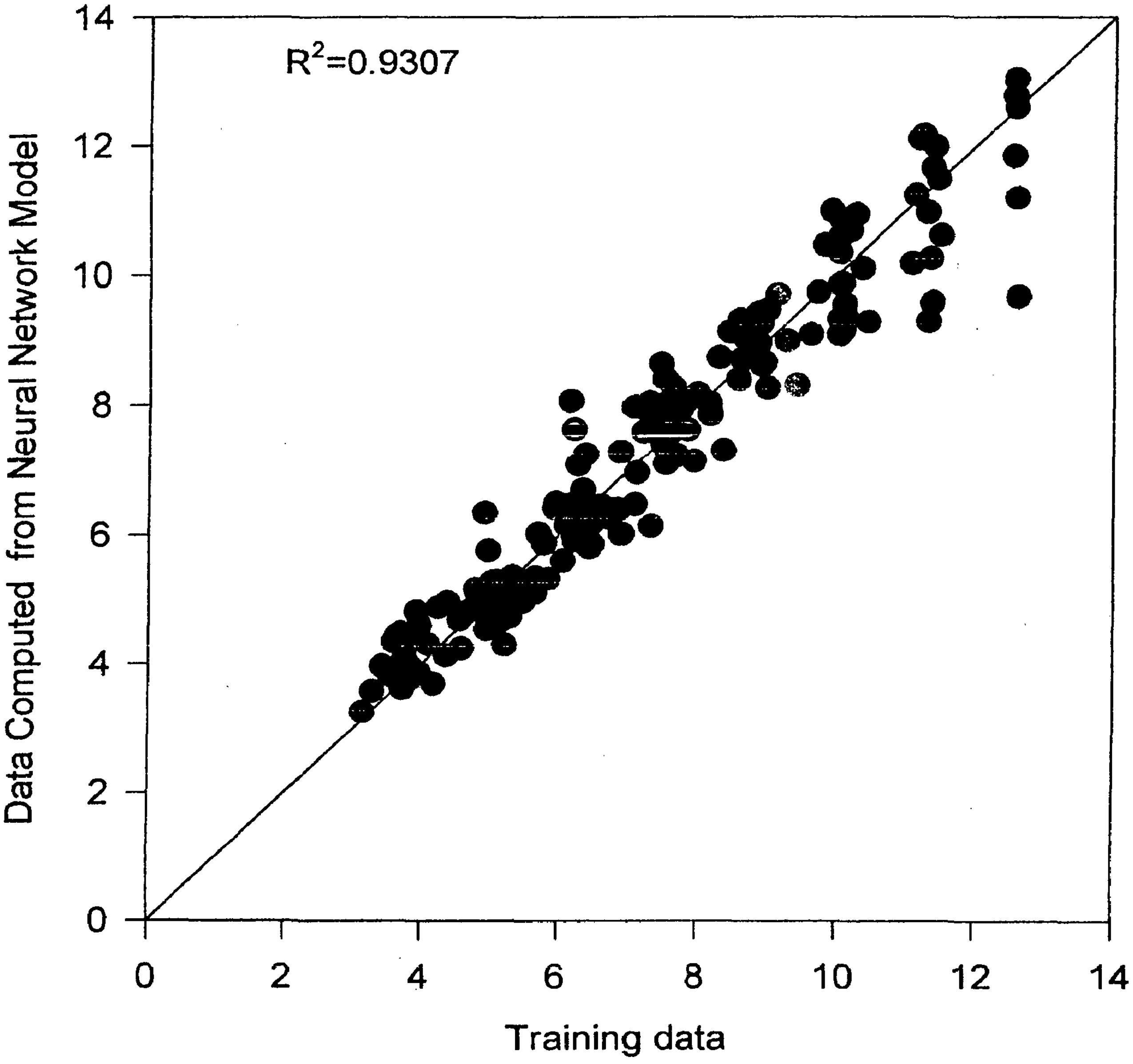


Fig.5.3 Comparison between outputs of training data and the predicted value of output by neural network model for the Optimum fuel allocation problem

10. From the table 5.11 it is clear that 6 hidden neurons are optimum for the present network. The weights of the network are obtained after 1000 epochs with a learning rate of 0.1. The weights and biases of the network are given in table D.9 through table D.12.

The result of training of the ANN model is shown in fig. 5.3. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes out to be 0.9307. The topology of the ANN model for the fuel allocation in power plants problem is given below:

Network Topology:

Input nodes	3
Output nodes	1
Hidden layer	1
Hidden nodes	6
Learning rate	0.1

After the neural network has been trained, it is able to predict the value of output corresponding to the each set of values of input variables, which may necessarily not be a part of the training data.

5.2.1.2 Optimization of Fuel allocation in power plants problem using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.3). When the initial guess of **x** =[20,0.5,0.5] is provided during the execution of program intermediate results of optimization are obtained which are shown in table 5.12.

Table 5.12: Intermediate results of optimization for initial guess x=[20,0.5,0.5] of the fuel allocation in power plants by ANN based approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [20,0.5,0.5]	4	8.95859	5	1
	8	3.02494	1.00439	1 mod Hess(2)
	12	3.22768	2.45348e-019	1
	13	3.22768	4.90695e-019	1

After 0.44 sec of CPU time and 13 functions and constraint evaluation, the ANN based optimization approach gives optimum value of function as 3.22768 tons/hr at x_1= 30, x_2= 20.0, x_3=0.0 and x_4= 0.583661. Number of function gradient evaluation is 4 and active constraints at optimum conditions are g(1), g(6), g(9).

5.2.1.3 Optimization of fuel allocation in power plants problem by Classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.23 along with the constraints eqn. 3.24-3.29 of the problem is used for optimization. MATLAB optimiser function "constr" finds the constrained minimisation of a function of several variables and is used in the present case, x=constr('fun',x0) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.3). The function 'fun' should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that $g \leq 0$.

When the initial guess of **x** =[20,0.5,0.5] is provided during execution of program intermediate results of optimization obtained are shown in table 5.13.

Table 5.13: Intermediate results of optimization for initial guess x=[20,0.5,0.5] of the fuel allocation in power plants problem by conventional approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [20,0.5,0.5]	4	6.39815	5	1
	8	3.29276	4.13702e-007	1
	12	3.00217	0.0880406	1
	16	3.04419	0.0259389	1
	20	3.06628	2.36572e-009	1 mod Hess
	24	3.05208	0	1 mod Hess(2)
	25	3.05208	0	1 mod Hess(2)

After 0.82 sec of CPU time and 25 functions and constraint evaluation, optimizer gives optimum value of function 3.05208 tons/hr at x_1= 30.0, x_2= 20.0, x_3= 0.0 and x_4= 0.583661. Number of function gradient evaluation is 7 and active constraints at optimum conditions are g(1), g(6), g(9).

5.2.1.4 Fuel allocation in power plants problem optimised by conventional optimization techniques by different authors ([H5], [L1], [H3],[R1])

The minimum fuel oil consumption is 3.22768 tons/hr, which is very close to the answer, 3.17 tons/hr obtained by Hovanessian and stout (1963) by means of separable programming where nonlinearities were approximated by linear sections and the program was solved by the standard linear programming procedure. Work done by some other researchers are presented in a table 5.14.

Table 5.14: Results of previous investigators for the fuel allocation in power plants problem for initial guess x=[20,0.5,0.5]

Minimum functions evaluations Required to reach 0.1 % of the Optimum	Results	Literature source
1952	3.05	Luss & Jaakola (1973) by ARSM

		Heuckroth & Gaddy (1976) by ARSM
255	3.05	Skewing alone
65	3.05	Skewing+range reduction
		Gade Pandu Rangaiah (1985)
72	3.05	By MM
396	3.05	By CM
397	3.05	By ARSM

From the above table it is clear that the number of function evaluations to arrive at optimum results depends on the type of conventional method selected and the minimum number of function evaluation varies from 65 to 1952.

5.2.1.5 Comparison of ANN based Optimiser with Model equation based optimiser for Fuel allocation in power plants problem with initial guess x=[20,0.5,0.5]

A comparison is carried out to check the pros and cons of the proposed Neural Network based approach vis-à-vis the conventional method for optimization. By starting with initial guess of **x**=[20,0.5,0.5] Results obtained by neural network based approach and conventional approach are presented in a table 5.15.

Table 5.15: Comparison between ANN based approach and conventional approach for initial guess x=[20,0.5,0.5] for Fuel allocation in power plants

Parameters	NN based approach	Conventional approach
Initial guess	x=[20,0.5,0.5]	x=[20,0.5,0.5]
Optimum points	x_1=30.0, x_2=20.0, x_3=0.0, x_4=0.583661	x_1=30.0, x_2=20.0, x_3=0.0, x_4=0.583661
Function Value	3.22768 tons/hr	3.05208 tons/hr
CPU time required	0.44 sec	0.82 sec
Function evaluation	13	25
Function Gradient Evaluation	4	7

It is clear form the above table that neural network based approach was able to give results quite close to conventional approach. Value of variables is exactly same as given by conventional approach while error in function is 5.754%. Time taken and function evaluation needed by neural network based approach is much less than conventional approach for same initial guess. The number of function evaluations needed by ANN based approach is only 13 as compared to conventional approach, which requires 25 function evaluations, clearly shows that neural network based approach has edge over conventional approach both in terms of time and function evaluation needed to converge at optimum. Both of these approaches needs much less function evaluations than minimum function evaluations reported by previous investigators, indicating that present work has shown remarkable improvement in terms of number of function evaluation needed to converge at optimum.

5.2.2 Drying process problem

Chung (1972) presented this problem, which is concerned with maximization of the moisture removal rate in a dryer. This non-linear example has only two independent variables but has three inequality constraints. The details of the problem are given in section 3.2.2 of Chapter 3. Many investigators [C2], [L1], [H3], [S1] etc, have solved this problem in past to test their optimisers as discussed in section 2.2.1.2.2 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.4. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the Drying process problem is given below:

5.2.2.1 Development of an ANN model for Drying problem

To model Drying process problem by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the ANN model are equal to the number of input/ output variables. There are 2 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 2 to 10 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 4 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 and found suitable for this problem. Activation function used for this model is unipolar, soft continuous function (LOGSIG/ LOGSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.30 with the help of eqn. 3.31 to 3.34 and constraints 3.35 through 3.37 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.4. The training data set consists of eighty two pair of values of input/ output variables. The time taken for the generation of above data base of training data is 0.05 sec (input data is presented in table E.4 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.1 and 5000 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.16.

Table 5.16: SSE for various NN structure (for Drying problem)

	NN structure (IxHxO)*	SSE
For learning rate 0.1 and 5000 epochs	2 x 2 x 1	0.0182007
	2 x 4 x 1	0.0101749
	2 x 6 x 1	0.0123674
	2 x 8 x 1	0.0112357
	2 x 10 x 1	0.0126395

* Where I is input nodes, H is hidden nodes and O is Output nodes.

For 10 hidden neurons error first increases than decreases i.e. rippling starts after 8 hidden neurons thus maximum number of hidden neuron tried were 10. From the table 5.16 it is clear that 4 hidden neurons are optimum for the present network. The weights of the network are obtained after 9000 epochs with a learning rate of 0.1. The weights and biases of the network are given in table D.13 through table D.16.

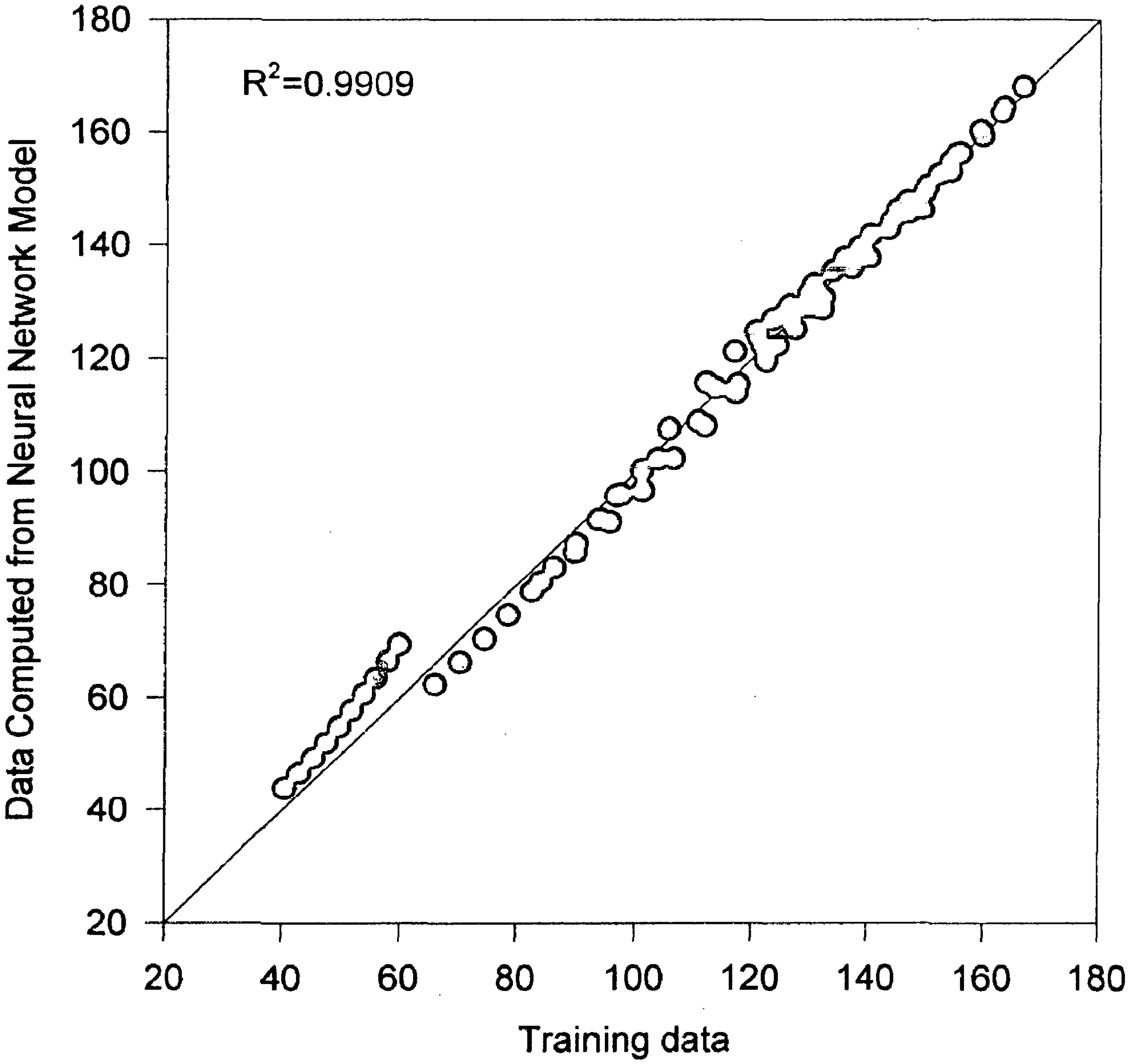


Fig.5.4: Comparison between outputs of training data and the predicted value of output by neural network model

For the Drying problem

The result of training of the ANN model is shown in fig. 5.4. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes out to be 0.9909. The topology of the ANN model for the drying process problem is given below:

Network Topology:

Input nodes	2
Output nodes	1
Hidden layer	1
Hidden nodes	4
Learning rate	0.1

5.2.2.2 Optimization of Drying problem using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.4). When the initial guess of **x** =[800,0.5] is provided during the execution of program, obtained intermediate results of optimization are shown in table 5.17.

Table 5.17: Intermediate results of optimization for initial guess x=[800,0.5] of the drying process problem by ANN based approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [800,0.5]	3	-143.631	-0.0495613	1
	10	-144.498	-0.0477685	0.0625
	13	-176.426	0.00586042	1
	16	-166.947	-0.00580107	1
	19	-171.518	-0.00471498	1
	22	-174.138	4.60568e-006	1
	25	-174.136	-8.71504e-007	1 mod Hess
	28	-174.136	-4.18215e-008	1 mod Hess
	29	-174.136	-3.07837e-013	1 mod Hess

After 0.6 sec of CPU time and 29 functions and constraint evaluation, the ANN based optimization approach gives optimum value of function as -174.136 at x_1= 984.002 and x_2= 0.516417. Number of function gradient evaluation is 9 and active constraints at optimum conditions are g(2).

5.2.2.3 Optimization of Drying problem by Classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.30 with the help of eqn 3.31-3.34 along with the constraints eqn. 3.35-3.37 of the problem is used for optimization. MATLAB optimiser function "constr" finds the constrained minimisation of a function of several variables and is used in the present case, x=constr('fun',x0) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.4). The function 'fun' should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that g ≤ 0.

When the initial guess of **x** =[800,0.5] is provided during execution of program intermediate results of optimization obtained are shown in table 5.18.

Table 5.18: Intermediate results of optimization for initial guess x=[800,0.5] of the drying process problem by conventional approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [800,0.5]	3	-143.503	-0.0495613	1
	6	-166.139	0.00532299	1
	9	-161.431	-0.0190034	1
	12	-171.788	-0.00133115	1
	15	-172.488	2.29322e-006	1
	18	-172.464	-0.0000274235	1
	21	-172.476	-0.0000208752	1
	24	-172.487	9.40369e-010	1 mod Hess
	25	-172.487	-9.60843e-013	1 mod Hess

After 0.82 sec of CPU time and 25 functions and constraint evaluation, optimizer gives optimum value of function as -172.487 at x_1=975.83 and x_2= 0.524446. Number of function gradient evaluation is 9 and active constraints at optimum conditions are g(2).

5.2.2.4 Drying process problem optimised by conventional optimization techniques by different authors ([L1],[H3],[M1],[R1])

Work done by some other researchers are presented in a table 5.19.

Table 5.19: Results of previous investigators for Drying process problem

	Minimum functions evaluations Required to reach 0.1 % of the Optimum	Results	Literature source
Initial Guess [800,0.5]	336	172.5	Luss & Jaakola (1973) by ARSM
	91	172.49	Heuckroth & Gaddy (1976) by ARSM
	29	172.5	Martin & Gaddy (1982) by ARDS
	75 57 122	172.487 172.487 172.487	Gade Pandu Rangaiah (1985) By MM By CM By ARSM

From the above table it is clear that the number of function evaluations to arrive at optimum results depends on the type of conventional method selected and the minimum number of function evaluation varies from 29 to 336.

5.2.2.5 Comparison of ANN based Optimiser with Model equation based optimiser for Drying process problem with initial guess x=[800,0.5]

A comparison is carried out to check the pros and cons of the proposed Neural Network based approach vis-à-vis the conventional method for optimization. By starting with initial guess of **x**=[800,0.5] Results obtained by neural network based approach and conventional approach are presented in a table 5.20.

Table 5.20: Comparison between ANN based approach and conventional approach for initial guess x=[800,0.5] for Drying process problem

Parameters	NN based approach	Conventional approach
Initial guess	x=[800,0.5]	x=[800,0.5]
Optimum points	x_1=984.002 x_2=0.516417	x_1=975.83 x_2=0.524446
Function Value	-174.136	-172.487
CPU time required	0.6 sec	0.82 sec
Function evaluation	29	25
Function Gradient Evaluation	9	9

It is clear form the above table that neural network based approach was able to give results quite close to conventional approach; error in function is only 0.956%. Number of function evaluation is slightly more than conventional approach but total time required by ANN based approach is less than conventional approach, shows that though ANN based approach requires fewer more function evaluations it compensates this drawback with its strength to evaluate the function in far less time than taken by conventional approach. Both of these approaches are near to the lower side of minimum function evaluation needed reported in the literature [M1], proves the usefulness of these approaches.

5.2.3 Alkylation process problem (a)

The objective of this non-linear example is to maximise the profit from an alkylation process. There are ten independent variables, seven of which can be eliminated by equality constraints. Therefore, there are only three independent variables. x_1, x_7 and x_8 chosen as independent variables and from these x_2, x_3, x_4, x_5, x_6, x_9 and x_{10} can be calculated. The details of the problem are given in section 3.2.3.1 of Chapter 3. Many investigators [P1], [S3], [K2], [R1] etc, have solved this problem in past to test their optimisers as discussed in section

2.2.1.2.3 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.5. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the Alkylation problem (a) is given below:

5.2.3.1 Development of an ANN model for Alkylation problem (a)

To model Alkylation process problem (a) by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the ANN model are equal to the number of input/ output variables. There are 3 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 3 to 7 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 5 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 but found unsuitable as it causes error to increase. Therefore, learning rate of 0.02 is taken and found to be suitable for this network. Activation function used for this model is unipolar, soft continuous function (LOGSIG/ LOGSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.38 and constraints 3.39 through 3.55 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.5. The training data set consists of nine hundred forty six set of values of input/ output variables. The time taken for the generation of above data base of training data is 1.09 sec (input data is presented in table E.5 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.02 and 5000 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.21.

Table 5.21: SSE for various NN structure (for Alkylation process (a))

For learning rate	NN structure (IxHxO)*	SSE
0.02	3 x 3 x 1	2.46203
and	3 x 4 x 1	0.920878
5000 epochs	3 x 5 x 1	0.696443

* Where I is input nodes, H is hidden nodes and O is Output nodes.

For 7 hidden neurons error first increases than decreases i.e. rippling starts after 5 hidden neurons thus maximum number of hidden neuron tried were 7. From the table 5.21 it is clear that 5 hidden neurons are optimum for the present network. The weights of the network are obtained after 15000 epochs with a learning rate of 0.02. The weights and biases of the network are given in table D.17 through table D.20.

The result of training of the ANN model is shown in fig. 5.5. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes out to be 0.9712. The topology of the ANN model for the Alkylation process problem (a) is given below:

Network Topology:

Input nodes	3
Output nodes	1
Hidden layer	1
Hidden nodes	5
Learning rate	0.02

After the neural network has been trained, it is able to predict the value of output corresponding to the each set of values of input variables, which may necessarily not be a part of the training data.

5.2.3.2 Optimization of Alkylation process problem (a) using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.5). When the initial guess of **x** =[1745,92.8,8.0] is provided during the execution of program intermediate results of optimization are obtained which are shown in table 5.22.

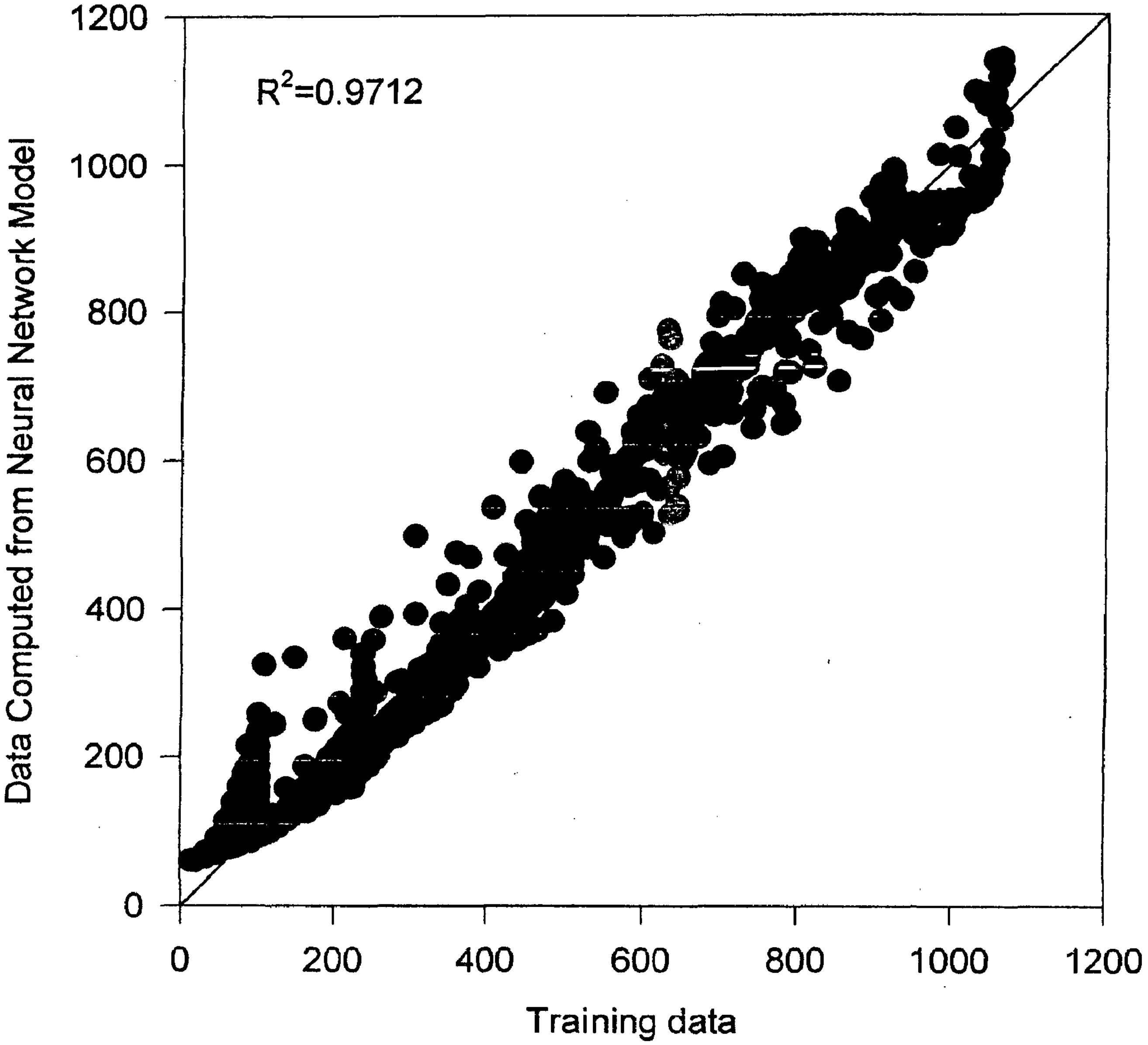


Fig. 5.5 Comparison between outputs of training data and the predicted value of output by neural network model for the Alkyaltion problem (a)

Table 5.22: Intermediate results of optimization for initial guess x=[1745,92.8,8.0] of the Alkylation process problem (a) by ANN based approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [1745,92.8,8]	4	-915.555	-0.4	1
	9	-946.633	-0.645946	0.5
	13	-987.797	90.0716	1
	17	-988.383	-0.0534745	1 mod Hess
	21	-1085.45	1.16509	1
	26	-1108.21	-0.418999	0.5
	31	-1155.89	-0.209499	0.5
	35	-1181.73	-0.212154	1
	39	-1184.1	-0.000852587	1
	43	-1184.16	-0.0213959	1
	47	-1184.16	-0.000372937	1 mod Hess
	48	-1184.16	-6.06115e-007	1 mod Hess

After 1.04 sec of CPU time and 48 functions and constraint evaluation, the ANN based optimization approach gives optimum value of function as -1184.16 at x_1= 1634.69, x_7= 94.8238 and x_8= 10.9381. Number of function gradient evaluation is 12 and active constraints at optimum conditions are g(7).

5.2.3.3 Optimization of Alkylation process problem (a) by classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.38 along with the constraints eqn. 3.39-3.55 of the problem is used for optimization. MATLAB optimiser function "constr" finds the constrained minimisation of a function of several variables and is used in the present case, x=constr('fun',xO) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.5). The function 'fun' should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that g ≤ 0.

When the initial guess of **x** =[1745,92.8,8.0] is provided during execution of program intermediate results of optimization obtained are shown in table 5.23.

Table 5.23: Intermediate results of optimization for initial guess x=[1745,92.8,8.0] of the Alkylation process problem (a) by conventional approach

	f-COUNT	FUNCTION	MAX{g}	STEP Procedures
Initial Guess [1745,92.8,8]	4	-866.151	-0.4	1
	9	-978.344	-0.645946	0.5
	13	-1101.33	90.0716	1
	17	-1146.27	-0.0534745	1
	21	-1158.31	-4.41999e-007	1
	25	-1161.54	0	1
	29	-1162.02	0	1
	33	-1162.03	0	1
	34	-1162.03	0	1

After 1.04 sec of CPU time and 34 functions and constraint evaluation, optimizer gives optimum value of function as -1162.03 at x_1= 1728.37, x_7= 94.1896 and x_8= 10.4144. Number of function gradient evaluation is 9 and active constraints at optimum conditions are g(7), g(10).

5.2.3.4 Alkylation process problem optimised by conventional optimization techniques by different authors ([L1],[K2],[M1],[H3],[R1])

Work done by some other researchers are presented in a table 5.24.

Table 5.24: Results of previous investigators for Alkylation process problem (a)

Minimum functions evaluations Required to reach 0.1 % of the Optimum	Results	Literature source
919	-1162	Luss & Jaakola (1973) by ARSM

3487	-1162	Keefer (1973) by Simpat
132	-1162	Heuckroth & Gaddy (1976) by ARSM
36	-1162	Martin & Gaddy (1982) by ARDS
		Gade Pandu Rangaiah (1985)
171	-1162	By CM
297	-1162	By ARSM

From the above table it is clear that the number of function evaluations to arrive at optimum results depends on the type of conventional method selected and the minimum number of function evaluation varies from 36 to 3487.

5.2.3.5 Comparison of ANN based Optimiser with Model equation based optimiser for Alkylation problem (a) with initial guess x=[1745,92.8,8.0]

A comparison is carried out to check the pros and cons of the proposed Neural Network based approach vis-à-vis the conventional method for optimization. By starting with initial guess of **x**=[1745,92.8,8.0] Results obtained by neural network based approach and conventional approach are presented in a table 5.25.

Table 5.25: Comparison between ANN based approach and conventional approach for initial guess x=[1745,92.8,8.0]

Parameters	NN based approach	Conventional approach
Initial guess	x=[1745,92.8,8]	x=[1745,92.8,8]
Optimum points	x_1= 1634.69,	x_1= 1728.37,
	x_2= 16000,	x_2= 16000,
	x_3= 98.7971,	x_3= 98.1362,
	x_4= 2881.26,	x_4= 3056.04,
	x_5= 1880.44,	x_5= 2000,
	x_6= 92.1083,	x_6= 90.6185,
	x_7= 94.8238,	x_7= 94.1896,
	x_8= 10.9381,	x_8= 10.4144,
	x_9= 2.19333,	x_9= 2.61574,
	x_{10}= 151.472.	x_{10}= 149.569.

Function Value	-1184.16	-1162.03
CPU time required	1.04 sec	1.04 sec
Function evaluation	48	34
Function Gradient Evaluation	12	9

It is clear form the above table that neural network based approach has been able to give results quite close to conventional approach; error in function is only 1.9%. Although function evaluation needed is more but time required by neural network model is same as conventional approach for optimum point, this shows that time required for calculating function by mathematical model is more than ANN model. Both of these approaches are near to the lower side of minimum function evaluation needed reported in the literature [M1], proves the usefulness of these approaches.

5.2.4 Alkylation process problem (b)

The objective of this non-linear example is to maximise the profit from an alkylation process. There are ten independent variables, three of which can be eliminated by equality constraints. Therefore, there are seven independent variables. x_4, x_5, x_6, x_7, x_8, x_9 and x_{10} chosen as independent variables and from this x_1, x_2 and x_3 can be calculated. The details of the problem are given in section 3.2.3.2 of Chapter 3. Many investigators [B6], [B1], [W1], [V3] etc, have solved this problem in past to test their optimisers as discussed in section 2.2.1.2.4 of Chapter 2. For solving this problem necessary code was developed in MATLAB environment, details of this code are presented in Appendix C.6. The first step in solving this problem is to develop a suitable ANN model of the problem. Details of ANN model for the Alkylation process problem (b) is given below:

5.2.4.1 Development of an ANN model for Alkylation problem (b)

To model Alkylation process problem (b) by ANN, a back propagation training algorithm is used. The number of input/ output nodes considered for the

ANN model are equal to the number of input/ output variables. There are 7 inputs and 1 output in this ANN model. For the selection of the number of hidden layer(s) as well as hidden nodes(s) in each layer, there hardly exists any algorithm. A hidden layer consisting 7 to 15 hidden nodes are tried and it is found (by visual inspection and cross validation test) that 9 hidden nodes with one hidden layer is sufficient for the modeling of this problem by ANN. Initially the learning rate (lr) is taken, as 0.1 but found unsuitable as it causes error to increase. Therefore, learning rate of 0.001 is taken and found to be suitable for this network. Activation function used for this model is bipolar, soft continuous function (TANSIG/ TANSIG)(details of this function are given in Appendix A). A database is developed for the cost function of the problem, equation 3.56 and constraints 3.57 through 3.76 with the help of a Fortran program. The details of the Fortran program are given in Appendix B.6. The training data set consists of one thousand three hundred fourteen set of values of input/ output variables. The time taken for the generation of above data base of training data is 14.01 sec (input data is presented in table E.6 along with corresponding target output data and the value of target data predicted by ANN model). For learning rate of 0.001 and 1000 epochs sum squared error (SSE) for various hidden nodes is presented in the table 5.26

Table 5.26: SSE for various NN structure (for Alkylation problem (b))

	NN structure (IxHxO)*	SSE
For learning rate 0.001 and 1000 epochs	7 x 7 x 1	2.50934
	7 x 8 x 1	3.94211
	7 x 9 x 1	0.834765
	7 x 10 x 1	1.15035
	7 x 11 x 1	1.4033
	7 x 13 x 1	1.34559
	7 x 15 x 1	0.844726

* Where I is input nodes, H is hidden nodes and O is Output nodes.

For 15 hidden neurons error first increases than decreases i.e. rippling starts after 13 hidden neurons thus maximum number of hidden neuron tried were 15. From the table 5.26 it is clear that 9 hidden neurons are optimum for the present network. The weights of the network are obtained after 10000 epochs with a learning rate of 0.001. The weights and biases of the network are given in table D.21 through table D.24.

The result of training of the ANN model is shown in fig. 5.6. From the figure it is clear that this ANN model has been trained to a good extend. It can be concluded from the value of the coefficient of determination (R^2), which comes out to be 0.9377. The topology of the ANN model for the Alkylation process problem (b) is given below:

Network Topology:

Input nodes	7
Output nodes	1
Hidden layer	1
Hidden nodes	9
Learning rate	0.001

After the neural network has been trained, it is able to predict the value of output corresponding to the each set of values of input variables, which may necessarily not be a part of the training data.

5.2.4.2 Optimization of Alkylation process problem (b) using ANN model

Once the ANN model has been developed, this is inducted in the MATLAB optimization toolbox to optimize the problem. In doing so mathematical model of the problem is replaced by equivalent ANN model while constraints of this problem are placed in a M-file named "fun1.m"(details are given in Appendix C.6). When the initial guess of **x** =[3.048,1.974,0.892,0.928,8,3.6,1.45] is provided during the execution of program intermediate results of optimization are obtained which are shown in table 5.27.

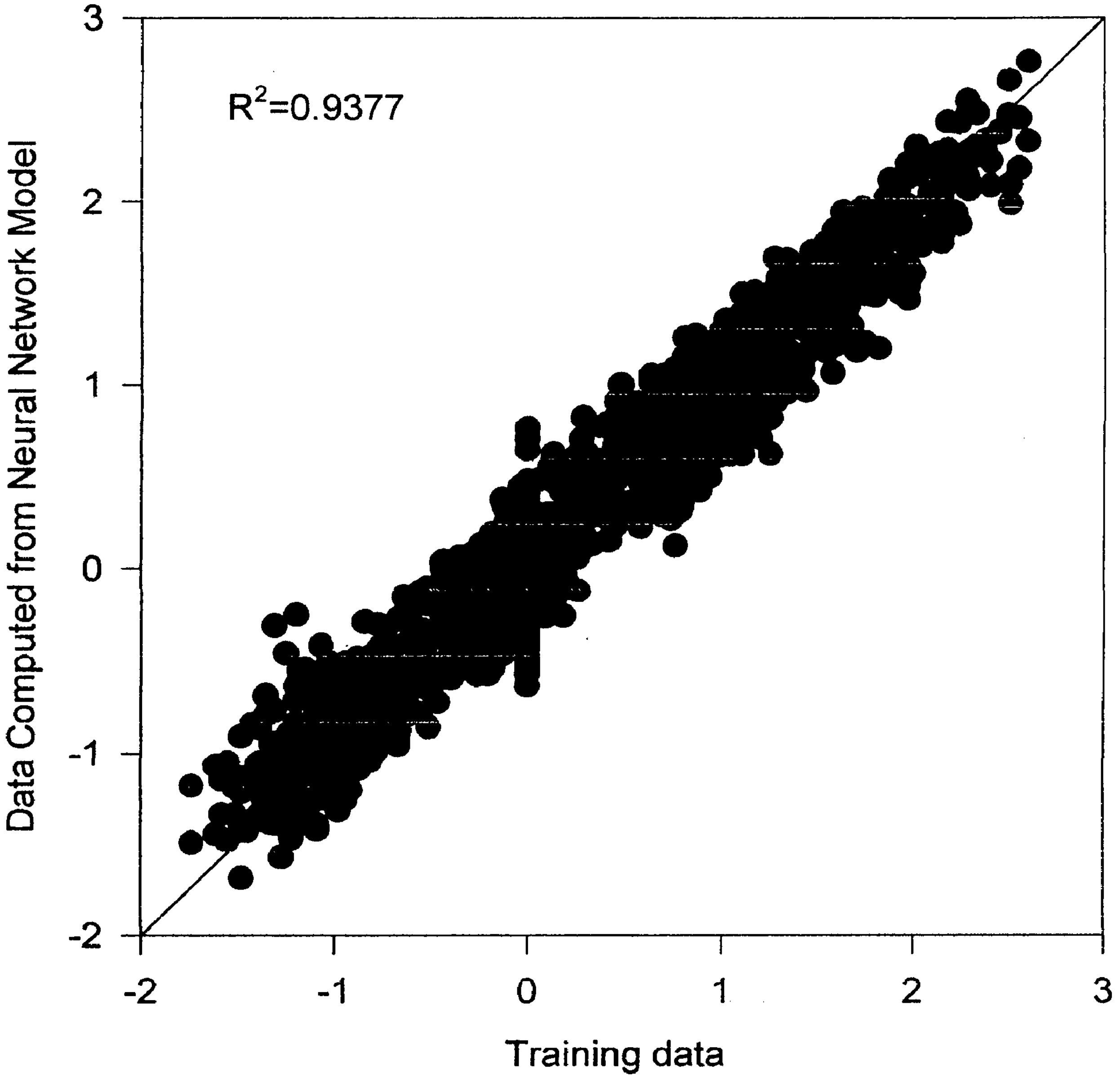


Fig. 5.6 Comparison between outputs of training data and the predicted value of outputs by neural network model

For the Alkylation Problem (b)

Table 5.27: Intermediate results of optimization for initial guess x=[3.048,1.974,0.892,0.928,8,3.6,1.45] of the Alkylation process problem (b) by ANN based approach

f-COUNT	FUNCTION	MAX{g}	STEP Procedures
8	-1.06031	0	1
16	-1.60717	0.000134797	1 mod Hess
24	-1.60956	0.0000507866	1
32	-1.61113	0.000127026	1
33	-1.61102	5.99905e-008	1

After 1.15 sec of CPU time and 33 functions and constraint evaluation, the ANN based optimization approach gives optimum value of function as -1.61102. Number of function gradient evaluation is 5 and active constraints at optimum conditions are g(1), g(4), g(5), g(7), g(14), g(16).

5.2.4.3 Optimization by Classical method

To check the effectiveness of ANN based approach; this problem is also solved with conventional optimization technique. For this, mathematical function, eqn. 3.56 along with the constraints eqn. 3.57-3.76 of the problem is used for optimization. MATLAB optimiser function "constr" finds the constrained minimisation of a function of several variables and is used in the present case, x=constr('fun',x0) starts at x0 and finds a constrained minimisation to the function, which is described in FUN (in fun.m)(details are given in Appendix C.6). The function 'fun' should return two arguments: a scalar value of the function to be minimised denoted by *f* and matrix of constraints, g: [f,g]=fun(x). The function is minimised such that $g \leq 0$.

When the initial guess of **x** =[3.048,1.974,0.892,0.928,8,3.6,1.45] is provided during execution of program intermediate results of optimization obtained are shown in table 5.28.

Table 5.28: Intermediate results of optimization for initial guess x=[3.048,1.974,0.892,0.928,8,3.6,1.45] of the Alkylation process problem (b) by conventional approach

f-COUNT	FUNCTION	MAX{g}	STEP Procedures
8	-0.862975	0	1
16	-1.6513	0.140182	1
24	-1.77718	0.00611215	1
32	-1.76499	1.83673e-006	1 mod Hess
42	-1.76501	0.000013048	0.25 mod Hess
50	-1.765	2.17783e-006	1 mod Hess
51	-1.765	1.24574e-010	1 mod Hess

After 1.7 sec of CPU time and 51 functions and constraint evaluation, optimizer gives optimum value of function as -1.765. Number of function gradient evaluation is 7 and active constraints at optimum conditions are g(1), g(4), g(5), g(7), g(14), g(16).

5.2.4.4 Comparison of ANN based Optimiser with Model equation based optimiser for Alkylation process problem (b) with initial guess x=[3.048,1.974,0.892,0.928,8,3.6,1.45]

To check the efficiency of neural network based approach for optimization, using same starting points as Berna, et al, (1980), Westerberg & Debrosse (1973) etc results obtained by previous investigators as well as results obtained by present work is given in a table 5.29.

Table 5.29: Results of previous investigators and present work for Alkylation problem(b) for initial guess x= [3.048,1.974,0.892,0.928,8,3.6,1.45]

Variable	Starting point	Westerberg & Debrosse (1973)	Berna, et al, (1980)	*Carlos Vinante, et al, (1985)*	NN based Approach	Classical Approach
x_1	1.745	1.6995	1.7040	1.382	1.72821	1.7037
x_2	1.200	1.6000	1.5850	1.151	1.22832	1.58471
x_3	1.100	0.5885	0.5430	0.449	0.84423	0.54309
x_4	3.048	3.0320	3.0360	2.551	3.05591	3.03582
x_5	1.974	2.0000	2.0000	1.726	2.00000	2.0000
x_6	0.892	0.9002	0.9002	0.826	0.92755	0.90132
x_7	0.928	0.9500	0.9500	0.957	0.95000	0.9500

x_8	8.000	10.5900	10.5900	9.570	8.26476	10.4755
x_9	3.600	1.7200	1.7200	1.255	1.56164	1.56164
x_{10}	1.450	1.5353	1.5353	1.550	1.53535	1.53535
Function Value		1.7150	1.7650	1.767	1.61102	1.765

Berna (1980) used Newton-Rapson type method and quadratic programming, Westerberg (1973) used optimization algorithm for structured design systems and Vinante (1985) used MM to obtain results shown in above table 5.29. G. P. Rangaiah (1985) solved this problem by Method of Multipliers and it took 645 function evaluations to reach at optimum point.

By starting with x=[3.048,1.974,0.892,0.928,8.0,3.6,1.45], results obtained by neural network based approach and conventional approach is presented in a table 5.30.

Table 5.30: Comparison between ANN based approach and conventional approach for initial guess x=[3.048,1.974,0.892,0.928,8,3.6,1.45] for Alkylation process problem (b)

Parameters	NN based approach	Conventional approach
Function Value	-1.61102	-1.765
CPU time required	1.15 sec	1.7 sec
Function evaluation	33	51
Function Gradient Evaluation	5	7

It is clear form the above table that neural network based approach was able to give results quite close to conventional approach as well as results of other investigators [W1], [B1] and [V3] (error in function value is 8.72%). Time taken and function evaluation needed to reach at optimum is quite less than conventional approach. This example shows that ANN based approach can be used in trivial cases where there are several decision variables in objective function.

5.3 General Comment on ANN based approach and conventional approach for all selected problems

Both of these approaches were successful in solving selected mathematical as well as Chemical engineering problems. It has been seen that in none of the cases neural network based approach is inferior to conventional approach. For Chemical engineering problems, it has shown superiority over conventional approach because in Fuel allocation in power plant problem, ANN based approach require only 13 function evaluation with 0.44 sec of CPU time as compared to 25 function evaluation with 0.82 sec of CPU time required by conventional approach; in Drying process problem although function evaluation is more (4 extra function evaluations needed by ANN based approach) but total time required is less by ANN based approach; in Alkylation process (a), for time taken by both approaches are same though function evaluation needed by ANN based approach is more; in Alkylation process (b), time and function evaluation needed by ANN based approach is less than conventional approach. Thus it has been seen that ANN based approach is fast and reliable for obtaining optimum results for Chemical engineering problems. It has also been proved by taking two different initial guesses for four variable problem, that number of function evaluations needed to converge towards optimum depends on choice of initial guess. Both of these approaches have limitation that they may some time give local solutions this has been observed while solving Hesse's function problem. A primary advantage of this ANN based approach is the ease with which it can be programmed and applied. Because of the good reliability, reasonable efficiency, and ease of programming, ANN approach may find wide application in process optimization problems.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATION

Following salient conclusions are drawn from the present investigation:

1. This study successfully demonstrated the use of ANN to model the functional relationships between input and output variables for mathematical functions as well as Chemical engineering problems with equal perfection. It has also been found that:

A) Large learning rate may some time create the convergence problem. For present work learning rate (lr) varies from 0.001 to 0.1.

B) If the hidden neurons are under specified, the neural network model may not generalize well and only reproduce training data correctly rather than predicting general trend correctly.

C) If the hidden neurons are over specified, then it may cause convergence problem due to oscillations.

D) Neural network does not require the knowledge about the physics of the problem. This is its main advantage that it can be used for modeling complex and less understood problems. Where mathematical modeling is difficult or not at all possible.

E) The main disadvantage of this approach is that it requires lot training data to model the system.

2. This study successfully demonstrated the use of ANN based optimization approach for non-linear constrained mathematical as well as Chemical engineering problems. During optimization of the problems it has been observed that:

i) Number of function evaluation needed and time required to converge at optimum primarily depend upon choice of initial guess of decision variables.

ii) If the same number of function evaluations needed for optimum solution, ANN based approach takes less time than conventional approach due to fast computation of objective function value by the former.

iii) Accuracy of the optimum value of function varies from 0.89 % to 8.72 %, which shows that optimum solution given by neural network based approach is quite comparable with conventional approach.

RECOMMENDATIONS FOR THE FUTURE STUDY:

Based on the present study, following is recommended for the future work.

a) The effect of various network topologies with different training algorithms (Levenberg- Marquardt) on process identification and optimization should be studied.

b) Present study is concerned with single objective optimization only; in future, applicability of Neural Network based approach for Multi-objective optimization should be studied.

c) A strategy involving artificial neural network and genetic algorithm should be evaluated for the global optimization with later as front-end optimizer.

REFERENCES

[B1] Berna, T. J., *et al*, "A New Approach to Optimisation of Chemical Processes," *AIChE J.*, **26**(1), 37 (1980).

[B2] Bhagat, P., "An Introduction to Neural Nets," *Chem. Engng. Prog.*, **8**, 55 (1990).

[B3] Bhat, N., and McAvoy, T. J., "Use of Neural Nets for Dynamic Modelling and Control of Chemical Process Systems," *Computers chem. Engng.*, **14**(4/5), 573 (1990).

[B4] Biggs, M. C., "Constrained Minimization Using Recursive Quadratic Programming," *Towards Global Optimization*, North-Holland, 341-349 (1975).

[B5] Billings, S. *et al*, "Properties of Neural Networks with application to modelling non-linear dynamical systems," *International Journal of Control,"* **55**(1), (1992).

[B6] Bracken, J., and McCormick, G. P., *Selected Applications of Non-linear programming,* John Wiley, N.Y. (1968).

[C1] Chen, Q., and Weigand, W. A., "Dynamic Optimization of Nonlinear Processes by Combining Neural Net Model with UDMC," *AIChE J.*, **40**(9), 1488 (1994).

[C2] Chung, S. F., "Mathematical Model and Optimisation of Drying Process for a Through-Circulation Dryer," *Can. J. Chem. Engng.*, **50**(10), 657 (1972).

[D1] Dan Hammerstrom, "Working with neural networks," *IEEE SPECTRUM*, 7, 46 (1993).

[D2] Demuth, H., *Neural Network Toolbox User's Guide,* The Math Works, Inc. January(1994).

[F1] Fletcher, R., "Practical Methods of Optimization," *Vol. 1, Unconstrained Optimization,* and *Vol. 2, Constrained Optimization,* John Wiley and Sons., (1980).

[G1] Gill, P. E. *et al, Practical Optimization,* Academic Press, London, (1981).

[G2] Gill, P. E. *et al, Numerical Linear Algebra and Optimization,* Vol.1, Addison Wesley, (1991).

[G3] Gould, F. J., “Non-linear pricing: Applications to Concave programming,” *Operations Research*, 19, 1026 (1971).

[G4] Grace, A., *Optimization Toolbox User's Guide,* The Math Works, Inc. October (1994).

[H1] Han, S. P., “A Globally Convergent Method for Nonlinear Programming,” *Journal of Optimization Theory and Applications*, **22**, 297 (1977).

[H2] Hesse, R., “A Heuristic Search procedure for Estimating a global solution of Non-convex Programming problem,” *Operations Research*, **21**, 1267 (1973).

[H3] Heuckroth, M. W., and Gaddy, J. L., “Process Optimization with the adaptive randomly directed search technique,*” AIChE J.*, **22**, 744 (1976).

[H4] Himmelblau, D. M., and Hoskins, J. C., “Artificial Neural Network models of Knowledge representation in Chemical Engineering,” *Computers chem. Engng.*, **12**(9/10), 881(1988).

[H5] Hovanessian, S. A. and Stout, T. M., “Optimum fuel allocation in power plants,” *Trans. IEEE Power Apparatus Syst.,* **82**, 329 (1963).

[H6] Hugget, A. et al, “Global Optimization of a Dryer by using Neural Networks and Genetic Algorithms,*” AIChE J.*, **45**(6), 1227 (1999).

[I1] Ishida, M. and Zhan, J., "Neural Model-Predictive Control of Distributed Parameter Crystal Growth Process,” *AIChE J.,* **41**(10), 2333 (1995).

[K1] Karla Yale, “Preparing the right data diet for training neural networks,” *IEEE SPECTRUM,* 3, 64 (1997).

[K2] Keefer, D.L., “ Simpat: Self bounding Direct Search method for Optimization,” *Ind. Engng. Chem. Proc. Des. Dev.*, **12**, 92 (1968).

[K3] Kosko, B., *Neural networks and fuzzy systems*, PHI Private Limited, 3rd ed. (1996)

[L1] Luus, R., and Jaakola, T. H. I., “A Direct approach to optimization of a Complex System,” *AIChE J.*, **19**(3), 645 (1973).

[L2] Luus, R. and Wang, B. C., “Reliability of Optimisation procedures for Obtaining global Optimum,” *AIChE J.*, **24**, 619 (1978).

[M1] Martin, D. L., and Gaddy, J. L., “Process Optimization with the Adaptive Randomly Directed Search,” *AIChE J. Symp. Ser.*, **78**, 99 (1982).

[N1] Nascimento, C. A. O., and Giudici, R., "Neural Network based approach for Optimization applied to an industrial Nylon-6,6 polymerization process," *Computers chem. Engng.*, **22**,Suppl., S595 (1998).

[P1] Payne, R. E., "Alkylation-What you should know about this process," *Petro. Refiner*, **37**, 316 (1958).

[P2] Powell, M. J. D., "The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations," *Nonlinear Programming 3*, Academic Press, (1978).

[P3] Powell, M. J. D., "Variable Metric Methods for Constrained Optimization," *Mathematical Programming: The State of Art,* Springer Verlag, 288-311 (1983).

[R1] Rangaiah, G. P., "Studies in Constrained Optimization of Chemical Process Problems," *Computers chem. Engng.*, **9**(4), 395 (1985).

[R2] Rosen, J. B., and Suzuki, S., "Construction of Non-linear Programming test problems," *Comm. ACM*, **8:2**, 113 (1965).

[S1] Salcedo, R., et al, "An improved Random Search Algorithm for Non-linear Optimization," *Computers chem. Engng.*, **9**(4), 395 (1990).

[S2] Samdani, G., "Neural nets- They learn from examples," *Chemical Engineering*, **97**(8), 37 (1990).

[S3] Sauer, R. N., et al, " Computer Points in the way to more profits, Hydrocarbon Processing," *Petro. Refiner*, **43**, 84 (1964).

[S4] Savkovic-Stevanovic J., "A Neural Network model for Analysis and Optimisation of Process," *Computers chem. Engng.*, **17**, S411 (1993).

[S5] Savkovic- Stevanovic J., "Neural Network for process analysis and optimisation: Modelling and Applications," *Computers chem. Engng.*, **18**(11/12), 1149 (1994).

[S6] Schittowski, K., and Hock, W., "A comparative Performance Evaluation of 27 Nonlinear Programming Codes," *Computing*, **30**, 335 (1983).

[S6] Schittowski, K., "NLQPL: A FORTRAN- subroutine Solving Constrained Nonlinear Programming Problems," *Operational Research*, **5**, 485 (1985).

[T1] Thompson, M. L., and Kramer, M. A., "Modelling Chemical Processes Using Prior Knowledge and Neural Networks," *AIChE J.*, **40**(8), 1328 (1994).

[V1] Van Can, H. J. L. *et al*, "Strategy for Dynamic Process Modelling Based on Neural Networks in Macroscopic Balances," *AIChE J.*, **42**(12), 3403 (1996).

[V2] Venkatasubramanian, V. and McAvoy, T. J., "Neural Network Applications in Chemical Engineering," *Computers chem. Engng.*, **16**(4), v (1992).

[V3] Vinante, C. and Valladares, E., "Application of the method of Multipliers to the Optimization of Chemical Processes," *Computers chem. Engng.*, **9**(1), 83 (1985).

[V4] Vishveswaran, V., *et al*, "A global Optimization Algorithm (GOP) for certain classes of Non-convex NLPS-II: Application of Theory and Test problems," *Computers chem. Engng.*, **14**(12), 1419 (1990).

[W1] Westerberg, A., and Debrose, C., "An Optimization Algorithm for Structured Design Systems," *AIChE J.*, **19**(2), 335 (1973).

[Y1] Yegnanarayana, B., *Artificial neural network*, Prentice-Hall of India, New Delhi (1999).

[Z1] Zurada, J. M., *Introduction to Artificial Neural systems*, Jaico publishing House, Mumbai (1997).

APPENDIX – A

OVERVIEW OF NEURAL NETWORK

each performing simple computation. Typically a human brain consists of 10-500 billion neurons. Neurons are arranged into networks, each network approximately consists of 100,000 neurons. The axon of each neuron connects to about 100 (but sometimes several thousands) other neurons.

Current ANNs are much simpler than biological counterparts, comprising many fewer components and operating in an abstracted manner. For example, the number of neurons, in what is currently considered a sizeable network, would be approximately 1000 neurons with 1,000,000 connections. [*Zurada (1996)*]

A.1.1.1 The biological prototype

The elementary nerve cell, called a neuron, is the fundamental building block of the biological neural network (figure A.1). A typical cell has three major regions: the cell body, which is also called soma, the axon, and the dendrites. Dendrites form a very fine bush of thin fibers around the neuron's body. Dendrites receive information from neurons through axons -- long fibers that serves as transmission lines. An axon is a long thin tube that carries impulses from the neuron. The axon-dendrite contact organ is called a synapse. The synapse is where the neuron introduces its signal to the neighboring neuron. The strength of the given signal is determined by the efficiency of the synaptic transmission. The signal flows impinging upon a dendrite may be either inhibitory or excitatory. A neuron will fire, i.e. send an impulse down its axon, if its excitation exceeds the inhibition by a critical amount, the threshold of the neuron. [*Zurada (1996)*]

A.1.2 The Artificial Neuron Model

An artificial neuron (figure A.2) models the behavior of the biological neuron. Each artificial neuron receives a set of inputs. Each input is multiplied by weights analogous to a synaptic strength. The sum of all the weighted inputs determines the degree of firing called the activation level.

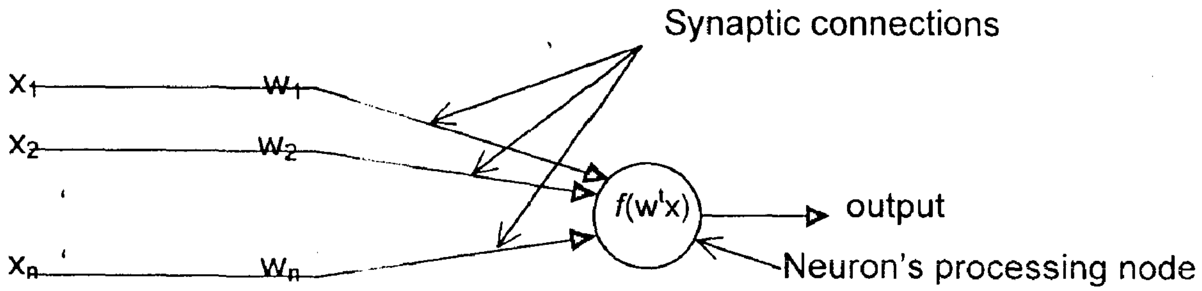


Figure A.2 : General symbol of neuron consisting of processing node and synaptic connections

A.1.3 Neural Network Architecture

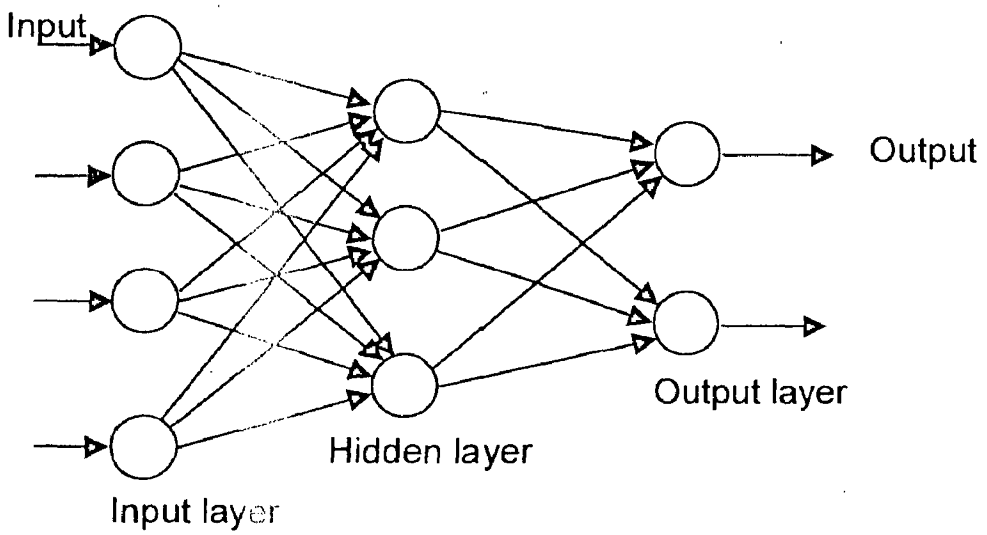


Figure A.3: Typical Neural Network

A neural network is massive parallel, interconnected network made-up of a number of inter connected software or hardware building blocks called elements or neurons. The degree to which one processing element relates to another is a function of its connection strength or weight. The weights of the pertinent connections are stored in a matrix. Inputs to each neuron are combined and the neuron produces an output if the sum of the inputs exceeds an internal threshold value.

The processing elements are usually grouped together into linear arrays called layers. Most ANNs consists of three to five layers, namely the input layer, the middle or hidden layer(s) and the output layer, with connections between neurons of intermediate layers as illustrated in Figure A.3. The input layer usually acts as an input data holder, which distributes input to the first hidden layer.[*Samdani,(1990),Himmelblau,et al(1988)*]

The signal flows from the input layer to the output layer. A basic element of the network, the *i*th neuron in the *l*th layer is shown in Figure A.4.

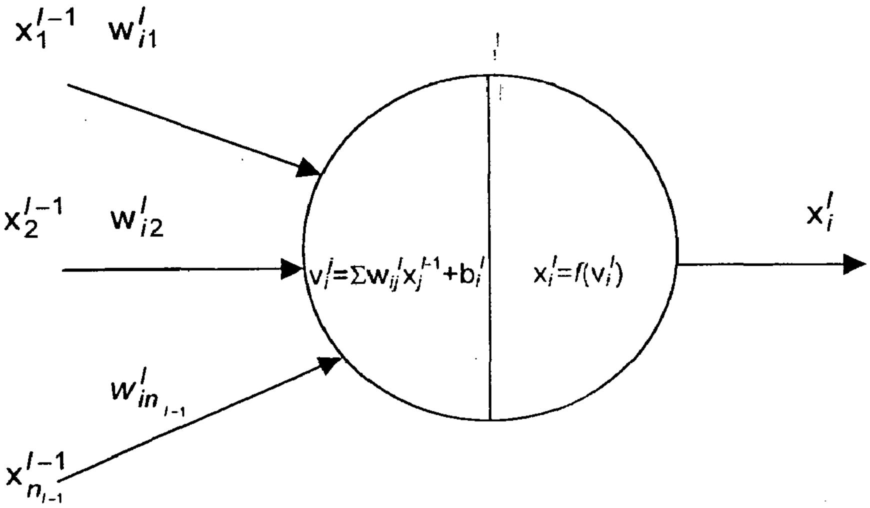


Figure A.4: A hidden neuron *i* of layer *l*

A neuron perform two functions namely the combining function and the activation function. The combining function produces activation for the neuron, v_i^l, defined as:

$$v_i^l = \sum_{j}^{n_{l-1}} w_{ij}^l x_j^{l-1} + b_i^l \tag{A.1}$$

where w_{ij}^l is the weight connection between the *j*th neuron of the (*l*-1)th layer and the *i*th neuron of the *l*th layer, b_i^l is the threshold of the neuron and n_{l-1} is the number of neuron in the (*l*-1)th layer. The activation function performs a non-linear transformation to give the output, x_i^l

$$x_i^l = f(v_i^l) \tag{A.2}$$

where *f*(.) is called the non-linear transformation or activation function. [*Billings, et al*,(1992)]

A.1.4 Selection Of The Network Topology

A.1.4.1 Number of Input nodes

For modeling of problems the number of input/output nodes is equal to the number of input/output variables.

A.1.4.2 Number of hidden units (or layers)

There is no way to determine a good network topology just from the number of input and outputs. It depends critically on the number of training examples and complexity of the classification. There are problems with one input and one output that requires millions of hidden units, problems with a million inputs and a million outputs that requires only one hidden unit, or not at all.

There is some "rule of thumb" for choosing a topology - N inputs plus N outputs divided by two, may be with a square root - but such rules are total garbage. Although there has been some research on the design of optimal ANN structure, it is still largely an art to determine the number of hidden layer and number of hidden units in hidden layer. [*Ishida, et al (1995)*]

A.1.5 Selection of Transfer Function

There are various types of transfer function used in neural network training, depending on the application (i.e. whether it is classification or approximation etc.), but for several reasons mentioned below the logistic sigmoid is most popular:

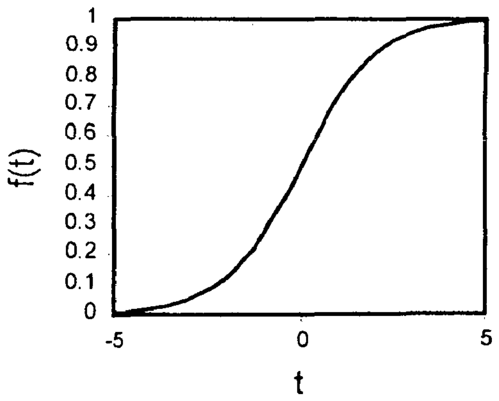


Figure A.5: Unipolar, soft, continuous logistic sigmoid transfer function

1. It is semi-linear (i.e. non-decreasing and differentiable everywhere).

2. It is expressive in closed form.
3. Modification and extensions lead to another squashing functions.
4. The derivative of the sigmoid w.r.t. network output is very easy to form.
5. It has a biological basis as the average firing frequency of biological neurons as a function of excitation, follows a sigmoidal characteristics.
6. It has greatest slope for inputs equal to zero, this serves to mitigate problems caused by the possible dominating effective of large input signals.

Logistic sigmoid transfer function (Figure A.5) is defined as follows:

$$f(\mathrm{t})=\frac{1}{1+e^{-t}}$$

The other transfer function used in neural network training is Bi-polar continuous function. The word "bipolar" is used to point out that both positive and negative responses of neurons are produced for this type of transfer function.

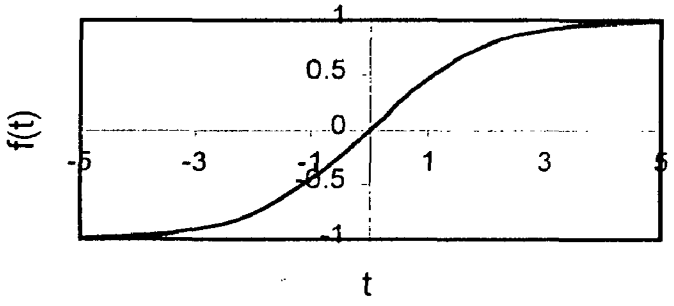


Figure A.6: Bipolar, soft, continuous transfer function

Bipolar continuous transfer function (Figure A.6) is defined as follows:

$$f(\mathrm{t})=\frac{2}{1+e^{-t}}-1$$

A.1.6 Models Of Artificial Neural Networks (ANNs)

One may define ANNs as physical cellular networks that are able to acquire, store, and utilize experimental knowledge and these characteristics has been related to the networks capabilities and performance. The neural network can also be defined as an interconnection through weights to all other neurons including themselves both lag-free and delay connections are allowed. [*Zurada (1996)*]

There are two general structures of neural networks: feed-forward and feed-backward (or recurrent).

A.1.6.1 Feed forward network

Feed-forward networks are most widely used for pattern recognition/ classification. Another relevant area is approximation theory, where the problem may be stated as follows: given two signals x(t) and y(t) find a 'suitable' approximation of the functional relation between the two. This aspect is used for process model development for optimization.[*Huggat (1999)*]

A.1.6.2 Feed-back network

A feedback network can be obtained from the feed-forward network by connecting the neuron's outputs to their inputs. These networks are qualitatively different from feed-forward one's because their structure incorporates feedback. In general, the output of every neuron is fed back with varying gains (weights) to the inputs of all neurons. Feedback (or Recurrent network) networks are quite powerful because they are sequential rather than combinatorial. The feedback from output to input allows the network to temporal behavior. For instance, the output of such networks can oscillate or converge. [*Demuth et al (1995)*]

A.1.6.3 Supervised and unsupervised networks

These feed-forward or feed back networks may be supervised or unsupervised. In supervised learning we assume that at each instant of time when the input is applied, the desired response of the system is provided by the teacher. The error between the actual and the desired response is measured and is used to correct network parameters externally. A set of input and output patterns called a training set is required for this learning mode.

In learning without supervision, the desired response is not known; thus, explicit error information cannot be used to improve network behavior. In this mode of learning, the network must discover for itself any possible existing patterns, regularities, separating properties, etc. Which is called Self-organization. [*Zurada (1996)*]

A.1.7 Learning

ANN learn patterns of activation hence learning can be equated to determining the proper values of connection strength that allows all the nodes to achieve the correct state of activation for given pattern of inputs. Once the pattern of activation is established, the resulting outputs let the network classify an input pattern. The adaptive nature of ANN allows the weights to be learned by experience, thus producing self-organizing system. [*Himmelbalu, et al(1988)*]

A robust learning heuristic for multi layered feed forward ANN called generalized delta rule (GDR) or back propagation learning rule used successfully in training the NN for wide applications. In the feed forward nets inputs feed through hidden layers to an output layer. Each neuron forms a weighted sum of inputs from previous layers to which it is connected, adds a threshold value and produces a non-linear function of this sum as its output value. The output values serves as the input to the next layer to which the neuron is connected, and the process is repeated until output values are obtained for the neurons in the output layer. Thus each neuron performs

$$y_j^p = f(\sum_i w_{ij}.x_j^p - \theta_j) \qquad \text{(A.3)}$$

where w_{ij} is the weight from neuron i to neuron j, w_{ij} can be positive or negative real number and θ_j is the threshold of the jth neuron, p means the pth pattern. The *f*(x) is non-linear function of activation that is often chosen to be of a sigmoidal form.

If d_j^P are the desired outputs and y_i^P are the outputs obtained from the output layer for the pth pattern. Neural nets are trained by minimizing the error function

$$E = \sum_{p=1}\sum_{i=1}(d_i^p - y_i^p)^2 \qquad \text{(A.4)}$$

where i indexes the number of neurons in the output layer, and p means the pth input pattern of the training set is presented on the input layer.

Minimizing the sum of square of errors is not always the best way of training a NN but for some applications it suffices. Back propagation by GDR, a kind of gradient descent method is one popular method. The gradient descent is described by the following equations. The commonly used steepest descents procedure in minimizing E is to change w_{ij} and θ_j by Δw_{ij} and $\Delta\theta_j$ where

```
        for k=1 to $n_3$ ($n_3$ is the number of neurons in the output layer)
        calculate neuron outputs in the output layer $y_k$
          calculate $\delta_k^p$
          $\theta_k^{(p)} = \theta_k^{(p-1)} + \Delta\theta_k^{(p)}$
        end for
        for j=1 to $n_2$ and k=1 to $n_3$
        $w_{jk}^{(p)} = w_{jk}^{(p-1)} + \Delta w_{jk}^{(p)}$
        end for
        for j=1 to $n_2$
        calculate $\delta_j^p$
        $\theta_j^{(p)} = \theta_j^{(p-1)} + \Delta\theta_j^{(p)}$
        end for
for i= 1 to $n_1$ and j= 1 to $n_2$
($n_1$ is number of neurons in the input layer )
$w_{ij}^{(p)} = w_{ij}^{(p-1)} + \Delta w_{ij}^{(p)}$
end for
end for
until $\Delta w < \xi$ ($\xi$ is the convergence criterion)
```

A.1.7.2 Properties of back propagation algorithm

1. The use of a gradient descent algorithm to train its weights makes it slow to converge. [*Zurada (1996)*]
2. It may be trapped at local minima. [*Himmelbalu, et al(1988)*]
3. It can be sensitive to user selectable parameter. [*Himmelbalu, et al(1988)*]
4. Being a feed-forward algorithm, the advantage of this method is that each adjustment step is computed quickly without presentation of all the patterns and without finding an overall direction of the descent for the training cycle. [*Himmelbalu, et al(1988)*, [*Zurada (1996)*]]

5. It modified the synaptic connection strengths with non-local error information. Nonlocality, synchrony supervision and lengthy training cycles precluded biological plausibility. [*Hammerstrom, (1993)*]

A.1.8 Training/ Testing Data Set And Performance Criteria For ANN

A.1.8.1 Training data set

The set of input & output vectors that the network is trained on is commonly referred as training data set. The BP algorithm, which is used to train layered feed-forward networks, minimizes the squared-error when the net is mapping samples of input vectors to output vectors. For each sample pair the connection weights are modified by a small amounts in the direction of the negative derivative of the squared error E (eqn.A.4) to the corresponding weights.

A.1.8.2 Testing data set

Once the training is completed, the network has memorized the knowledge contained in the training data set in the form of adjusted weights. In order to evaluate the performance of the ANN, these weights are then used to process the data whose target values are also known similar to training data. If the performance of the network on testing data is found to be satisfactory, the network is supposed to have generalization capability over any other set of similar data.

A.1.8.3 Overfitting

'Overfitting' (often also called 'overtraining' or 'overlearning') is the phenomenon that in most cases a network gets worse instead of better after a certain point during training, when it is trained to as low error as possible. This is because such long training may make the network 'memorise' the training patterns, including all of their peculiarities. However, one is usually interested in the generalisation of the network, i.e. the error it exhibits on examples not seen during training. Learning the peculiarities of the training set makes the generalisation worse. The network should only learn the general structure of the examples.

There are various methods to fight overfitting. The two most important classes of such methods are regularisation method (such as weight decay) and early stopping. Regularisation methods try to limit the complexity of the network such that it is unable to learn peculiarities. Early stopping aims at stopping the training at the point of optimal generalisation.

A.1.8.4 Stopping criteria

In ANNs training means adjusting connection weights to get useful output set. In training phase, the process of adjusting the weights based on the gradients is repeated until a minimum error is reached. In practice, one has to choose a stopping condition. There are several criteria that can be considered e.g. 1. Based on the error to be minimised 2. Based on the gradient 3. Based on cross-validation performance.

The first two criteria are sensitive to choice of parameters and may lead to poor results if the parameters are poorly chosen. The cross-validation criteria do not have this draw back. It can avoid overfitting the data and can actually improve performance of the network. However, cross validation is much more computationally insensitive and often demands more data.

APPENDIX - B

FORTRAN PROGRAMS FOR DATA GENERATION

APPENDIX B.1

FORTRAN program for data generation of 4 variable problem

```
C     4 Variable Problem
      INCLUDE  'FLIB.FI'
      INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund
      END

      INCLUDE  'FLIB.FD'
      INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund
      OPEN (2,FILE='4Variable.DAT')
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
      N=0
      X1=-1
      DO 6 J=1,5
      X2=-1
      X1=X1+0.5
      DO 4 K=1,5
      X3=-1
      X2=X2+0.5
      DO 3 L=1,7
      X4=-2
      X3=X3+0.5
      DO 5 I=1,5
      X4=X4+0.5
      A=(X1**2+X2**2+X3**2+X4**2-5*X1-5*X2-21*X3+7*X4)
      B=(X1**2+2*X2**2+X3**2+2*X4**2-X1-X4-10)
      C=(2*X1**2+X2**2+X3**2+2*X1-X2-X4-5)
      IF((A.LE.0).AND.(B.LE.0).AND.(C.LE.0))THEN
      Y=(X1**2+X2**2+2*X3**2+X4**2-5*X1-5*X2-21*X3+7*X4)
      M=M+1
      WRITE(2,8)X1,X2,X3,X4,Y
 8    FORMAT(2X,F6.2,2X,F6.2,2X,F6.2,2X,F6.2,2X,F6.2)
      GO TO 5
      ELSE
      N=N+1
      GOTO 5
      ENDIF
5     CONTINUE
3     CONTINUE
4     CONTINUE
6     CONTINUE
      WRITE(2,*)'TOTAL VALUES ARE=',M
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
      STOP
      END
C************************
      SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund, thour
      CHARACTER*1 ap

      IF (hour .GT. 12) THEN
        ap = 'p'
        thour = hour - 12
      ELSE
        thour = hour
        ap = 'a'
      ENDIF

      WRITE (2, 9002) hour, minute, second, hund

9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')
      END
```

APPENDIX B.2

FORTRAN program for data generation of Hesse's function

```
C     18 MAXIMA PROBLEM  (HESSE'S FUNCTION)

      INCLUDE  'FLIB.FI'
      INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund
      END
      INCLUDE  'FLIB.FD'
      INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund
      INTEGER Z,H
      OPEN (2,FILE='18MAX.DAT')
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)

      X1=-1
      DO 2 J=1,6
      X2=-1
      X1=X1+2
      DO 3 K=1,6
      X3=-3
      X2=X2+2
      DO 4 L=1,9
      X4=-2
      X3=X3+2
      DO 5 I=1,7
      X5=-1
      X4=X4+2
      DO 6 Z=1,6
      X6=-2
      X5=X5+2
      DO 7 H=1,11
      X6=X6+2
      A=X1+X2
      B=-X1+X2
      C=X1-X2*3
      D=(X3-3)**2+X4
      F=(X5-3)**2+X6
      IF((A.LE.6).AND.(A.GE.2).AND.(B.LE.2).AND.(C.LE.2).AND.(D.GE.4)
     +.AND.(F.GE.4).AND.(X3.LE.5).AND.(X4.LE.6).AND.(X5.LE.5).AND.(X6
     +.LE.10).AND.(X1.GE.0).AND.(X2.GE.0).AND.(X3.GE.1).AND.(X4.GE.0)
     +.AND.(X5.GE.1).AND.(X6.GE.0))THEN
      Y=25*(X1-2)**2+(X2-2)**2+(X3-1)**2+(X4-4)**2+(X5-1)**2+(X6-4)**2
      M=M+1
      WRITE(2,10)X1,X2,X3,X4,X5,X6,Y
10     FORMAT(1X,F6.2,1X,F6.2,1X,F6.2,1X,F6.2,1X,F6.2,1X,F6.2,1X,F10.3)
      GO TO 7
      ELSE
      GOTO 7
      ENDIF
```

```
7    CONTINUE
6    CONTINUE
5    CONTINUE
4    CONTINUE
3    CONTINUE
2    CONTINUE
     WRITE(2,*)'TOTAL VALUES ARE=',M
     CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
     CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
     STOP
     END
C***************************************************************
     SUBROUTINE SHOWTIME(hour, minute, second, hund)
     INTEGER*2 hour, minute, second, hund, thour
     CHARACTER*1 ap
     IF (hour .GT. 12) THEN
       ap = 'p'
       thour = hour - 12
     ELSE
       thour = hour
       ap = 'a'
     ENDIF
     WRITE (2, 9002) hour, minute, second, hund
9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')
     END
```

APPENDIX B.3

FORTRAN program for data generation of Optimum Fuel Allocation

```
C     FUEL ALLOCATION

      INCLUDE  'FLIB.FI'
      INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund
      END

      INCLUDE  'FLIB.FD'
      INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund

      OPEN (2,FILE='power.DAT')
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)

      X1=17

      DO 2 J=1,30
      X2=-0.2

      X1=X1+1

      DO 3 K=1,30
      X2=X2+0.2
      X3=-0.2
      DO 4I=1,30
      X3=X3+0.2


      f1=1.4609+0.15186*X1+0.00145*X1*X1
      f2=1.5742+0.1631*X1+0.001358*X1*X1

      r2=50-X1

      g1=0.8008+0.2031*r2+0.000916*r2*r2
      g2=0.7266+0.2256*r2+0.000778*r2*r2
      A=((1-X2)*f2+(1-X3)*g2)

      IF((A.LE.10).AND.(r2.GE.14).AND.(r2.LE.25).AND.(X1.LE.30).AND.(X1.
     +GE.18).AND.(X2.GE.0).AND.(X2.LE.1).AND.(X3.LE.1).AND.(X3.GE.0))THE
     +N


      Y=((X2*f1)+(X3*g1))

      M=M+1
      WRITE(2,*)X1,X2,X3,Y

      GO TO 4
      ELSE
      GOTO 4
      ENDIF
4      CONTINUE
3      CONTINUE
2      CONTINUE
      WRITE(2,*)'TOTAL VALUES ARE=',M
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
      STOP
      END
C***********************************************************************
      SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund, thour

      CHARACTER*1 ap

      IF (hour .GT. 12) THEN
        ap = 'p'
        thour = hour - 12
      ELSE
        thour = hour
        ap = 'a'
      ENDIF

      WRITE (2, 9002) hour, minute, second, hund

9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')

      END
```

APPENDIX B.4

FORTRAN program for data generation of Drying problem

```
C   Drying rate
    INCLUDE  'FLIB.FI'
    INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
    INTEGER*2 hour, minute, second, hund
    END
    INCLUDE  'FLIB.FD'
    INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund

    OPEN (2,FILE='DRYING.DAT')
    CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
    CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)

    X1=500
    DO 2 J=1,10
    X2=0.01
    X1=X1+50
    DO 3 K=1,15
    X2=X2+0.1

    Z=(X2)/(X1**0.41)
    F=1.-EXP(-107.9*Z)
    G=1-EXP(-5.39*Z)
    H=(0.036/F)+0.095-(((9.27E-4)*LOG(G/F))/(Z))
    A=0.2-(4.62E-10)*(X1**2.85)*X2-(1.055E-4)*X1
    B=(1.75/12)-(8.2E-7)*(X1**1.85)*X2
    C=2.-(109.6*Z*H)
    IF((A.GE.0).AND.(B.GE.0).AND.(C.GE.0))THEN
    Y=(0.033*X1)/H
    M=M+1

    WRITE(2,10)X1,X2,Y
 10 FORMAT(1X,F6.1,1X,F10.2,5X,F9.5)

    GO TO 3
    ELSE

    GOTO 3
    ENDIF
3   CONTINUE
2   CONTINUE
    WRITE(2,*)'TOTAL VALUES ARE=',M
    CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
    CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
    STOP
    END
C*****************************************************************
    SUBROUTINE SHOWTIME(hour, minute, second, hund)
    INTEGER*2 hour, minute, second, hund, thour
```

```
      CHARACTER*1 ap
      IF (hour .GT. 12) THEN
        ap = 'p'
        thour = hour - 12
      ELSE
        thour = hour
        ap = 'a'
      ENDIF

      WRITE (2, 9002) hour, minute, second, hund
9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')
      END
```

APPENDIX B.5

FORTRAN program for data generation of Alkylation process (a)

```
C     ALKYLATION PROBLEM a
      INCLUDE  'FLIB.FI'
      INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund
      END

      INCLUDE  'FLIB.FD'
      INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund

      OPEN (2,FILE='ALKYa.DAT')
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)

      X1=-50

      DO 2 J=1,40
      X7=89.8

      X1=X1+200

      DO 3 K=1,40
      X8=2.6
      X7=X7+0.2

      DO 4 I=1,40
      X8=X8+0.4

      X4=X1*(1.12+0.13167*X8-0.006667*X8**2)
      X5=1.22*X4-X1
      X2=(X1*X8)-X5
      X6=89+((X7-(86.35+1.098*X8-0.038*X8**2))/(0.325))
      X10=-133+(X7*3)
      X9=35.82-(0.222*X10)
      X3=(0.001*X4*X6*X9)/(98-X6)
      IF((X4.GE.0.01).AND.(X4.LE.5000).AND.(X5.GE.0.01).AND.(X5.LE.200
     +0).AND.(X6.GE.85).AND.(X6.LE.93).AND.(X10.GE.145).AND.(X10.LE.16
     +2).AND.(X9.GE.1.2).AND.(X9.LE.4).AND.(X3.GE.0.01).AND.(X3.LE.120
     +).AND.(X1.GE.0.01).AND.(X1.LE.2000).AND.(X2.GE.0.01).AND.(X2.LE
     +.16000).AND.(X3.GE.0.01).AND.(X3.LE.120).AND.(X7.GE.90).AND.(
     +X7.LE.95).AND.(X8.GE.3).AND.(X8.LE.12))THEN

      Y=0.063*X4*X7-5.04*X1-0.035*X2-10*X3-3.36*X5

      M=M+1
      WRITE(2,10)X1,X7,X8,Y
10     FORMAT(1X,F6.1,1X,F4.1,1X,F4.1,1X,F11.6)
      GO TO 4
      ELSE
      GO TO 4
      ENDIF
4      CONTINUE
3      CONTINUE
2      CONTINUE
      WRITE(2,*)'TOTAL VALUES ARE=',M
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
      STOP
      END
C*******************************************************************
      SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund, thour

      CHARACTER*1 ap

      IF (hour .GT. 12) THEN
        ap = 'p'
        thour = hour - 12
      ELSE
        thour = hour
        ap = 'a'
      ENDIF

      WRITE (2, 9002) hour, minute, second, hund

9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')

      END
```

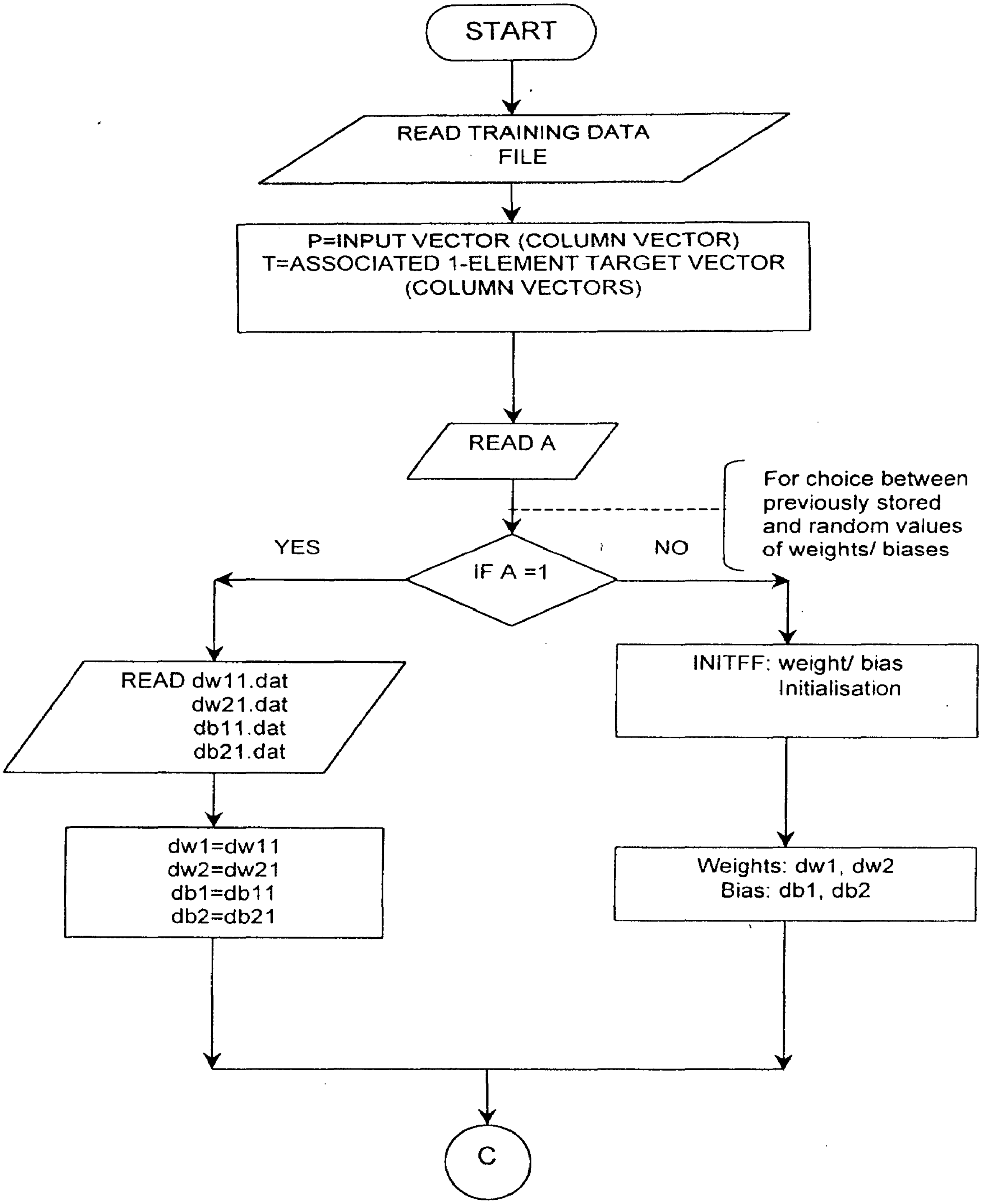
APPENDIX B.6

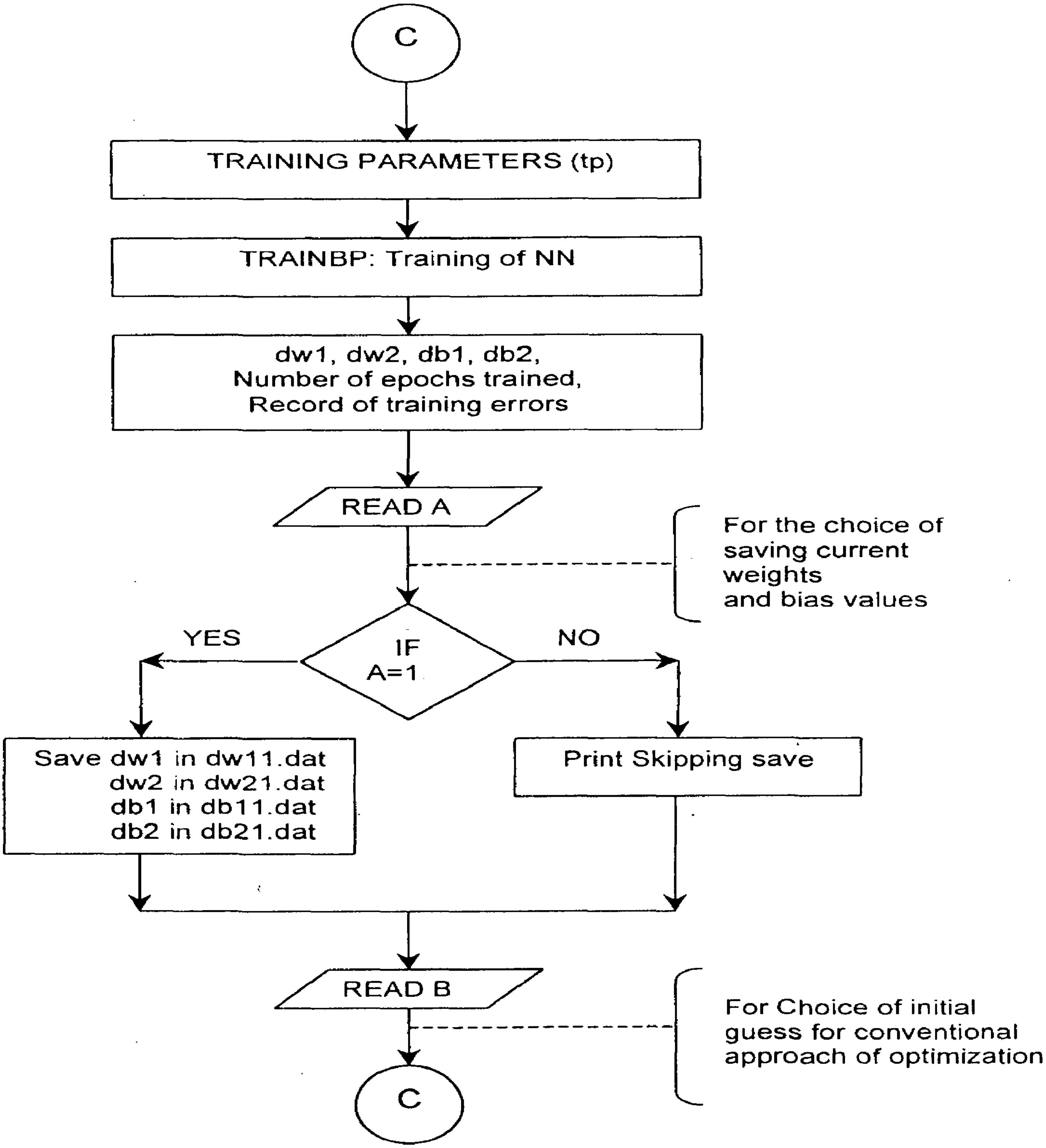
FORTRAN program for data generation of Alkylation problem (b)

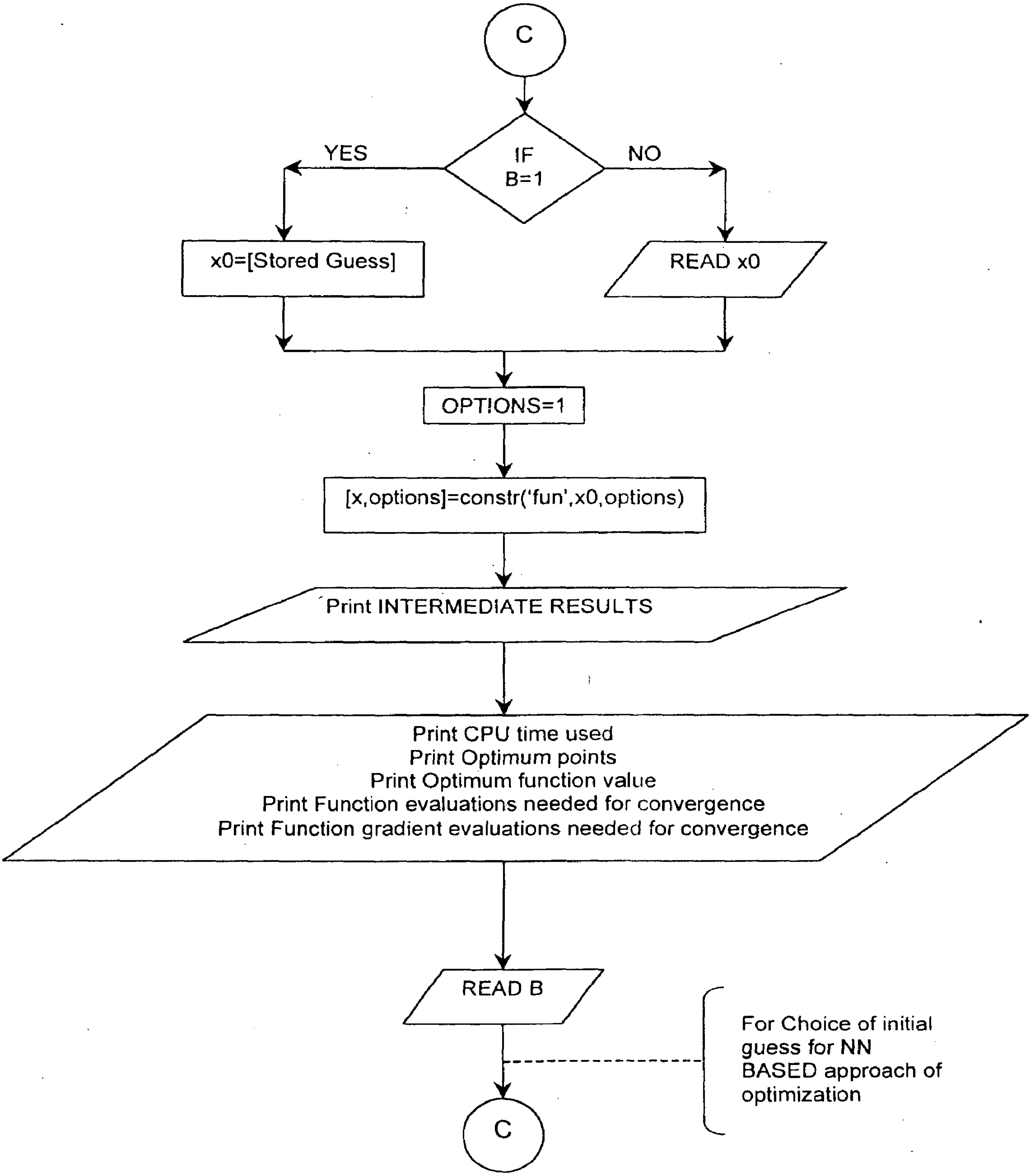
```
C     ALKYLATION PROBLEM B

      INCLUDE  'FLIB.FI'
      INTERFACE TO SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund
      END

      INCLUDE  'FLIB.FD'
      INTEGER*2 tmphour, tmpminute, tmpsecond, tmphund
      INTEGER Z,M,I,J,K,L,N,Q
      OPEN(2,FILE='ALYAB.DAT')
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)

      X4=-1
      DO 2 I=1,10
      X7=0.875
      X4=X4+1

      DO 3 J=1,10
      X8=0
      X7=X7+0.025

      DO 4 K=1,10
      X9=-0.2
      X8=X8+3

      DO 5 L=1,10
      X10=1.365
      X9=X9+1.4

      DO 6 Q=1,10
      X6=0.83
      X10=X10+0.085

      DO 7N=1,10
      X5=-0.5
      X6=X6+0.02

      DO 8Z=1,10
      X5=X5+0.5

      X1=1.22*X4-X5
      X3=(X4*X9*X6*0.01)/(0.98-X6)
      X2=0.1*(-X5+X1*X8)

      IF((X4.GE.0).AND.(X4.LE.5).AND.(X5.GE.0).AND.(X5.LE.2).AND.(X6.GE.
     +0.85).AND.(X6.LE.0.93).AND.(X10.GE.1.45).AND.(X10.LE.1.62).AND.(X9
     +.GE.1.2).AND.(X9.LE.4).AND.(X3.GE.0).AND.(X3.LE.1.2).AND.(X1.GE.0)
     +.AND.(X1.LE.2).AND.(X2.GE.0).AND.(X2.LE.1.6).AND.(X7.GE.0.9).AND.(
     +X7.LE.0.95).AND.(X8.GE.3).AND.(X8.LE.12))THEN

      Y=(6.3*X4*X7-5.04*X1-0.35*X2-X3-3.36*X5)

      M=M+1
      WRITE(2,10)X4,X5,X6,X7,X8,X9,X10,Y
10    FORMAT(1X,F6.3,1X,F6.3,1X,F6.3,1X,F6.3,1X,F6.3,1X,F6.3,1X,F6.3,1X,
     +F14.10)


      GO TO 8
      ELSE
      GO TO 8
      ENDIF
8     CONTINUE
7     CONTINUE
6     CONTINUE
5     CONTINUE
4     CONTINUE
3     CONTINUE
2     CONTINUE
      WRITE(2,*)'TOTAL VALUES=',M
      CALL GETTIM(tmphour, tmpminute, tmpsecond, tmphund)
      CALL SHOWTIME(tmphour, tmpminute, tmpsecond, tmphund)
      STOP
      END
C*******************************************************************
      SUBROUTINE SHOWTIME(hour, minute, second, hund)
      INTEGER*2 hour, minute, second, hund, thour

      CHARACTER*1 ap

      IF (hour .GT. 12) THEN
        ap = 'p'
        thour = hour - 12
      ELSE
        thour = hour
        ap = 'a'
      ENDIF

      WRITE (2, 9002) hour, minute, second, hund

9002  FORMAT (5X,I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')

      END
```

APPENDIX - C

MATLAB PROGRAMS FOR ANN MODELING AND OPTIMIZATION OF PROBLEMS


START
READ TRAINING DATA FILE
P=INPUT VECTOR (COLUMN VECTOR)
T=ASSOCIATED 1-ELEMENT TARGET VECTOR (COLUMN VECTORS)
READ A
For choice between previously stored and random values of weights/ biases
YES
IF A =1
NO
READ dw11.dat
dw21.dat
db11.dat
db21.dat
INITFF: weight/ bias Initialisation
dw1=dw11
dw2=dw21
db1=db11
db2=db21
Weights: dw1, dw2
Bias: db1, db2
C


C
TRAINING PARAMETERS (tp)
TRAINBP: Training of NN
dw1, dw2, db1, db2,
Number of epochs trained,
Record of training errors
READ A
For the choice of
saving current
weights
and bias values
IF
A=1
YES
NO
Save dw1 in dw11.dat
dw2 in dw21.dat
db1 in db11.dat
db2 in db21.dat
Print Skipping save
READ B
For Choice of initial
guess for conventional
approach of optimization
C


C
IF
B=1
YES
NO
x0=[Stored Guess]
READ x0
OPTIONS=1
[x,options]=constr('fun',x0,options)
Print INTERMEDIATE RESULTS
Print CPU time used
Print Optimum points
Print Optimum function value
Print Function evaluations needed for convergence
Print Function gradient evaluations needed for convergence
READ B
For Choice of initial guess for NN BASED approach of optimization
C

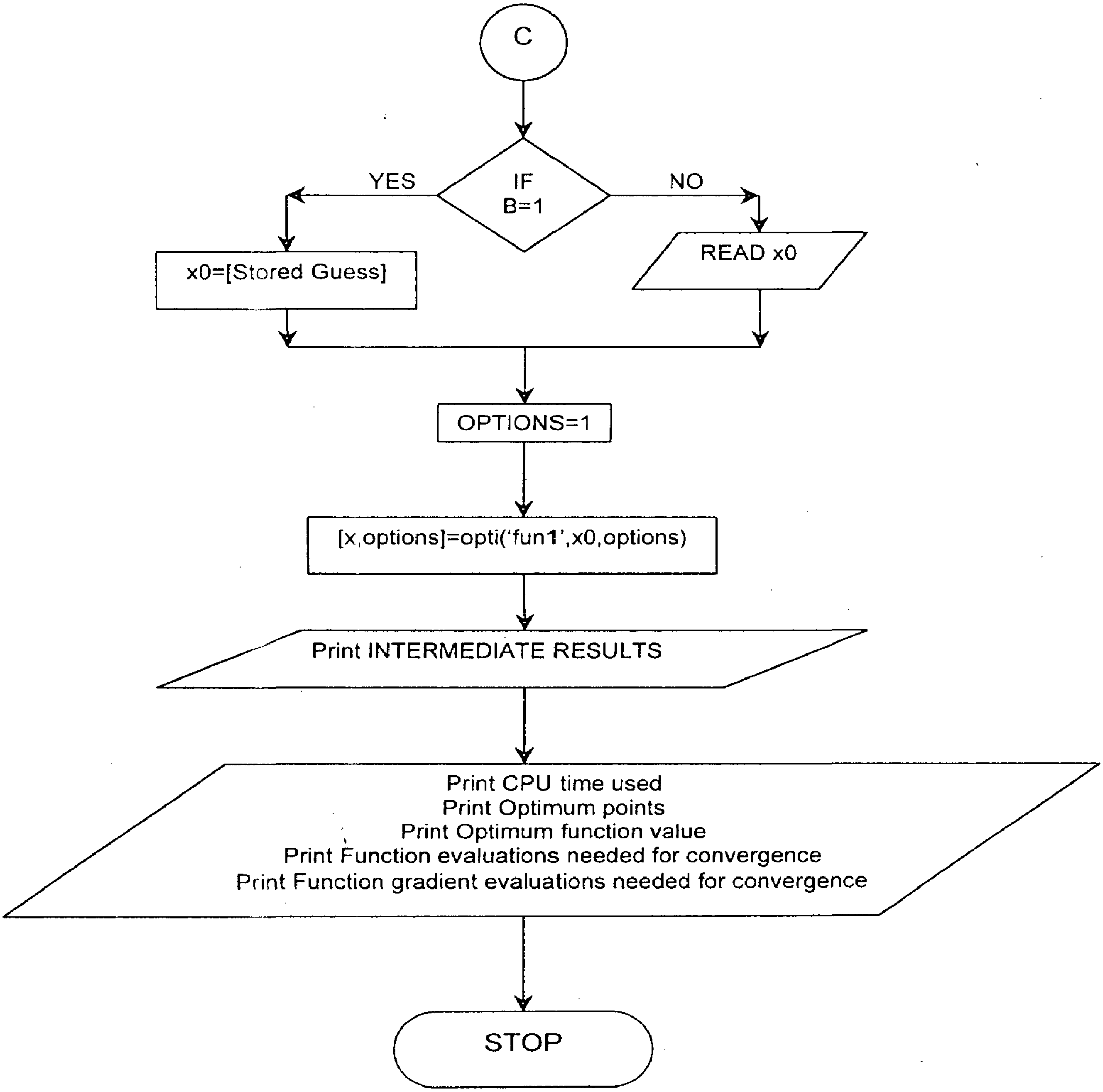

C
IF
B=1
YES
NO
x0=[Stored Guess]
READ x0
OPTIONS=1
[x,options]=opti('fun1',x0,options)
Print INTERMEDIATE RESULTS
Print CPU time used
Print Optimum points
Print Optimum function value
Print Function evaluations needed for convergence
Print Function gradient evaluations needed for convergence
STOP

Fig. C.1: Flow Chart of MATLAB Program for Solving Problem with Neural Network Based Approach As Well as Conventional Approach.

APPENDIX C.1

MATLAB program and function "fun1.m" and "fun.m" for 4 variable problem

```
echo on
%============================================================
%                    MATLAB PROGRAM FOR 4 VARIABLE PROBLEM
%============================================================
%       INITFF      -       Initializes a feed-forward network.
%       TRAINBP     -       Trains a feed-forward network with back propagation.
%       SIMUFF      -       Simulates a feed-forward network.
%       FUNCTION APPROXIMATION WITH TANSIG/TANSIG NETWORK.
%       Using the above functions three-layer network is trained to
%       respond to specific inputs with target outputs.
pause % Strike any key to continue........(Press Ctrl+c to abort)
clc
%       DEFINING A VECTOR ASSOCIATED PROBLEM
%============================================================

load prob4.dat
P1=(prob4(:,1)./10)';
P2=(prob4(:,2)./10)';
P3=(prob4(:,3)./10)';
P4=(prob4(:,4)./10)';
P=[P1;P2;P3;P4];
T=(prob4(:,5)./100.)';
%       P defines input vector (column vectors)
%       T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue...........
clc
%       DESIGN THE NETWORK
%============================================================

%       A two-layer TANSIG/TANSIG network will be trained.
%       The number of hidden TANSIG neurons should reflect
%       the complexity of the problem.

tp=0;
A=0;
A=menu('Click on the choice','1:Prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21);
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;

else
% INITFF is used to initialize the weights and biases for
%       the TANSIG/TANSIG network.
```

```
[dw1,db1,dw2,db2]=initff(P,8,'tansig',T,'tansig');
end
pause % Strike any key to continue..........
clc
%       TRAINING THE NETWORK
%=================================================================
%       TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=100;
%       Frequency of progress displays (in epochs).
max_epoch=1000;
%       Maximum number of epochs to train.
err_goal=0.013;
%       Sum-square error goal.
lr=0.002;
%       Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%       Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
pause % Strike any key to continue..........
clc
%       TRAINING THE NETWORK
%=================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'tansig',dw2,db2,'tansig',P,T,tp);
%       ........and finally finishes.
%       TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator.......
clc
%       USING THE PATTERN ASSOCIATOR
%=================================================================
echo off
%       Let's now test the associator with SIMUFF
[train1]=[T']*100;
[train2]=simuff(P,dw1,db1,'tansig',dw2,db2,'tansig')'*100;
x=[1:344];
h=figure;
figure(gcf)
plot(x,train1,x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end
```

```
echo on
pause % Strike any key to Optimize the problem by Classical Method
clc
%      OPTIMIZATION BY CLASSICAL METHOD
%=================================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[0,0,0,0]; %          Stored Guess
else
disp('##########################################################################')
disp(' ENTER THE GUESS FOR FOUR VARIABLES....x0=[x1,x2,x3,x4];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('##########################################################################')
keyboard
x0=[x0]/10;
end
options=1;
echo on
pause %                    Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                           Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                           Strike Any Key to see Optimum points.......
echo off
n=[1:1:4;x*10];
fprintf('Optimum Point x(%g) is %f. \n',n);
echo on
pause %                           Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                           Strike Any Key to see Total Function & Constrained Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                           Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause % Strike any key to Optimize by Neural Network Method
clc
%      OPTIMIZATION BY NEURAL NETWORK METHOD
%=================================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
```

```
x0=[0,0,0,0]; % Stored Guess
else
disp('####################################################################')
disp(' ENTER THE GUESS FOR FOUR VARIABLES....x0=[x1,x2,x3,x4];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('####################################################################')
keyboard
x0=[x0]/10;
end
options=1;
echo on
pause %                    Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                          Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                          Strike Any Key to see Optimum points.......
echo off
n=[1:1:4;x*10];
fprintf('Optimum Point x(%g) is %f. \n',n);
echo on
pause %                          Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                          Strike Any Key to see Total Function & Constrained
%                                               Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                          Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));

disp('=====================THE END============================')
```

FUNCTION FUN1.M

```
function[g]=fun1(x)
g(1)= ((10*(x(1))).^2)+((10*(x(2))).^2)+((10*(x(3))).^2)+((10*(x(4))).^2)+(10*(x(1)))-(10*(x(2)))+(10*(x(3)))-
(10*(x(4)))-8;
g(2)= ((10*(x(1))).^2)+2*((10*(x(2))).^2)+((10*(x(3))).^2)+2*((10*(x(4))).^2)-(10*(x(1)))-(10*(x(4)))-10;
g(3)= 2*((10*(x(1))).^2)+((10*(x(2))).^2)+((10*(x(3))).^2)+(20*(x(1)))-(10*(x(2)))-(10*(x(4)))-5;
```

FUNCTION FUN.M

```
function[f,g]=fun(x)
f=(10*(x(1))).^2+(10*(x(2))).^2+2*(10*(x(3))).^2+(10*(x(4))).^2-(50*x(1))-(50*x(2))-(210*x(3))+(70*x(4));
g(1)=((10*(x(1))).^2)+((10*(x(2))).^2)+((10*(x(3))).^2)+((10*(x(4))).^2)+(10*(x(1)))-(10*(x(2)))+(10*(x(3)))-
(10*(x(4)))-8;
g(2)=((10*(x(1))).^2)+2*((10*(x(2))).^2)+((10*(x(3))).^2)+2*((10*(x(4))).^2)-(10*(x(1)))-(10*(x(4)))-10;
g(3)= 2*((10*(x(1))).^2)+((10*(x(2))).^2)+((10*(x(3))).^2)+(20*(x(1)))-(10*(x(2)))-(10*(x(4)))-5;
```

APPENDIX C.2

MATLAB program and function “fun1.m” and “fun.m” for Hesse’s function

```
echo on
%==============================================================
% MATLAB PROGRAM FOR 18 MAXIMA PROBLEM (HESSE'S FUNCTION)
%==============================================================
%       INITFF      -       Initializes a feed-forward network.
%       TRAINBP     -       Trains a feed-forward network with back propagation.
%       SIMUFF      -       Simulates a feed-forward network.
%       FUNCTION APPROXIMATION WITH LOGSIG/LOGSIG NETWORK.
%       Using the above functions three-layer network is trained to
%       respond to specific inputs with target outputs.
pause % Strike any key to continue........(Press Ctrl+c to abort)
clc
%       DEFINING A VECTOR ASSOCIATED PROBLEM
%==============================================================
load prob18.dat
P1=(prob18(:,1)./10)';
P2=(prob18(:,2)./10)';
P3=(prob18(:,3)./10)';
P4=(prob18(:,4)./10)';
P5=(prob18(:,5)./10)';
P6=(prob18(:,6)./35)';
P=[P1;P2;P3;P4;P5;P6];
T=(prob18(:,7)./400.)';
%       P defines input vector (column vectors)
%       T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue...........
clc

%       DESIGN THE NETWORK
%==============================================================
%       A two-layer LOGSIG/LOGSIG network will be trained.
%       The number of hidden LOGSIG neurons should reflect
%       the complexity of the problem.
tp=0;
A=0;
A=menu('Click on the choice','1:Prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21)';
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;
else
% INITFF is used to initialize the weights and biases for
%       the LOGSIG/LOGSIG network.
[dw1,db1,dw2,db2]=initff(P,10,'logsig',T,'logsig');
end
```

```
pause % Strike any key to continue..........
clc
%       TRAINING THE NETWORK
%======================================================================
%       TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=100;
%       Frequency of progress displays (in epochs).
max_epoch=5000;
%       Maximum number of epochs to train.
err_goal=0.5;
%       Sum-square error goal.
lr=0.02;
%       Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%       Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
pause % Strike any key to continue..........
clc

%       TRAINING THE NETWORK
%======================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'logsig',dw2,db2,'logsig',P,T,tp);
%       ........and finally finishes.
%       TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator.......
clc

%       USING THE PATTERN ASSOCIATOR
%======================================================================
echo off
%       Let's now test the associator with SIMUFF
[train1]=[T']*400;
[train2]=simuff(P,dw1,db1,'logsig',dw2,db2,'logsig')'*400;
x=[1:800];
h=figure;
figure(gcf)
plot(x,train1,x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end
```

```
echo on
pause % Strike any key to Optimize the problem by Classical Method
clc

%       OPTIMIZATION BY CLASSICAL METHOD
%=====================================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[5,5,5,0,5,10];%      Stored Guess
else
disp('###################################################################')
disp(' ENTER THE GUESS FOR SIX VARIABLES....x0=[x1,x2,x3,x4,x5,x6];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('###################################################################')
keyboard
end
options=1;
echo on
pause %                  Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                        Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                        Strike Any Key to see Optimum points.......
echo off
n=[1:1:6;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                        Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                        Strike Any Key to see Total Function & Constrained Evaluation......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                        Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause % Strike any key to Optimize by Neural Network Method
clc
%       OPTIMIZATION BY NEURAL NETWORK METHOD
%=====================================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[0.5,0.5,0.5,0,0.5,10/35]; % Stored Guess
```

```
else
disp('##########################################################################################')
disp(' ENTER THE GUESS FOR SIX VARIABLES....x0=[x1,x2,x3,x4,x5,x6];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('##########################################################################################')
keyboard
end
options=1;
echo on
pause %                     Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                            Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                            Strike Any Key to see Optimum points.......
echo off
x=[x(1)*10,x(2)*10,x(3)*10,x(4)*10,x(5)*10,x(6)*35];
n=[1:1:6;x];
fprintf('Optimum Point x(%g) is %6.2f. \n',n);
echo on
pause %                            Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                            Strike Any Key to see Total Function & Constrained
%                                                 Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                            Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));

disp('====================THE END===========================')
```

FUNCTION FUN1.M

```
function [g] =fun1(x)
g(1)=x(1)*10+x(2)*10-6;
g(2)=2-x(1)*10-x(2)*10;
g(3)=-x(1)*10+x(2)*10-2;
g(4)=x(1)*10-3*x(2)*10-2;
g(5)=4-(x(3)*10-3).^2-x(4)*10;
g(6)=4-(x(5)*10-3).^2-x(6)*35;
g(7)=-x(1)*10;
g(8)=-x(2)*10;
g(9)=x(3)*10-5;
g(10)=1-x(3)*10;
g(11)=-x(4)*10;
g(12)=x(4)*10-6;
g(13)=1-x(5)*10;
g(14)=x(5)*10-5;
g(15)=-x(6)*35;
g(16)=x(6)*35-10;
```

FUNCTION FUN.M

```
function [f,g] =fun(x)
f= -(25*(x(1)-2).^2+(x(2)-2).^2+(x(3)-1).^2+(x(4)-4).^2+(x(5)-1).^2+(x(6)-4).^2);
g(1)=x(1)+x(2)-6;
g(2)=2-x(1)-x(2);
g(3)=-x(1)+x(2)-2;
g(4)=x(1)-3*x(2)-2;
g(5)=4-(x(3)-3).^2-x(4);
g(6)=4-(x(5)-3).^2-x(6);
g(7)=-x(1);
g(8)=-x(2);
g(9)=x(3)-5;
g(10)=1-x(3);
g(11)=-x(4);
g(12)=x(4)-6;
g(13)=1-x(5);
g(14)=x(5)-5;
g(15)=-x(6);
g(16)=x(6)-10;
```

APPENDIX C.3

MATLAB program and function “fun1.m” and “fun.m” for Optimum fuel allocation

```
echo on
%=============================================================
% MATLAB PROGRAM FOR FUEL ALLOCATION IN POWER PLANT PROBLEM
%=============================================================
%       INITFF          -       Initializes a feed-forward network.
%       TRAINBP         -       Trains a feed-forward network with back propagation.
%       SIMUFF          -       Simulates a feed-forward network.
%       FUNCTION APPROXIMATION WITH LOGSIG/LOGSIG NETWORK.
%       Using the above functions three-layer network is trained to
%       respond to specific inputs with target outputs.
pause % Strike any key to continue........(Press Ctrl+c to abort)
clc

%       DEFINING A VECTOR ASSOCIATED PROBLEM
%=============================================================
load power.dat
P1=(power(:,1)./100)';
P2=(power(:,2)./1.5)';
P3=(power(:,3)./1.5)';
P=[P1;P2;P3];
T=(power(:,4)./35.)';
%       P defines input vector (column vectors)
%       T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue...........
clc

%       DESIGN THE NETWORK
%=============================================================
%       A two-layer LOGSIG/LOGSIG network will be trained.
%       The number of hidden LOGSIG neurons should reflect
%       the complexity of the problem.
tp=0;
A=0;
A=menu('Click on the choice','1:Prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21);
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;
else
% INITFF is used to initialize the weights and biases for
%       the LOGSIG/LOGSIG network.
[dw1,db1,dw2,db2]=initff(P,6,'logsig',T,'logsig');
end
pause % Strike any key to continue...........
clc
```

```
%       TRAINING THE NETWORK
%=====================================================================
%       TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=100;
%       Frequency of progress displays (in epochs).
max_epoch=1000;
%       Maximum number of epochs to train.
err_goal=0.06;
%       Sum-square error goal.
lr=0.1;
%       Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%       Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
pause % Strike any key to continue..........
clc

%       TRAINING THE NETWORK
%=====================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'logsig',dw2,db2,'logsig',P,T,tp);
%       ........and finally finishes.
%       TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator.......
clc

%       USING THE PATTERN ASSOCIATOR
%=====================================================================
echo off
%       Let's now test the associator with SIMUFF
[train1]=[T']*35;
[train2]=simuff(P,dw1,db1,'logsig',dw2,db2,'logsig')'*35;
x=[1:180];
h=figure;
figure(gcf)
plot(x,train1,x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end
echo on
```

```
pause % Strike any key to Optimize the problem by Classical Method
clc

%       OPTIMIZATION BY CLASSICAL METHOD
%==============================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[20,0.5,0.5];%        Stored Guess
else
disp('#####################################################################')
disp(' ENTER THE GUESS FOR THREE VARIABLES....x0=[x1,x2,x3];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('#####################################################################')
keyboard
end
options=1;
echo on
pause %                 Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                        Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                        Strike Any Key to see Optimum points.......
echo off
n=[1:1:3;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                        Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                        Strike Any Key to see Total Function & Constrained Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                        Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause % Strike any key to Optimize by Neural Network Method
clc

%       OPTIMIZATION BY NEURAL NETWORK METHOD
%==============================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[20/100,0.5/1.5,0.5/1.5]; % Stored Guess
```

```
else
disp('######################################################################')
disp(' ENTER THE GUESS FOR THREE VARIABLES....x0=[x1,x2,x3];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('######################################################################')
keyboard
end
options=1;
echo on
pause %                    Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                           Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                           Strike Any Key to see Optimum points.......
echo off
x=[x(1)*100,x(2)*1.5,x(3)*1.5];
n=[1:1:3;x];
fprintf('Optimum Point x(%g) is %f. \n',n);
echo on
pause %                           Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                           Strike Any Key to see Total Function & Constrained
%                                               Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                           Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));

disp('=====================THE END=============================')
```

FUNCTION FUN1.M

```
function[g]=fun1(x)
f1=1.4609+15.186*x(1)+14.5*x(1)*x(1);
f2=1.5742+16.31*x(1)+13.58*x(1)*x(1);
r2=50-(x(1)*100);
g1=0.8008+0.2031*r2+0.000916*r2*r2;
g2=0.7266+0.2256*r2+0.000778*r2*r2;
g(1)=x(1)*100-30;
g(2)=18-x(1)*100;
g(3)=r2-25;
g(4)=14-r2;
g(5)=x(2)*1.5-1;
g(6)=-x(2)*1.5;
g(7)=x(3)*1.5-1;
g(8)=-x(3)*1.5;
g(9)=((1-(x(2)*1.5))*f2+(1-(x(3)*1.5))*g2)-10;
```

FUNCTION FUN.M

```
function[f,g]=fun(x)
f1=1.4609+0.15186*x(1)+0.00145*x(1)*x(1);
f2=1.5742+0.1631*x(1)+0.001358*x(1)*x(1);
r2=50-x(1);
g1=0.8008+0.2031*r2+0.000916*r2*r2;
g2=0.7266+0.2256*r2+0.000778*r2*r2;
f=((x(2)*f1)+(x(3)*g1));
g(1)=x(1)-30;
g(2)=18-x(1);
g(3)=r2-25;
g(4)=14-r2;
g(5)=x(2)-1;
g(6)=-x(2);
g(7)=x(3)-1;
g(8)=-x(3);
g(9)=((1-x(2))*f2+(1-x(3))*g2)-10;
```

APPENDIX C.4

MATLAB program and function “fun1.m” and “fun.m” for Drying problem

```
echo on
%=====================================================================
%                   MATLAB PROGRAM FOR DRYING PROBLEM
%=====================================================================
%      INITFF        -      Initialises a feed-forward network.
%      TRAINBP       -      Trains a feed-forward network with back propagation.
%      SIMUFF        -      Simulates a feed-forward network.
%      FUNCTION APPROXIMATION WITH LOGSIG/LOGSIG NETWORK.
%      Using the above functions three-layer network is trained to
%      respond to specific inputs with target outputs.

pause % Strike any key to continue........(Press Ctrl+c to abort)
clc
%      DEFINING A VECTOR ASSOCIATED PROBLEM
%=====================================================================
load drying.dat
P1=(drying(:,1)./2500)';
P2=(drying(:,2)./1.5)';
P=[P1;P2];
T=(drying(:,3)./400.)';
%      P defines input vector (column vectors)
%      T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue..........
clc
%      DESIGN THE NETWORK
%=====================================================================
%      A two-layer LOGSIG/LOGSIG network will be trained.
%      The number of hidden LOGSIG neurons should reflect
%      the complexity of the problem.
tp=0;
A=0;
A=menu('Click on the choice','1:Prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21);
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;
else
% INITFF is used to initialise the weights and biases for
%      the LOGSIG/LOGSIG network.
[dw11,db11,dw22,db22]=initff(P,4,'logsig',T,'logsig');
dw1=[dw11(:,1)/35. dw11(:,2)/20.];
db1=db11/20.;
dw2=dw22/2.;
db2=db22/3.;
end
```

```
pause % Strike any key to continue..........
clc
%       TRAINING THE NETWORK
%=================================================================
%       TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=1000;
%       Frequency of progress displays (in epochs).
max_epoch=5000;
%       Maximum number of epochs to train.
err_goal=0.005;
%       Sum-square error goal.
lr=0.1;
%       Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%       Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
pause % Strike any key to continue..........
clc
%       TRAINING THE NETWORK
%=================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'logsig',dw2,db2,'logsig',P,T,tp);
%       ........and finally finishes.
%       TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator......
clc
%       USING THE PATTERN ASSOCIATOR
%=================================================================
echo off
%       Let's now test the associator with SIMUFF
[train1]=[T']*400;
[train2]=simuff(P,dw1,db1,'logsig',dw2,db2,'logsig')'*400;
x=[1:82];
h=figure;
figure(gcf)
plot(x,train1,x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end

echo on
```

```
pause % Strike any key to Optimize the problem by Classical Method
clc
%       OPTIMIZATION BY CLASSICAL METHOD
%==============================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[800/2500,0.5/1.5]; %                Stored Guess
else
disp('##########################################################################')
disp(' ENTER THE GUESS FOR TWO VARIABLES....x0=[x1,x2];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('##########################################################################')
keyboard
x0=[x0(:,1)/2500,x0(:,2)/1.5];
end
options=1;
echo on
pause %                 Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                      Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                      Strike Any Key to see Optimum points.......
echo off
x=[x(1)*2500,x(2)*1.5];
n=[1:1:2;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                      Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                      Strike Any Key to see Total Function & Constrained Evaluation.....
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                      Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause % Strike any key to Optimize by Neural Network Method
clc
%       OPTIMIZATION BY NEURAL NETWORK METHOD
%==============================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[800/2500,0.5/1.5]; % Stored Guess
```

```
else
disp('######################################################################')
disp(' ENTER THE GUESS FOR TWO VARIABLES....x0=[x1,x2];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('######################################################################')
keyboard
x0=[x0(:,1)/2500,x0(:,2)/1.5];
end
options=1;
echo on
pause %                    Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                           Strike Any Key to see time used...........
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                           Strike Any Key to see Optimum points.......
echo off
x=[x(1)*2500,x(2)*1.5];
n=[1:1:2;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                           Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                           Strike Any Key to see Total Function & Constrained
%                                              Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                           Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));

disp('====================THE END===========================')
```

FUNCTION FUN1.M

```
function [g] =fun1(x)
r1=(x(2)*1.5)/((x(1)*2500).^0.41);
r2=(1.-exp(-107.9*r1));
r3=(1.-exp(-5.39*r1));
r4=(0.036/r2);
r5=(log(r3/r2))/(r1);
r6=(r4+0.095-((9.27e-4)*r5));
g(1)=-0.2+(4.62e-10)*((x(1)*2500).^2.85)*x(2)*1.5+(1.055e-4)*x(1)*2500;
g(2)=-(1.75/12)+(8.2e-7)*((x(1)*2500).^1.85)*x(2)*1.5;
g(3)=-2+109.6*r1*r6;
g(4)=-x(1)*2500;
g(5)=-x(2)*1.5;
```

FUNCTION FUN.M

```
function[f,g]=fun(x)
r1=(x(2)*1.5)/((x(1)*2500).^0.41);
r2=(1.-exp(-107.9*r1));
r3=(1.-exp(-5.39*r1));
r4=(0.036/r2);
r5=(log(r3/r2))/(r1);
r6=(r4+0.095-((9.27e-4)*r5));
f=-(82.5*x(1))/(r6);
g(1)=-0.2+(4.62e-10)*((x(1)*2500).^2.85)*x(2)*1.5+(1.055e-4)*x(1)*2500;
g(2)=-(1.75/12)+(8.2e-7)*((x(1)*2500).^1.85)*x(2)*1.5;
g(3)=-2+109.6*r1*r6;
```

APPENDIX C.5

MATLAB program and function "fun1.m" and "fun.m" for Alkylation problem (a)

```
echo on
%================================================================
% MATLAB PROGRAM FOR ALKYLATION PROCESS PROBLEM (a)
%================================================================
%       INITFF        -     Initialises a feed-forward network.
%       TRAINBP       -     Trains a feed-forward network with back propagation.
%       SIMUFF        -     Simulates a feed-forward network.
%       FUNCTION APPROXIMATION WITH LOGSIG/LOGSIG NETWORK.
%       Using the above functions three-layer network is trained to
%       respond to specific inputs with target outputs.
pause % Strike any key to continue........(Press Ctrl+c to abort)
clc
%       DEFINING A VECTOR ASSOCIATED PROBLEM
%================================================================
load alkyprob.dat
P1=(alkyprob(:,1)./3500)';
P2=(alkyprob(:,2)./400)';
P3=(alkyprob(:,3)./25)';
P=[P1;P2;P3];
T=(alkyprob(:,4)./2500.)';
%       P defines input vector (column vectors)
%       T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue...........
clc
%       DESIGN THE NETWORK
%================================================================

%       A two-layer LOGSIG/LOGSIG network will be trained.
%       The number of hidden LOGSIG neurons should reflect
%       the complexity of the problem.
tp=0;
A=0;
A=menu('Click on the choice','1:Prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21);
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;
else
% INITFF is used to initialise the weights and biases for
%       the LOGSIG/LOGSIG network.

[dw1,db1,dw2,db2]=initff(P,5,'logsig',T,'logsig');
end
pause % Strike any key to continue...........
clc
```

```
%        TRAINING THE NETWORK
%==================================================================
%        TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=1000;
%        Frequency of progress displays (in epochs).
max_epoch=5000;
%        Maximum number of epochs to train.
err_goal=0.35;
%        Sum-square error goal.
lr=0.02;
%        Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%        Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
pause % Strike any key to continue..........
clc
%        TRAINING THE NETWORK
%==================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'logsig',dw2,db2,'logsig',P,T,tp);
%        ........and finally finishes.
%        TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator.......
clc
%        USING THE PATTERN ASSOCIATOR
%==================================================================
echo off
%        Let's now test the associator with SIMUFF
[train1]=[T']*2500;
[train2]=simuff(P,dw1,db1,'logsig',dw2,db2,'logsig')'*2500;
x=[1:946];
h=figure;
figure(gcf)
plot(x,train1,'o',x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end

echo on
pause % Strike any key to Optimize the problem by Classical Method
clc
```

```
%       OPTIMIZATION BY CLASSICAL METHOD
%=====================================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[0.4938,0.2355,0.4166];%    Stored Guess
else
disp('####################################################################')
disp(' ENTER THE GUESS FOR THREE VARIABLES....x0=[x1,x2,x3];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp(' Range of x1 is 0-2000, x2= 90-95, x3= 3-12')
disp('####################################################################')
keyboard
x0=[x0(:,1)/3500,x0(:,2)/400,x0(:,3)/25];
end
options=1;
echo on
pause %                 Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                         Strike Any Key to see time used............
echo off
fprintf('Time Used %g sec. \n',S);
echo on
pause %                         Strike Any Key to see Optimum points.......
echo off
r4=(x(1)*3500)*(1.12+(0.13167*(x(3)*25))-(0.006667*((x(3)*25).^2)));
r5=(1.22*r4)-(x(1)*3500);
r7=(x(1)*x(3)* 87500)-r5;
r6=89+((((x(2)*400)-(86.35+( 27.45*x(3))-0.038*(25*x(3)).^2)))*(3.076923076923));
r10=-133+(x(2)*1200);
r9=35.82-(0.222*r10);
r8=(0.001*r4*r6*r9)/(98-r6);
y=[x(1)*3500,r7,r8,r4,r5,r6,x(2)*400,x(3)*25,r9,r10];
n=[1:1:10;y];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                         Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                         Strike Any Key to see Total Function & Constrained Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                         Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause % Strike any key to Optimize by Neural Network Method
clc
%       OPTIMIZATION BY NEURAL NETWORK METHOD
```

```
%===============================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[0.4938,0.2355,0.4166]; % Stored Guess
else
disp('##########################################################################')
disp(' ENTER THE GUESS FOR THREE VARIABLES....x0=[x1,x2,x3];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp(' Range of x1 is 0-2000, x2= 90-95, x3= 3-12')
disp('##########################################################################')
keyboard
x0=[x0(:,1)/3500,x0(:,2)/400,x0(:,3)/25];
end
options=1;
echo on
pause %                 Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                     Strike Any Key to see time used...........
echo off
fprintf('Time Used %g. sec \n',S);
echo on
pause %                     Strike Any Key to see Optimum points.......
echo off
r4=(x(1)*3500)*(1.12+(0.13167*(x(3)*25))-(0.006667*((x(3)*25).^2)));
r5=(1.22*r4)-(x(1)*3500);
r7=(x(1)*x(3)* 87500)-r5;
r6=89+((((x(2)*400)-(86.35+( 27.45*x(3))-0.038*(25*x(3)).^2)))*(3.076923076923));
r10=-133+(x(2)*1200);
r9=35.82-(0.222*r10);
r8=(0.001*r4*r6*r9)/(98-r6);
y=[x(1)*3500,r7,r8,r4,r5,r6,x(2)*400,x(3)*25,r9,r10];
n=[1:1:10;y];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                     Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                     Strike Any Key to see Total Function & Constrained
%                                     Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                     Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
disp('====================THE END===========================')
```

FUNCTION FUN1.M

```
function [g]= fun1(x)
g(1)=(x(1)*3500)-2000;
g(2)=(x(2)*400)-95;
g(3)=(x(3)*25)-12;
g(4)=0.01-(x(1)*3500);
g(5)=90-(x(2)*400);
g(6)=3-(x(3)*25);
r4=(x(1)*3500)*(1.12+(0.13167*(x(3)*25))-(0.006667*((x(3)*25).^2)));
r5=(1.22*r4)-(x(1)*3500);
r7=(x(1)*x(3)* 87500)-r5;
r6=89+((((x(2)*400)-(86.35+( 27.45*x(3))-0.038*(25*x(3)).^2)))*(3.076923076923));
r10=-133+(x(2)*1200);
r9=35.82-(0.222*r10);
r8=(0.001*r4*r6*r9)/(98-r6);
g(7)=r7-16000;
g(8)=r8-120;
g(9)=r4-5000;
g(10)=r5-2000;
g(11)=r6-93;
g(12)=r9-4;
g(13)=r10-162;
g(14)=0.01-r7;
g(15)=0.01-r8;
g(16)=0.01-r4;
g(17)=0.01-r5;
g(18)=85-r6;
g(19)=1.2-r9;
g(20)=145-r10;
```

FUNCTION FUN.M

```
function[f,g]= fun(x)
r4=(x(1)*3500)*(1.12+(0.13167*(x(3)*25))-(0.006667*((x(3)*25).^2)));
r5=(1.22*r4)-(x(1)*3500);
r7=(x(1)*x(3)* 87500)-r5;
r6=89+((((x(2)*400)-(86.35+( 27.45*x(3))-0.038*(25*x(3)).^2)))*(3.076923076923));
r10=-133+(x(2)*1200);
r9=35.82-(0.222*r10);
r8=(0.001*r4*r6*r9)/(98-r6);
f=(-0.063*(r4*x(2)*400)+(5.04*(x(1)*3500))+(0.035*(r7))+(10*(r8))+(3.36*r5));
g(1)=(x(1)*3500)-2000;
g(2)=(x(2)*400)-95;
g(3)=(x(3)*25)-12;
g(4)=0.01-(x(1)*3500);
g(5)=90-(x(2)*400);
g(6)=3-(x(3)*25);
g(7)=r7-16000;
g(8)=r8-120;
g(9)=r4-5000;
g(10)=r5-2000;
g(11)=r6-93;
g(12)=r9-4;
g(13)=r10-162;
g(14)=0.01-r7;
g(15)=0.01-r8;
g(16)=0.01-r4;
g(17)=0.01-r5;
g(18)=85-r6;
g(19)=1.2-r9;
g(20)=145-r10;
```

APPENDIX C.6

MATLAB program and function "fun1.m" and "fun.m" for Alkylation problem (b)

```
echo on
%=================================================================
% MATLAB PROGRAM FOR ALKYLATION PROCESS(b) PROBLEM
%=================================================================
%       INITFF        -       Initialises a feed-forward network.
%       TRAINBP       -       Trains a feed-forward network with back propagation.
%       SIMUFF        -       Simulates a feed-forward network.
%       FUNCTION APPROXIMATION WITH TANSIG/TANSIG NETWORK.
%       Using the above functions three-layer network is trained to
%       respond to specific inputs with target outputs.
pause % Strike any key to continue........(Press Ctrl+c to abort)
clc
%       DEFINING A VECTOR ASSOCIATED PROBLEM
%=================================================================
load alyab.dat
P1=(alyab(:,1)./20)';
P2=(alyab(:,2)./8)';
P3=(alyab(:,3)./1.5)';
P4=(alyab(:,4)./1.5)';
P5=(alyab(:,5)./35)';
P6=(alyab(:,6)./10)';
P7=(alyab(:,7)./8)';
P=[P1;P2;P3;P4;P5;P6;P7];
T=(alyab(:,8)./10.)';
%       P defines input vector (column vectors)
%       T defines the associated 1-element targets (column vectors)
pause % Strike any key to continue...........
clc
%       DESIGN THE NETWORK
%=================================================================
%       A two-layer TANSIG/TANSIG network will be trained.
%       The number of hidden TANSIG neurons should reflect
%       the complexity of the problem.
tp=0;
A=0;
A=menu('Click on the choice','1:prevously stored weights','2:Random weights');
if A==1
fprintf('opening files\n');
load('dw11.dat');
dw1=dw11;
load('dw21.dat');
dw2=(dw21);
load('db11.dat');
db1=db11;
load('db21.dat');
db2=db21;
else
% INITFF is used to initialise the weights and biases for
%       the TANSIG/TANSIG network.
[dw1,db1,dw2,db2]=initff(P,9,'tansig',T,'tansig');
end
```

```
pause % Strike any key to continue..........
clc
%	TRAINING THE NETWORK
%=======================================================================
%	TRAINBP uses backpropagation to train feed-forward networks.
disp_freq=1;
%	Frequency of progress displays (in epochs).
max_epoch=1;
%	Maximum number of epochs to train.
err_goal=0.75;
%	Sum-square error goal.
lr=0.001;
%	Learning rate.
mu=0.95;
% Momentum factor.
err_ratio=1.04;
%	Maximum Error Ratio
tp=[disp_freq max_epoch err_goal lr mu err_ratio];
tp1=[disp_freq,max_epoch,err_goal,lr,1.05,0.7,mu,1.04];
pause % Strike any key to continue..........
clc
%	TRAINING THE NETWORK
%=======================================================================
%Training Begins..... please wait (this takes a while!).....
[dw1,db1,dw2,db2,epoc,tr]=trainbp(dw1,db1,'tansig',dw2,db2,'tansig',P,T,tp);
%	........and finally finishes.
%	TRAINBP has returned new weight and bias values, the number of
% epochs trained, and a record of training errors.
pause % Strike any key to use the function approximator......
echo off
clc
%	USING THE PATTERN ASSOCIATOR
%=======================================================================
%	Let's now test the associator with SIMUFF
[train1]=[T']*10.;
[train2]=simuff(P,dw1,db1,'tansig',dw2,db2,'tansig')'*10.;
x=[1:1314];
h=figure;
figure(gcf)
plot(x,train1,x,train2,'x');
xlabel('Data Number');
ylabel('Function Value');
title('Testing of Training Data');
h=legend('Training data','Predicted Value');
A=0;
A=menu('Click on the choice','1:Want to save Weights and Biases','2:Skip Save');
if A==1
fprintf('Saving weights and Biases\n');
save dw11.dat dw1 -ascii
save dw21.dat dw2 -ascii
save db11.dat db1 -ascii
save db21.dat db2 -ascii
else
fprintf('Skipping save\n');
end
```

```
echo on
pause % Strike any key to Optimize the problem by classical method
%       OPTIMIZATION BY CLASSICAL METHOD
%=======================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[3.048/20,1.974/8,0.892/1.5,0.928/1.5,8/35,3.6/10,1.45/8];%  Stored Guess
else
disp('######################################################################')
disp(' ENTER THE GUESS FOR SEVEN VARIABLES....x0=[x1,x2,x3,x4,x5,x6,X7];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('######################################################################')
keyboard
end
options=1;
echo on
pause %                  Strike any key to continue...............
echo off
t=cputime;
[x,options]=constr('fun',x0,options);
S=cputime-t;
echo on
pause %                          Strike Any Key to see time used...........
echo off
fprintf('Time Used %g. sec \n',S);
echo on
pause %                          Strike Any Key to see Optimum points.......
echo off
r1=1.22*(x(1)*20)-(x(2)*8);
r3=((x(1)*20)*(x(6)*10)*(x(3)*1.5)*0.01)/(0.98-(x(3)*1.5));
r2=0.1*(-(x(2)*8)+r1*(x(5)*35));
fprintf('Optimum Point x(1) is %g. \n',r1)
fprintf('Optimum Point x(2) is %g. \n',r2)
fprintf('Optimum Point x(3) is %g. \n',r3)
x=[x(1)*20,x(2)*8,x(3)*1.5,x(4)*1.5,x(5)*35,x(6)*10,x(7)*8];
n=[4:1:10;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                          Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                          Strike Any Key to see Total Function & Constrained Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                          Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
echo on
pause %                          Strike any key to Optimize by Neural Network method
clc
%       OPTIMIZATION BY NEURAL NETWORK METHOD
%=======================================================
echo off
B=0;
B=menu('Click on the choice','1:Prevously stored Guess','2:Input by User');
if B==1
x0=[3.048/20,1.974/8,0.892/1.5,0.928/1.5,8/35,3.6/10,1.45/8];%  Stored Guess
else
disp('#########################################################################')
disp(' ENTER THE GUESS FOR SEVEN VARIABLES....x0=[x1,x2,x3,x4,x5,x6,X7];')
disp(' AND THEN TYPE "return" AND PRESS ENTER.........')
disp('#########################################################################')
keyboard
end
options=1;
echo on
pause %                    Strike any key to continue...............
echo off
t=cputime;
[x,options]=opti('fun1',x0,options);
S=cputime-t;
echo on
pause %                          Strike Any Key to see time used...........
echo off
fprintf('Time Used %g. sec \n',S);
echo on
pause %                          Strike Any Key to see Optimum points.......
echo off
r1=1.22*(x(1)*20)-(x(2)*8);
r3=((x(1)*20)*(x(6)*10)*(x(3)*1.5)*0.01)/(0.98-(x(3)*1.5));
r2=0.1*(-(x(2)*8)+r1*(x(5)*35));
fprintf('Optimum Point x(1) is %g. \n',r1)
fprintf('Optimum Point x(2) is %g. \n',r2)
fprintf('Optimum Point x(3) is %g. \n',r3)
x=[x(1)*20,x(2)*8,x(3)*1.5,x(4)*1.5,x(5)*35,x(6)*10,x(7)*8];
n=[4:1:10;x];
fprintf('Optimum Point x(%g) is %g. \n',n);
echo on
pause %                          Strike Any Key to see Function Value at Optimum point.......
echo off
fprintf('Function Value at Optimum Condition is %g.\n',options(8));
echo on
pause %                          Strike Any Key to see Total Function & Constrained
%                                              Evaluation.......
echo off
fprintf('Functions & Constraint evaluation are %g. \n',options(10));
echo on
pause %                          Strike Any Key to see Function Gradient Evaluation.......
echo off
fprintf('Function Gradient Evaluation are %g. \n',options(11));
disp('====================THE END============================')
```

FUNCTION FUN1.M

```
function[g]=fun1(x)
r1=1.22*(x(1)*20)-(x(2)*8);
r3=((x(1)*20)*(x(6)*10)*(x(3)*1.5)*0.01)/(0.98-(x(3)*1.5));
r2=0.1*(-(x(2)*8)+r1*(x(5)*35));
g(1)=-(r1*(1.12+0.13167*(x(5)*35)-0.0067*(x(5)*35)*(x(5)*35))-(0.99*(x(1)*20)));
g(2)=-(-r1*(1.12+0.13167*(x(5)*35)-0.0067*(x(5)*35)*(x(5)*35))+((x(1)*20)/0.99));
g(3)=-(35.82-(22.2*(x(7)*8))-((x(6)*10)*0.90));
g(4)=-(-1.33+(3*(x(4)*1.5))-(0.99*(x(7)*8)));
g(5)=-(0.8635+(1.098*(x(5)*35)-0.038*(x(5)*35)*(x(5)*35))*0.01+(0.325*((x(3)*1.5)-0.89))-
(0.99*(x(4)*1.5)));
g(6)=-(-0.8635-(1.098*(x(5)*35)-0.038*(x(5)*35)*(x(5)*35))*0.01-(0.325*((x(3)*1.5)-
0.89))+((x(4)*1.5)/0.99));
g(7)=-(-35.82+(22.2*(x(7)*8))+((x(6)*10)/0.90));
g(8)=-(1.33-(3*(x(4)*1.5))+((x(7)*8)/0.99));
g(9)=r1-2.0;
g(10)=-r1;
g(11)=r2-1.6;
g(12)=r3-1.2;
g(13)=(x(1)*20)-5.0;
g(14)=(x(2)*8)-2.0;
g(15)=(x(3)*1.5)-0.93;
g(16)=(x(4)*1.5)-0.95;
g(17)=(x(5)*35)-12;
g(18)=(x(6)*10)-4.0;
g(19)=(x(7)*8)-1.62 ;
g(20)=-r2;
g(21)=-r3;
g(22)=-(x(1)*20);
g(23)=-(x(2)*8);
g(24)=0.85-(x(3)*1.5);
g(25)=0.90-(x(4)*1.5);
g(26)=3.0-(x(5)*35);
g(27)=1.2-(x(6)*10);
g(28)=1.45-(x(7)*8);
```

FUNCTION FUN.M

```
function[f,g]=fun(x)
r1=1.22*(x(1)*20)-(x(2)*8);
r3=((x(1)*20)*(x(6)*10)*(x(3)*1.5)*0.01)/(0.98-(x(3)*1.5));
r2=0.1*(-(x(2)*8)+r1*(x(5)*35));
f=-(6.3*(x(1)*20)*(x(4)*1.5)-5.04*r1-0.35*r2-r3-3.36*(x(2)*8));
g(1)=-(r1*(1.12+0.13167*(x(5)*35)-0.0067*(x(5)*35)*(x(5)*35))-(0.99*(x(1)*20)));
g(2)=-(-r1*(1.12+0.13167*(x(5)*35)-0.0067*(x(5)*35)*(x(5)*35))+((x(1)*20)/0.99));
g(3)=-(35.82-(22.2*(x(7)*8))-((x(6)*10)*0.90));
g(4)=-(-1.33+(3*(x(4)*1.5))-(0.99*(x(7)*8)));
g(5)=-(0.8635+(1.098*(x(5)*35)-0.038*(x(5)*35)*(x(5)*35))*0.01+(0.325*((x(3)*1.5)-0.89))-
(0.99*(x(4)*1.5)));
g(6)=-(-0.8635-(1.098*(x(5)*35)-0.038*(x(5)*35)*(x(5)*35))*0.01-(0.325*((x(3)*1.5)-
0.89))+((x(4)*1.5)/0.99));
g(7)=-(-35.82+(22.2*(x(7)*8))+((x(6)*10)/0.90));
g(8)=-(1.33-(3*(x(4)*1.5))+((x(7)*8)/0.99));
g(9)=r1-2.0;
g(10)=-r1;
g(11)=r2-1.6;
g(12)=r3-1.2;
g(13)=(x(1)*20)-5.0;
g(14)=(x(2)*8)-2.0;
g(15)=(x(3)*1.5)-0.93;
g(16)=(x(4)*1.5)-0.95;
g(17)=(x(5)*35)-12;
g(18)=(x(6)*10)-4.0;
g(19)=(x(7)*8)-1.62 ;
g(20)=-r2;
g(21)=-r3;
g(22)=-(x(1)*20);
g(23)=-(x(2)*8);
g(24)=0.85-(x(3)*1.5);
g(25)=0.90-(x(4)*1.5);
g(26)=3.0-(x(5)*35);
g(27)=1.2-(x(6)*10);
g(28)=1.45-(x(7)*8);
```

Table D.1: Adjusted weight matrix for input layer (4 units) to hidden layer (8 units) (for 4 variable problem)

Hidden Layer units	Input layer units			
	1	2	3	4
1	-7.2616	-8.9554	2.8897	3.8831
2	-9.3332	-6.8924	1.0919	6.3360
3	5.1623	0.7627	7.5767	5.6833
4	5.7514	4.1420	6.7277	-5.6332
5	8.5479	-7.2603	0.3310	-6.6808
6	-3.5277	-2.8760	-7.8827	5.3662
7	0.6230	-10.3869	2.9427	-4.1100
8	13.8146	-2.6171	-2.1605	4.1139

Table D.2: Adjusted weight matrix for hidden layer (8 units) to output layer (1 units) (for 4 variable problem)

Output layer unit	Hidden layer units							
	1	2	3	4	5	6	7	8
1	-0.0353	-0.0313	-0.0761	-0.1343	-0.0182	0.1654	0.0066	0.0347

Table D.3: Adjusted bias matrix for hidden layer (for 4 variable problem)

Hidden layer units	Bias value
1	1.1046
2	1.8986
3	0.0031
4	-1.9971
5	1.0826
6	0.4812
7	0.6258
8	-0.0036

Table D.4: Adjusted bias value for output layer (For 4 variable problem)

Output layer unit	Bias value
1	-0.1344

Table D.5: Adjusted weight matrix for input layer (6 units) to hidden layer (10 units) (for Hesse's function problem)

Hidden Layer units	Input layer units					
	1	2	3	4	5	6
1	-10.9010	1.2932	0.3504	0.3443	-4.4349	-16.3082
2	-13.7342	10.6781	-12.9378	3.1356	4.0007	-0.4914
3	3.9308	-23.9346	3.6283	3.2847	10.8320	15.0740
4	4.8650	-6.5885	-1.3128	6.9200	-7.3495	16.8987
5	19.9516	-22.7759	-0.4526	0.4137	-2.1729	-6.5186
6	-1.7571	-5.1049	11.6700	-4.6492	10.4369	15.3736
7	1.1075	13.4876	9.7812	12.2389	0.0396	0.3157
8	14.7792	5.0481	-7.5213	1.0782	-7.6267	1.1155
9	-12.2568	20.5191	-6.5106	1.7446	-2.5288	-8.1396
10	-11.0808	15.5906	5.6280	0.0351	-3.4985	17.7696

Table D.6: Adjusted weight matrix for hidden layer (10 units) to output layer (1 units) (for Hesse's function problem)

Output Layer unit	Hidden layer units									
	1	2	3	4	5	6	7	8	9	10
1	-1.7635	-0.0415	0.3118	-0.5521	3.1538	0.3062	-0.1085	0.1915	-0.8256	0.5937

Table D.7: Adjusted bias matrix for hidden layer (for Hesse's function problem)

Hidden layer units	Bias value
1	10.5921
2	2.4006
3	-5.7674
4	1.2767
5	-5.5200
6	-3.4385
7	-7.4085
8	-5.5059
9	9.4473
10	-5.9362

Table D.8: Adjusted bias value for output layer (for Hesse's function problem)

Output layer unit	Bias value
1	0.7407

Table D.9: Adjusted weight matrix for input layer (3 units) to hidden layer (6 units) (for Fuel Allocation in power plant problem)

Hidden layer unit	Input layer unit		
	1	2	3
1	-100.8902	0.5189	-13.2959
2	-160.4783	9.1934	-2.3203
3	54.9398	-11.0420	-10.0791
4	74.5500	-14.0014	-3.0299
5	186.4846	0.7532	5.7740
6	-105.5554	10.9678	5.8809

Table D.10: Adjusted weight matrix for hidden layer (6 units) to output layer (1 units) (for Fuel Allocation in power plant problem)

Output Layer unit	Hidden layer units					
	1	2	3	4	5	6
1	-0.1780	0.2012	-0.5890	-0.5099	0.4780	0.6071

Table D.11: Adjusted bias matrix for hidden layer (for Fuel Allocation in power plant problem)

Hidden layer units	Bias value
1	36.1802
2	47.2495
3	-8.6216
4	-19.8385
5	-52.5323
6	20.7927

Table D.12: Adjusted bias value for output layer (for Fuel Allocation in power plant problem)

Output layer unit	Bias value
1	-1.6889

Table D.13: Adjusted weight matrix for input layer (2 units) to hidden layer (4 units) (Drying problem)

Hidden layer unit	Input layer unit	
	1	2
1	-3.0777	-0.1138
2	-3.3325	0.2957
3	1.6203	-6.7736
4	0.8947	0.2648

Table D.14: Adjusted weight matrix for hidden layer (4 units) to output layer (1 units) (Drying problem)

Output Layer unit	Hidden layer units			
	1	2	3	4
1	-3.5226	-2.6822	-4.9675	0.4572

Table D.15: Adjusted bias matrix for hidden layer (Drying problem)

Hidden layer units	Bias value
1	0.4520
2	0.4941
3	-0.7771
4	-0.5929

Table D.16: Adjusted bias value for output layer (Drying problem)

Output layer unit	Bias value
1	1.8891

Table D.17: Adjusted weight matrix for input layer (3 units) to hidden layer (5 units) (Alkylation problem-a)

Hidden layer unit	Input layer unit		
	1	2	3
1	-15.7857	-0.6636e+3	-0.8319
2	-20.9888	0.0699e+3	12.7292
3	-2.0934	0.9303e+3	-38.6535
4	2.6242	-1.5821e+3	-8.7332
5	12.9172	-1.0244e+3	-22.8764

Table D.18: Adjusted weight matrix for hidden layer (5 units) to output layer (1 units) (Alkylation problem-a)

Output Layer unit	Hidden layer units				
	1	2	3	4	5
1	-1.7591	-0.3690	-1.5707	-0.9216	1.5878

Table D.19: Adjusted bias matrix for hidden layer (Alkylation problem-a)

Hidden layer units	Bias value
1	157.9549
2	-13.5502
3	-204.7271
4	376.6418
5	249.0974

Table D.20: Adjusted bias value for output layer (Alkylation problem-a)

Output layer unit	Bias value
1	-1.0375

Table D.21: Adjusted weight matrix for input layer (7 units) to hidden layer (9 units) (Alkylation problem-b)

Hidden layer unit	Input layer unit						
	1	2	3	4	5	6	7
1	-3.5301	-3.4539	20.2903	-13.7154	3.4242	3.5634	-78.6287
2	-6.6908	0.2836	19.3757	-29.9907	1.9799	-0.2671	84.6298
3	4.5133	2.6462	2.4659	27.0034	3.6733	-7.2197	1.5987
4	3.1653	-5.2481	-26.4147	-13.3271	1.6172	-4.0528	4.0039
5	8.0725	-1.3212	10.8953	11.2567	-4.4654	-2.6333	-47.0823
6	-1.8616	-4.2597	-5.2523	18.1946	1.2996	-5.7603	109.6978
7	0.3866	-1.0082	14.4858	42.5961	4.2909	-0.3291	-1.6735
8	5.4938	1.6637	25.4676	-10.0318	-1.9142	7.0847	-54.6507
9	-7.1871	0.8295	14.8500	-17.6261	-0.6424	6.5464	-88.5228

Table D.22: Adjusted weight matrix for hidden layer (9 units) to output layer (1 units) (Alkylation problem-b)

Output Layer unit	Hidden layer units					
	1	2	3	4	5	6
1	-0.1050	-0.0084	0.0835	-0.2054	-0.0693	-0.1182
Output Layer unit	Hidden layer units					
	7		8		9	
1	-0.0582		-0.2316		0.1055	

Table D.23: Adjusted bias matrix for hidden layer (Alkylation problem-b)

Hidden layer units	Bias value
1	10.0614
2	-9.1016
3	-17.8396
4	24.7017
5	-2.9562
6	-26.1071
7	-35.6256
8	-1.4731
9	17.0679

Table D.24: Adjusted bias value for output layer (Alkylation problem-b)

Output layer unit	Bias value
1	0.0810

Table E.1: Training data with corresponding target and calculated outputs by ANN model for 4 variable problem

Input variables				Target	Calculated
x_1	x_2	x_3	x_4	T	Y
-0.5000	-0.5000	0	-1.5000	-2.7500	-4.8448
-0.5000	-0.5000	0	-1.0000	-0.5000	-0.9320
-0.5000	-0.5000	0.5000	-1.5000	-12.7500	-13.3511
-0.5000	-0.5000	0.5000	-1.0000	-10.5000	-9.9372
-0.5000	-0.5000	0.5000	-0.5000	-7.7500	-6.7439
-0.5000	-0.5000	0.5000	0	-4.5000	-4.0323
-0.5000	-0.5000	0.5000	0.5000	-0.7500	-1.8308
-0.5000	-0.5000	1.0000	-1.5000	-21.7500	-21.5535
-0.5000	-0.5000	1.0000	-1.0000	-19.5000	-19.3019
-0.5000	-0.5000	1.0000	-0.5000	-16.7500	-16.6386
-0.5000	-0.5000	1.0000	0	-13.5000	-13.3555
-0.5000	-0.5000	1.0000	0.5000	-9.7500	-9.6232
-0.5000	-0.5000	1.5000	-1.5000	-29.7500	-29.2344
-0.5000	-0.5000	1.5000	-1.0000	-27.5000	-27.5130
-0.5000	-0.5000	1.5000	-0.5000	-24.7500	-25.2749
-0.5000	-0.5000	1.5000	0	-21.5000	-22.0731
-0.5000	-0.5000	1.5000	0.5000	-17.7500	-17.8259
-0.5000	-0.5000	2.0000	-0.5000	-31.7500	-31.8236
-0.5000	-0.5000	2.0000	0	-28.5000	-28.8331
-0.5000	-0.5000	2.0000	0.5000	-24.7500	-25.0151
-0.5000	0	0	-1.5000	-5.5000	-7.1583
-0.5000	0	0	-1.0000	-3.2500	-3.2981
-0.5000	0	0	-0.5000	-0.5000	-0.0814
-0.5000	0	0.5000	-1.5000	-15.5000	-15.7300
-0.5000	0	0.5000	-1.0000	-13.2500	-12.6385
-0.5000	0	0.5000	-0.5000	-10.5000	-9.5104
-0.5000	0	0.5000	0	-7.2500	-6.5124
-0.5000	0	0.5000	0.5000	-3.5000	-3.7780
-0.5000	0	1.0000	-1.5000	-24.5000	-23.8928
-0.5000	0	1.0000	-1.0000	-22.2500	-21.8462
-0.5000	0	1.0000	-0.5000	-19.5000	-19.3757
-0.5000	0	1.0000	0	-16.2500	-16.1078
-0.5000	0	1.0000	0.5000	-12.5000	-12.1082
-0.5000	0	1.5000	-1.5000	-32.5000	-31.6953
-0.5000	0	1.5000	-1.0000	-30.2500	-29.8876
-0.5000	0	1.5000	-0.5000	-27.5000	-27.6333
-0.5000	0	1.5000	0	-24.2500	-24.5132
-0.5000	0	1.5000	0.5000	-20.5000	-20.3331
-0.5000	0	2.0000	-1.0000	-37.2500	-36.6690
-0.5000	0	2.0000	-0.5000	-34.5000	-34.0087
-0.5000	0	2.0000	0	-31.2500	-30.8951
-0.5000	0	2.0000	0.5000	-27.5000	-27.1260
-0.5000	0.5000	0	-1.5000	-7.7500	-8.9655
-0.5000	0.5000	0	-1.0000	-5.5000	-5.3560
-0.5000	0.5000	0	-0.5000	-2.7500	-2.0662
-0.5000	0.5000	0.5000	-1.5000	-17.7500	-17.6209
-0.5000	0.5000	0.5000	-1.0000	-15.5000	-14.9342
-0.5000	0.5000	0.5000	-0.5000	-12.7500	-12.0239
-0.5000	0.5000	0.5000	0	-9.5000	-8.9137
-0.5000	0.5000	0.5000	0.5000	-5.7500	-5.8050
-0.5000	0.5000	1.0000	-1.5000	-26.7500	-25.9214
-0.5000	0.5000	1.0000	-1.0000	-24.5000	-24.0655
-0.5000	0.5000	1.0000	-0.5000	-21.7500	-21.7840
-0.5000	0.5000	1.0000	0	-18.5000	-18.6299
-0.5000	0.5000	1.0000	0.5000	-14.7500	-14.5800
-0.5000	0.5000	1.5000	-1.5000	-34.7500	-33.9301
-0.5000	0.5000	1.5000	-1.0000	-32.5000	-32.1131
-0.5000	0.5000	1.5000	-0.5000	-29.7500	-29.7882
-0.5000	0.5000	1.5000	0	-26.5000	-26.6993
-0.5000	0.5000	1.5000	0.5000	-22.7500	-22.6744
-0.5000	0.5000	2.0000	-1.0000	-39.5000	-38.9603
-0.5000	0.5000	2.0000	-0.5000	-36.7500	-36.1496
-0.5000	0.5000	2.0000	0	-33.5000	-32.8570
-0.5000	0.5000	2.0000	0.5000	-29.7500	-29.0973
-0.5000	1.0000	0	-1.5000	-9.5000	-10.2219
-0.5000	1.0000	0	-1.0000	-7.2500	-6.8645
-0.5000	1.0000	0	-0.5000	-4.5000	-3.6058
-0.5000	1.0000	0	0	-1.2500	-0.8955
-0.5000	1.0000	0.5000	-1.5000	-19.5000	-18.9608
-0.5000	1.0000	0.5000	-1.0000	-17.2500	-16.6194
-0.5000	1.0000	0.5000	-0.5000	-14.5000	-14.0015

Table E.1 contd.

-0.5000	1.0000	0.5000	0	-11.2500	-10.9907
-0.5000	1.0000	0.5000	0.5000	-7.5000	-7.7647
-0.5000	1.0000	1.0000	-1.5000	-28.5000	-27.4850
-0.5000	1.0000	1.0000	-1.0000	-26.2500	-25.8117
-0.5000	1.0000	1.0000	-0.5000	-23.5000	-23.7033
-0.5000	1.0000	1.0000	0	-20.2500	-20.7657
-0.5000	1.0000	1.0000	0.5000	-16.5000	-16.9132
-0.5000	1.0000	1.5000	-1.0000	-34.2500	-33.9937
-0.5000	1.0000	1.5000	-0.5000	-31.5000	-31.6515
-0.5000	1.0000	1.5000	0	-28.2500	-28.6142
-0.5000	1.0000	1.5000	0.5000	-24.5000	-24.8447
-0.5000	1.0000	2.0000	-1.0000	-41.2500	-40.8367
-0.5000	1.0000	2.0000	-0.5000	-38.5000	-38.1050
-0.5000	1.0000	2.0000	0	-35.2500	-34.7455
-0.5000	1.0000	2.0000	0.5000	-31.5000	-31.0472
-0.5000	1.5000	0	-1.0000	-8.5000	-7.8725
-0.5000	1.5000	0	-0.5000	-5.7500	-4.6256
-0.5000	1.5000	0	0	-2.5000	-1.8651
-0.5000	1.5000	0.5000	-1.0000	-18.5000	-17.7059
-0.5000	1.5000	0.5000	-0.5000	-15.7500	-15.2962
-0.5000	1.5000	0.5000	0	-12.5000	-12.5387
-0.5000	1.5000	0.5000	0.5000	-8.7500	-9.4402
-0.5000	1.5000	1.0000	-1.0000	-27.5000	-27.0133
-0.5000	1.5000	1.0000	-0.5000	-24.7500	-25.0082
-0.5000	1.5000	1.0000	0	-21.5000	-22.3471
-0.5000	1.5000	1.0000	0.5000	-17.7500	-18.8726
-0.5000	1.5000	1.5000	-0.5000	-32.7500	-33.0621
-0.5000	1.5000	1.5000	0	-29.5000	-30.1627
-0.5000	1.5000	1.5000	0.5000	-25.7500	-26.7083
-0.5000	1.5000	2.0000	0	-36.5000	-36.4472
-0.5000	1.5000	2.0000	0.5000	-32.7500	-32.9135
0	-0.5000	0	-1.5000	-5.5000	-7.2797
0	-0.5000	0	-1.0000	-3.2500	-3.3337
0	-0.5000	0	-0.5000	-0.5000	-0.0659
0	-0.5000	0.5000	-1.5000	-15.5000	-16.5337
0	-0.5000	0.5000	-1.0000	-13.2500	-13.4033
0	-0.5000	0.5000	-0.5000	-10.5000	-10.1011
0	-0.5000	0.5000	0	-7.2500	-6.7571
0	-0.5000	0.5000	0.5000	-3.5000	-3.6199
0	-0.5000	1.0000	-1.5000	-24.5000	-25.3053
0	-0.5000	1.0000	-1.0000	-22.2500	-22.9815
0	-0.5000	1.0000	-0.5000	-19.5000	-20.0431
0	-0.5000	1.0000	0	-16.2500	-16.2334
0	-0.5000	1.0000	0.5000	-12.5000	-11.7699
0	-0.5000	1.5000	-1.5000	-32.5000	-33.3693
0	-0.5000	1.5000	-1.0000	-30.2500	-31.0136
0	-0.5000	1.5000	-0.5000	-27.5000	-28.1007
0	-0.5000	1.5000	0	-24.2500	-24.4148
0	-0.5000	1.5000	0.5000	-20.5000	-19.8622
0	-0.5000	2.0000	0	-31.2500	-30.7477
0	-0.5000	2.0000	0.5000	-27.5000	-26.6471
0	0	0	-1.5000	-8.2500	-9.4634
0	0	0	-1.0000	-6.0000	-5.8692
0	0	0	-0.5000	-3.2500	-2.6035
0	0	0	0	0	-0.0364
0	0	0.5000	-1.5000	-18.2500	-18.8087
0	0	0.5000	-1.0000	-16.0000	-16.1356
0	0	0.5000	-0.5000	-13.2500	-13.1018
0	0	0.5000	0	-10.0000	-9.6634
0	0	0.5000	0.5000	-6.2500	-6.0582
0	0	1.0000	-1.5000	-27.2500	-27.6872
0	0	1.0000	-1.0000	-25.0000	-25.5948
0	0	1.0000	-0.5000	-22.2500	-22.8828
0	0	1.0000	0	-19.0000	-19.2234
0	0	1.0000	0.5000	-15.2500	-14.6648
0	0	1.5000	-1.5000	-35.2500	-35.8346
0	0	1.5000	-1.0000	-33.0000	-33.5867
0	0	1.5000	-0.5000	-30.2500	-30.6947
0	0	1.5000	0	-27.0000	-27.0919
0	0	1.5000	0.5000	-23.2500	-22.6501
0	0	2.0000	-1.0000	-40.0000	-40.1762
0	0	2.0000	-0.5000	-37.2500	-36.9489
0	0	2.0000	0	-34.0000	-33.2479
0	0	2.0000	0.5000	-30.2500	-29.1234
0	0.5000	0	-1.5000	-10.5000	-11.0613

Table E.1 contd.

0	0.5000	0	-1.0000	-8.2500	-7.9126
0	0.5000	0	-0.5000	-5.5000	-4.8373
0	0.5000	0	0	-2.2500	-2.0839
0	0.5000	0.5000	-1.5000	-20.5000	-20.5104
0	0.5000	0.5000	-1.0000	-18.2500	-18.3108
0	0.5000	0.5000	-0.5000	-15.5000	-15.6684
0	0.5000	0.5000	0	-12.2500	-12.3311
0	0.5000	0.5000	0.5000	-8.5000	-8.4389
0	0.5000	1.0000	-1.5000	-29.5000	-29.5584
0	0.5000	1.0000	-1.0000	-27.2500	-27.7743
0	0.5000	1.0000	-0.5000	-24.5000	-25.3172
0	0.5000	1.0000	0	-21.2500	-21.8765
0	0.5000	1.0000	0.5000	-17.5000	-17.3805
0	0.5000	1.5000	-1.5000	-37.5000	-37.6929
0	0.5000	1.5000	-1.0000	-35.2500	-35.8090
0	0.5000	1.5000	-0.5000	-32.5000	-33.0447
0	0.5000	1.5000	0	-29.2500	-29.5195
0	0.5000	1.5000	0.5000	-25.5000	-25.1940
0	0.5000	2.0000	-1.0000	-42.2500	-42.2669
0	0.5000	2.0000	-0.5000	-39.5000	-39.3252
0	0.5000	2.0000	0	-36.2500	-35.6457
0	0.5000	2.0000	0.5000	-32.5000	-31.4719
0	1.0000	-0.5000	-1.5000	-1.2500	-2.8800
0	1.0000	0	-1.5000	-12.2500	-12.4719
0	*1.0000*	*0*	*-1.0000*	*-10.0000*	*-9.5360*
0	1.0000	0	-0.5000	-7.2500	-6.5792
0	1.0000	0	0	-4.0000	-3.6590
0	1.0000	0	0.5000	-0.2500	-0.9562
0	1.0000	0.5000	-1.5000	-22.2500	-21.9035
0	1.0000	0.5000	-1.0000	-20.0000	-19.9590
0	1.0000	0.5000	-0.5000	-17.2500	-17.6001
0	1.0000	0.5000	0	-14.0000	-14.4156
0	1.0000	0.5000	0.5000	-10.2500	-10.4270
0	1.0000	1.0000	-1.5000	-31.2500	-30.9291
0	1.0000	1.0000	-1.0000	-29.0000	-29.4408
0	1.0000	1.0000	-0.5000	-26.2500	-27.1870
0	1.0000	1.0000	0	-23.0000	-23.9457
0	1.0000	1.0000	0.5000	-19.2500	-19.6341
0	1.0000	1.5000	-1.0000	-37.0000	-37.4275
0	1.0000	1.5000	-0.5000	-34.2500	-34.9419
0	1.0000	1.5000	0	-31.0000	-31.5351
0	1.0000	1.5000	0.5000	-27.2500	-27.3520
0	1.0000	2.0000	-1.0000	-44.0000	-43.5647
0	1.0000	2.0000	-0.5000	-41.2500	-41.1855
0	1.0000	2.0000	0	-38.0000	-37.7430
0	1.0000	2.0000	0.5000	-34.2500	-33.6226
0	1.5000	-0.5000	-1.0000	-0.2500	-0.5913
0	1.5000	0	-1.0000	-11.2500	-11.0761
0	1.5000	0	-0.5000	-8.5000	-7.9770
0	1.5000	0	0	-5.2500	-4.8203
0	1.5000	0	0.5000	-1.5000	-1.7863
0	1.5000	0.5000	-1.0000	-21.2500	-21.4075
0	1.5000	0.5000	-0.5000	-18.5000	-18.9960
0	1.5000	0.5000	0	-15.2500	-15.8384
0	1.5000	0.5000	0.5000	-11.5000	-11.8887
0	1.5000	1.0000	-1.0000	-30.2500	-30.7489
0	1.5000	1.0000	-0.5000	-27.5000	-28.4975
0	1.5000	1.0000	0	-24.2500	-25.3181
0	1.5000	1.0000	0.5000	-20.5000	-21.2454
0	1.5000	1.5000	-1.0000	-38.2500	-38.3909
0	1.5000	1.5000	-0.5000	-35.5000	-36.2067
0	1.5000	1.5000	0	-32.2500	-32.9549
0	1.5000	1.5000	0.5000	-28.5000	-28.9734
0	1.5000	*2.0000*	*0*	-39.2500	-39.2307
0	1.5000	2.0000	0.5000	-35.5000	-35.3876
0.5000	-0.5000	0	-1.5000	-7.7500	-8.7497
0.5000	-0.5000	0	-1.0000	-5.5000	-5.3584
0.5000	-0.5000	0	-0.5000	-2.7500	-2.3673
0.5000	-0.5000	0.5000	-1.5000	-17.7500	-18.7548
0.5000	-0.5000	0.5000	-1.0000	-15.5000	-16.0225
0.5000	-0.5000	0.5000	-0.5000	-12.7500	-12.8968
0.5000	-0.5000	0.5000	0	-9.5000	-9.3598
0.5000	-0.5000	0.5000	0.5000	-5.7500	-5.7488
0.5000	-0.5000	1.0000	-1.5000	-26.7500	-28.0761
0.5000	-0.5000	1.0000	-1.0000	-24.5000	-25.5207

Table E.1 contd.

0.5000	-0.5000	1.0000	-0.5000	-21.7500	-22.3546
0.5000	-0.5000	1.0000	0	-18.5000	-18.4457
0.5000	-0.5000	1.0000	0.5000	-14.7500	-13.9234
0.5000	-0.5000	1.5000	-0.5000	-29.7500	-29.8945
0.5000	-0.5000	1.5000	0	-26.5000	-25.9836
0.5000	-0.5000	1.5000	0.5000	-22.7500	-21.5890
0.5000	0	0	-1.5000	-10.5000	-10.5694
0.5000	0	0	-1.0000	-8.2500	-7.5518
0.5000	0	0	-0.5000	-5.5000	-4.6871
0.5000	0	0	0	-2.2500	-2.1285
0.5000	0	0.5000	-1.5000	-20.5000	-20.7557
0.5000	0	0.5000	-1.0000	-18.2500	-18.4008
0.5000	0	0.5000	-0.5000	-15.5000	-15.5589
0.5000	0	0.5000	0	-12.2500	-12.0728
0.5000	0	0.5000	0.5000	-8.5000	-8.1926
0.5000	0	1.0000	-1.5000	-29.5000	-30.2166
0.5000	0	1.0000	-1.0000	-27.2500	-27.9427
0.5000	0	1.0000	-0.5000	-24.5000	-24.9542
0.5000	0	1.0000	0	-21.2500	-21.1970
0.5000	0	1.0000	0.5000	-17.5000	-16.7177
0.5000	0	1.5000	-1.0000	-35.2500	-35.8128
0.5000	0	1.5000	-0.5000	-32.5000	-32.4957
0.5000	0	1.5000	0	-29.2500	-28.6187
0.5000	0	1.5000	0.5000	-25.5000	-24.2786
0.5000	0	2.0000	0.5000	-32.5000	-30.5233
0.5000	0.5000	-0.5000	-1.5000	-1.7500	-1.7930
0.5000	0.5000	0	-1.5000	-12.7500	-12.1902
0.5000	0.5000	0	-1.0000	-10.5000	-9.3741
0.5000	0.5000	0	-0.5000	-7.7500	-6.6620
0.5000	0.5000	0	0	-4.5000	-3.9868
0.5000	0.5000	0	0.5000	-0.7500	-1.4536
0.5000	0.5000	0.5000	-1.5000	-22.7500	-22.3906
0.5000	0.5000	0.5000	-1.0000	-20.5000	-20.2910
0.5000	0.5000	0.5000	-0.5000	-17.7500	-17.7326
0.5000	0.5000	0.5000	0	-14.5000	-14.4217
0.5000	0.5000	0.5000	0.5000	-10.7500	-10.4408
0.5000	0.5000	1.0000	-1.5000	-31.7500	-31.7501
0.5000	0.5000	1.0000	-1.0000	-29.5000	-29.8535
0.5000	0.5000	1.0000	-0.5000	-26.7500	-27.1190
0.5000	0.5000	1.0000	0	-23.5000	-23.5738
0.5000	0.5000	1.0000	0.5000	-19.7500	-19.2353
0.5000	0.5000	1.5000	-1.5000	-39.7500	-39.3528
0.5000	0.5000	1.5000	-1.0000	-37.5000	-37.6190
0.5000	0.5000	1.5000	-0.5000	-34.7500	-34.7136
0.5000	0.5000	1.5000	0	-31.5000	-31.0099
0.5000	0.5000	1.5000	0.5000	-27.7500	-26.7508
0.5000	0.5000	2.0000	0.5000	-34.7500	-33.0417
0.5000	1.0000	-0.5000	-1.5000	-3.5000	-4.0376
0.5000	1.0000	-0.5000	-1.0000	-1.2500	-0.1156
0.5000	1.0000	0	-1.5000	-14.5000	-14.3056
0.5000	1.0000	0	-1.0000	-12.2500	-11.3302
0.5000	1.0000	0	-0.5000	-9.5000	-8.4958
0.5000	1.0000	0	0	-6.2500	-5.5388
0.5000	1.0000	0	0.5000	-2.5000	-2.4860
0.5000	1.0000	0.5000	-1.5000	-24.5000	-24.2320
0.5000	1.0000	0.5000	-1.0000	-22.2500	-22.1085
0.5000	1.0000	0.5000	-0.5000	-19.5000	-19.5624
0.5000	1.0000	0.5000	0	-16.2500	-16.2680
0.5000	1.0000	0.5000	0.5000	-12.5000	-12.1686
0.5000	1.0000	1.0000	-1.5000	-33.5000	-33.0432
0.5000	1.0000	1.0000	-1.0000	-31.2500	-31.4277
0.5000	1.0000	1.0000	-0.5000	-28.5000	-28.8663
0.5000	1.0000	1.0000	0	-25.2500	-25.4341
0.5000	1.0000	1.0000	0.5000	-21.5000	-21.1841
0.5000	1.0000	1.5000	-1.0000	-39.2500	-38.7335
0.5000	1.0000	1.5000	-0.5000	-36.5000	-36.3596
0.5000	1.0000	1.5000	0	-33.2500	-32.9419
0.5000	1.0000	1.5000	0.5000	-29.5000	-28.7806
0.5000	1.0000	2.0000	0.5000	-36.5000	-35.2200
0.5000	1.5000	-0.5000	-1.0000	-2.5000	-2.5842
0.5000	1.5000	-0.5000	-0.5000	0.2500	1.2180
0.5000	1.5000	0	-1.0000	-13.5000	-13.7929
0.5000	1.5000	0	-0.5000	-10.7500	-10.5067
0.5000	1.5000	0	0	-7.5000	-7.0327
0.5000	1.5000	0	0.5000	-3.7500	-3.3454

Table E.1 contd.

0.5000	1.5000	0.5000	-1.0000	-23.5000	-24.2745
0.5000	1.5000	0.5000	-0.5000	-20.7500	-21.4077
0.5000	1.5000	0.5000	0	-17.5000	-17.8098
0.5000	1.5000	0.5000	0.5000	-13.7500	-13.4431
0.5000	1.5000	1.0000	-1.0000	-32.5000	-33.0212
0.5000	1.5000	1.0000	-0.5000	-29.7500	-30.4378
0.5000	1.5000	1.0000	0	-26.5000	-26.8794
0.5000	1.5000	1.0000	0.5000	-22.7500	-22.5331
0.5000	1.5000	1.5000	-0.5000	-37.7500	-37.4969
0.5000	1.5000	1.5000	0	-34.5000	-34.3103
0.5000	1.5000	1.5000	0.5000	-30.7500	-30.2137
1.0000	-0.5000	0	0	-1.2500	-3.1937
1.0000	-0.5000	0.5000	0	-11.2500	-12.6490
1.0000	-0.5000	0.5000	0.5000	-7.5000	-8.7267
1.0000	0	0	-1.0000	-10.0000	-9.7387
1.0000	0	0	-0.5000	-7.2500	-7.4177
1.0000	0	0	0	-4.0000	-4.9073
1.0000	0	0	0.5000	-0.2500	-2.3977
1.0000	0	0.5000	-0.5000	-17.2500	-18.0836
1.0000	0	0.5000	0	-14.0000	-14.8213
1.0000	0	0.5000	0.5000	-10.2500	-10.9438
1.0000	0	1.0000	0	-23.0000	-23.4771
1.0000	0	1.0000	0.5000	-19.2500	-19.3559
1.0000	0.5000	-0.5000	-1.0000	-1.2500	0.1152
1.0000	0.5000	0	-1.0000	-12.2500	-11.4279
1.0000	0.5000	0	-0.5000	-9.5000	-8.9986
1.0000	0.5000	0	0	-6.2500	-6.3784
1.0000	0.5000	0	0.5000	-2.5000	-3.6042
1.0000	0.5000	0.5000	-1.0000	-22.2500	-22.1278
1.0000	0.5000	0.5000	-0.5000	-19.5000	-19.6617
1.0000	0.5000	0.5000	0	-16.2500	-16.5594
1.0000	0.5000	0.5000	0.5000	-12.5000	-12.7952
1.0000	0.5000	1.0000	0	-25.2500	-25.2573
1.0000	0.5000	1.0000	0.5000	-21.5000	-21.3970
1.0000	1.0000	-0.5000	-0.5000	-0.2500	0.6594
1.0000	1.0000	0	-1.0000	-14.0000	-13.8808
1.0000	1.0000	0	-0.5000	-11.2500	-10.9901
1.0000	1.0000	0	0	-8.0000	-7.9341
1.0000	1.0000	0	0.5000	-4.2500	-4.6432
1.0000	1.0000	0.5000	-0.5000	-21.2500	-21.4355
1.0000	1.0000	0.5000	0	-18.0000	-18.1469
1.0000	1.0000	0.5000	0.5000	-14.2500	-14.2540
1.0000	1.0000	1.0000	0	-27.0000	-26.7539
1.0000	1.0000	1.0000	0.5000	-23.2500	-22.9494
1.0000	1.5000	0	0	-9.2500	-9.8593
1.0000	1.5000	0	0.5000	-5.5000	-5.8308
1.0000	1.5000	0.5000	0	-19.2500	-19.9487
1.0000	1.5000	0.5000	0.5000	-15.5000	-15.6158

Table E.2: Training data with corresponding target and calculated outputs by ANN model for Hesse's function

Input Variables						Target	Calculated
x_1	x_2	x_3	x_4	x_5	x_6	T	Y
1	1	1	0	1	0	58	42.1801
1	1	1	0	1	2	46	39.5860
1	1	1	0	1	4	42	39.7774
1	1	1	0	1	6	46	42.4337
1	1	1	0	1	8	58	47.4817
1	1	1	0	1	10	78	55.7836
1	1	1	0	3	4	46	50.0450
1	1	1	0	3	6	50	49.0519
1	1	1	0	3	8	62	51.9238
1	1	1	0	3	10	82	60.0457
1	1	1	0	5	0	74	67.0204
1	1	1	0	5	2	62	67.5266
1	1	1	0	5	4	58	66.4954
1	1	1	0	5	6	62	64.3227
1	1	1	0	5	8	74	63.5817
1	1	1	0	5	10	94	67.9320
1	1	1	2	1	0	46	35.7059
1	1	1	2	1	2	34	34.8384
1	1	1	2	1	4	30	35.9021
1	1	1	2	1	6	34	39.1185
1	1	1	2	1	8	46	45.1760
1	1	1	2	1	10	66	54.9486
1	1	1	2	3	4	34	43.6253
1	1	1	2	3	6	38	46.4998
1	1	1	2	3	8	50	51.9209
1	1	1	2	3	10	70	61.0597
1	1	1	2	5	0	62	61.7423
1	1	1	2	5	2	50	60.6473
1	1	1	2	5	4	46	58.5919
1	1	1	2	5	6	50	57.7666
1	1	1	2	5	8	62	59.4450
1	1	1	2	5	10	82	65.3628
1	1	1	4	1	0	42	32.0018
1	1	1	4	1	2	30	31.9655
1	1	1	4	1	4	26	32.9433
1	1	1	4	1	6	30	35.6378
1	1	1	4	1	8	42	41.5081
1	1	1	4	1	10	62	52.0338
1	1	1	4	3	4	30	39.0190
1	1	1	4	3	6	34	43.9529
1	1	1	4	3	8	46	50.6096
1	1	1	4	3	10	66	59.7301
1	1	1	4	5	0	58	50.5636
1	1	1	4	5	2	46	50.5934
1	1	1	4	5	4	42	51.5861
1	1	1	4	5	6	46	53.2882
1	1	1	4	5	8	58	56.1366
1	1	1	4	5	10	78	62.1960
1	1	1	6	1	0	46	30.0359
1	1	1	6	1	2	34	30.2487
1	1	1	6	1	4	30	31.1520
1	1	1	6	1	6	34	33.5067
1	1	1	6	1	8	46	38.8779
1	1	1	6	1	10	66	49.0792
1	1	1	6	3	4	34	36.2011
1	1	1	6	3	6	38	41.4044
1	1	1	6	3	8	50	48.2970
1	1	1	6	3	10	70	57.3230
1	1	1	6	5	0	62	41.2706
1	1	1	6	5	2	50	43.9838
1	1	1	6	5	4	46	47.2981
1	1	1	6	5	6	50	50.2216
1	1	1	6	5	8	62	53.4901
1	1	1	6	5	10	82	59.4299
1	1	3	4	1	0	46	33.2738
1	1	3	4	1	2	34	34.8872
1	1	3	4	1	4	30	38.5162
1	1	3	4	1	6	34	44.7047
1	1	3	4	1	8	46	54.2804
1	1	3	4	1	10	66	66.1254
1	1	3	4	3	4	34	45.4248

Table E.2 contd.

1	1	3	4	3	6	38	50.5527
1	1	3	4	3	8	50	58.4099
1	1	3	4	3	10	70	69.2043
1	1	3	4	5	0	62	58.1506
1	1	3	4	5	2	50	55.6840
1	1	3	4	5	4	46	54.2871
1	1	3	4	5	6	50	54.9836
1	1	3	4	5	8	62	58.7786
1	1	3	4	5	10	82	67.6878
1	1	3	6	1	0	50	31.1494
1	1	3	6	1	2	38	32.5827
1	1	3	6	1	4	34	36.0316
1	1	3	6	1	6	38	42.8331
1	1	3	6	1	8	50	53.7890
1	1	3	6	1	10	70	66.6673
1	1	3	6	3	4	38	44.8458
1	1	3	6	3	6	42	50.7459
1	1	3	6	3	8	54	58.4366
1	1	3	6	3	10	74	68.7040
1	1	3	6	5	0	66	50.4024
1	1	3	6	5	2	54	50.6542
1	1	3	6	5	4	50	51.5109
1	1	3	6	5	6	54	53.3656
1	1	3	6	5	8	66	57.5645
1	1	3	6	5	10	86	66.3967
1	1	5	0	1	0	74	57.2879
1	1	5	0	1	2	62	53.7237
1	1	5	0	1	4	58	53.9855
1	1	5	0	1	6	62	59.2175
1	1	5	0	1	8	74	67.3541
1	1	5	0	1	10	94	75.7919
1	1	5	0	3	4	62	62.4433
1	1	5	0	3	6	66	65.4765
1	1	5	0	3	8	78	74.0837
1	1	5	0	3	10	98	84.6355
1	1	5	0	5	0	90	74.9365
1	1	5	0	5	2	78	77.9617
1	1	5	0	5	4	74	79.6858
1	1	5	0	5	6	78	80.1370
1	1	5	0	5	8	90	82.8038
1	1	5	0	5	10	110	90.9225
1	1	5	2	1	0	62	45.7366
1	1	5	2	1	2	50	45.5242
1	1	5	2	1	4	46	48.7562
1	1	5	2	1	6	50	55.9203
1	1	5	2	1	8	62	65.4637
1	1	5	2	1	10	82	74.7508
1	1	5	2	3	4	50	53.9309
1	1	5	2	3	6	54	59.9813
1	1	5	2	3	8	66	70.0992
1	1	5	2	3	10	86	80.6134
1	1	5	2	5	0	78	70.9611
1	1	5	2	5	2	66	70.3068
1	1	5	2	5	4	62	67.5503
1	1	5	2	5	6	66	66.5861
1	1	5	2	5	8	78	71.2577
1	1	5	2	5	10	98	81.7447
1	1	5	4	1	0	58	39.7277
1	1	5	4	1	2	46	42.3607
1	1	5	4	1	4	42	47.4465
1	1	5	4	1	6	46	56.0414
1	1	5	4	1	8	58	66.7534
1	1	5	4	1	10	78	76.2916
1	1	5	4	3	4	46	52.0796
1	1	5	4	3	6	50	59.5036
1	1	5	4	3	8	62	69.6266
1	1	5	4	3	10	82	79.5785
1	1	5	4	5	0	74	64.5197
1	1	5	4	5	2	62	60.8863
1	1	5	4	5	4	58	58.7461
1	1	5	4	5	6	62	60.5110
1	1	5	4	5	8	74	67.2938
1	1	5	4	5	10	94	78.5998
1	1	5	6	1	0	62	36.4881
1	1	5	6	1	2	50	40.5956

Table E.2 contd.

1	1	5	6	1	4	46	47.2849
1	1	5	6	1	6	50	57.4211
1	1	5	6	1	8	62	69.0035
1	1	5	6	1	10	82	78.2000
1	1	5	6	3	4	50	52.7998
1	1	5	6	3	6	54	60.2570
1	1	5	6	3	8	66	69.9500
1	1	5	6	3	10	86	79.3908
1	1	5	6	5	0	78	56.6175
1	1	5	6	5	2	66	54.8492
1	1	5	6	5	4	62	55.2855
1	1	5	6	5	6	66	58.6515
1	1	5	6	5	8	78	66.0846
1	1	5	6	5	10	98	77.3931
1	3	1	0	1	0	58	49.4770
1	3	1	0	1	2	46	48.6808
1	3	1	0	1	4	42	50.6781
1	3	1	0	1	6	46	55.9573
1	3	1	0	1	8	58	62.0017
1	3	1	0	1	10	78	66.7528
1	3	1	0	3	4	46	62.2840
1	3	1	0	3	6	50	65.5837
1	3	1	0	3	8	62	68.5418
1	3	1	0	3	10	82	71.2026
1	3	1	0	5	0	74	64.2099
1	3	1	0	5	2	62	66.9083
1	3	1	0	5	4	58	69.6947
1	3	1	0	5	6	62	73.2610
1	3	1	0	5	8	74	76.7789
1	3	1	0	5	10	94	80.0401
1	3	1	2	1	0	46	40.1450
1	3	1	2	1	2	34	39.3492
1	3	1	2	1	4	30	42.5037
1	3	1	2	1	6	34	48.4580
1	3	1	2	1	8	46	54.8694
1	3	1	2	1	10	66	60.5582
1	3	1	2	3	4	34	48.1189
1	3	1	2	3	6	38	53.3796
1	3	1	2	3	8	50	60.0528
1	3	1	2	3	10	70	65.5602
1	3	1	2	5	0	62	57.1681
1	3	1	2	5	2	50	58.9657
1	3	1	2	5	4	46	59.4527
1	3	1	2	5	6	50	60.4149
1	3	1	2	5	8	62	64.1115
1	3	1	2	5	10	82	70.7356
1	3	1	4	1	0	42	33.6892
1	3	1	4	1	2	30	34.8510
1	3	1	4	1	4	26	38.9322
1	3	1	4	1	6	30	44.6335
1	3	1	4	1	8	42	50.0564
1	3	1	4	1	10	62	55.0379
1	3	1	4	3	4	30	39.4226
1	3	1	4	3	6	34	46.1524
1	3	1	4	3	8	46	54.5306
1	3	1	4	3	10	66	61.6103
1	3	1	4	5	0	58	48.1972
1	3	1	4	5	2	46	47.4247
1	3	1	4	5	4	42	47.4125
1	3	1	4	5	6	46	50.6636
1	3	1	4	5	8	58	57.8254
1	3	1	4	5	10	78	67.4342
1	3	1	6	1	0	46	31.6369
1	3	1	6	1	2	34	33.7536
1	3	1	6	1	4	30	38.1225
1	3	1	6	1	6	34	43.5864
1	3	1	6	1	8	46	48.2374
1	3	1	6	1	10	66	52.0902
1	3	1	6	3	4	34	36.2913
1	3	1	6	3	6	38	42.7472
1	3	1	6	3	8	50	51.0222
1	3	1	6	3	10	70	59.0208
1	3	1	6	5	0	62	39.2580
1	3	1	6	5	2	50	37.9358
1	3	1	6	5	4	46	40.1754

Table E.2 contd.

1	3	1	6	5	6	50	46.5182
1	3	1	6	5	8	62	56.3909
1	3	1	6	5	10	82	68.1171
1	3	3	4	1	0	46	38.0004
1	3	3	4	1	2	34	41.8356
1	3	3	4	1	4	30	48.2666
1	3	3	4	1	6	34	54.7310
1	3	3	4	1	8	46	59.7107
1	3	3	4	1	10	66	63.0488
1	3	3	4	3	4	34	52.8521
1	3	3	4	3	6	38	58.6670
1	3	3	4	3	8	50	62.9045
1	3	3	4	3	10	70	66.1921
1	3	3	4	5	0	62	57.5178
1	3	3	4	5	2	50	57.3328
1	3	3	4	5	4	46	58.0902
1	3	3	4	5	6	50	61.5498
1	3	3	4	5	8	62	66.9716
1	3	3	4	5	10	82	74.1808
1	3	3	6	1	0	50	34.9606
1	3	3	6	1	2	38	39.4656
1	3	3	6	1	4	34	45.4911
1	3	3	6	1	6	38	51.2338
1	3	3	6	1	8	50	56.3107
1	3	3	6	1	10	70	60.6702
1	3	3	6	3	4	38	48.0273
1	3	3	6	3	6	42	55.9567
1	3	3	6	3	8	54	61.8262
1	3	3	6	3	10	74	66.3300
1	3	3	6	5	0	66	48.0753
1	3	3	6	5	2	54	47.9530
1	3	3	6	5	4	50	51.3646
1	3	3	6	5	6	54	58.4670
1	3	3	6	5	8	66	66.9982
1	3	3	6	5	10	86	75.9967
1	3	5	0	1	0	74	75.0474
1	3	5	0	1	2	62	78.9228
1	3	5	0	1	4	58	76.7907
1	3	5	0	1	6	62	72.6990
1	3	5	0	1	8	74	70.0120
1	3	5	0	1	10	94	69.0813
1	3	5	0	3	4	62	85.8712
1	3	5	0	3	6	66	82.8113
1	3	5	0	3	8	78	77.1737
1	3	5	0	3	10	98	73.8220
1	3	5	0	5	0	90	71.2511
1	3	5	0	5	2	78	78.6827
1	3	5	0	5	4	74	87.2814
1	3	5	0	5	6	78	92.2277
1	3	5	0	5	8	90	91.9628
1	3	5	0	5	10	110	90.2639
1	3	5	2	1	0	62	62.5130
1	3	5	2	1	2	50	65.3993
1	3	5	2	1	4	46	66.3298
1	3	5	2	1	6	50	66.5201
1	3	5	2	1	8	62	66.6862
1	3	5	2	1	10	82	67.1441
1	3	5	2	3	4	50	73.3427
1	3	5	2	3	6	54	70.8789
1	3	5	2	3	8	66	69.2527
1	3	5	2	3	10	86	69.7062
1	3	5	2	5	0	78	69.0576
1	3	5	2	5	2	66	75.0469
1	3	5	2	5	4	62	80.5757
1	3	5	2	5	6	66	81.5694
1	3	5	2	5	8	78	80.0309
1	3	5	2	5	10	98	81.5248
1	3	5	4	1	0	58	50.4790
1	3	5	4	1	2	46	56.3751
1	3	5	4	1	4	42	61.2409
1	3	5	4	1	6	46	64.0173
1	3	5	4	1	8	58	65.4640
1	3	5	4	1	10	78	66.5179
1	3	5	4	3	4	46	64.1207
1	3	5	4	3	6	50	65.5366

Table E.2 contd.

1	3	5	4	3	8	62	66.9054
1	3	5	4	3	10	82	69.2229
1	3	5	4	5	0	74	65.4422
1	3	5	4	5	2	62	68.2121
1	3	5	4	5	4	58	70.3862
1	3	5	4	5	6	62	71.7625
1	3	5	4	5	8	74	74.4846
1	3	5	4	5	10	94	80.0001
1	3	5	6	1	0	62	43.7907
1	3	5	6	1	2	50	50.9536
1	3	5	6	1	4	46	57.4626
1	3	5	6	1	6	50	61.8089
1	3	5	6	1	8	62	64.3094
1	3	5	6	1	10	82	65.9983
1	3	5	6	3	4	50	60.0402
1	3	5	6	3	6	54	63.8129
1	3	5	6	3	8	66	66.6553
1	3	5	6	3	10	86	70.1752
1	3	5	6	5	0	78	57.6813
1	3	5	6	5	2	66	58.6677
1	3	5	6	5	4	62	62.6206
1	3	5	6	5	6	66	68.2336
1	3	5	6	5	8	78	74.4460
1	3	5	6	5	10	98	81.2211
3	1	1	0	1	0	58	53.1316
3	1	1	0	1	2	46	47.3262
3	1	1	0	1	4	42	45.6957
3	1	1	0	1	6	46	46.9252
3	1	1	0	1	8	58	50.5507
3	1	1	0	1	10	78	58.3138
3	1	1	0	3	4	46	53.3535
3	1	1	0	3	6	50	54.7422
3	1	1	0	3	8	62	60.8780
3	1	1	0	3	10	82	74.5165
3	1	1	0	5	0	74	76.3130
3	1	1	0	5	2	62	73.5817
3	1	1	0	5	4	58	70.6859
3	1	1	0	5	6	62	70.8340
3	1	1	0	5	8	74	78.6123
3	1	1	0	5	10	94	99.1465
3	1	1	2	1	0	46	49.2624
3	1	1	2	1	2	34	44.8051
3	1	1	2	1	4	30	43.2115
3	1	1	2	1	6	34	44.3262
3	1	1	2	1	8	46	48.6347
3	1	1	2	1	10	66	57.7189
3	1	1	2	3	4	34	49.4120
3	1	1	2	3	6	38	53.7122
3	1	1	2	3	8	50	61.0066
3	1	1	2	3	10	70	74.0267
3	1	1	2	5	0	62	67.4267
3	1	1	2	5	2	50	64.1868
3	1	1	2	5	4	46	63.0705
3	1	1	2	5	6	50	65.6692
3	1	1	2	5	8	62	74.5982
3	1	1	2	5	10	82	94.7638
3	1	1	4	1	0	42	46.9692
3	1	1	4	1	2	30	42.8201
3	1	1	4	1	4	26	40.9248
3	1	1	4	1	6	30	41.4630
3	1	1	4	1	8	42	45.3961
3	1	1	4	1	10	62	54.5501
3	1	1	4	3	4	30	45.7371
3	1	1	4	3	6	34	50.8705
3	1	1	4	3	8	46	58.2337
3	1	1	4	3	10	66	70.2270
3	1	1	4	5	0	58	55.2527
3	1	1	4	5	2	46	55.4011
3	1	1	4	5	4	42	57.3111
3	1	1	4	5	6	46	61.1815
3	1	1	4	5	8	58	69.8576
3	1	1	4	5	10	78	88.5773
3	1	1	6	1	0	46	46.1665
3	1	1	6	1	2	34	42.0279
3	1	1	6	1	4	30	40.0661

Table E.2 contd.

3	1	1	6	1	6	34	40.4101
3	1	1	6	1	8	46	43.7398
3	1	1	6	1	10	66	51.6168
3	1	1	6	3	4	34	43.6924
3	1	1	6	3	6	38	47.8446
3	1	1	6	3	8	50	54.4835
3	1	1	6	3	10	70	65.8370
3	1	1	6	5	0	62	48.5114
3	1	1	6	5	2	50	50.0972
3	1	1	6	5	4	46	53.0793
3	1	1	6	5	6	50	57.5247
3	1	1	6	5	8	62	65.9505
3	1	1	6	5	10	82	83.4879
3	1	3	4	1	0	46	45.7935
3	1	3	4	1	2	34	43.6341
3	1	3	4	1	4	30	44.2999
3	1	3	4	1	6	34	47.4667
3	1	3	4	1	8	46	53.4283
3	1	3	4	1	10	66	64.1442
3	1	3	4	3	4	34	53.0162
3	1	3	4	3	6	38	56.4170
3	1	3	4	3	8	50	61.9041
3	1	3	4	3	10	70	73.6709
3	1	3	4	5	0	62	64.2756
3	1	3	4	5	2	50	60.7055
3	1	3	4	5	4	46	59.7595
3	1	3	4	5	6	50	62.1012
3	1	3	4	5	8	62	70.2415
3	1	3	4	5	10	82	89.6587
3	1	3	6	1	0	50	45.4137
3	1	3	6	1	2	38	42.7600
3	1	3	6	1	4	34	43.0577
3	1	3	6	1	6	38	46.5147
3	1	3	6	1	8	50	53.1228
3	1	3	6	1	10	70	63.8944
3	1	3	6	3	4	38	52.6821
3	1	3	6	3	6	42	56.1488
3	1	3	6	3	8	54	61.1606
3	1	3	6	3	10	74	71.9992
3	1	3	6	5	0	66	58.2877
3	1	3	6	5	2	54	57.5674
3	1	3	6	5	4	50	57.9369
3	1	3	6	5	6	54	60.5276
3	1	3	6	5	8	66	68.1423
3	1	3	6	5	10	86	86.2406
3	1	5	0	1	0	74	64.6142
3	1	5	0	1	2	62	56.9786
3	1	5	0	1	4	58	53.9397
3	1	5	0	1	6	62	55.3415
3	1	5	0	1	8	74	62.7034
3	1	5	0	1	10	94	79.2094
3	1	5	0	3	4	62	62.8385
3	1	5	0	3	6	66	64.7766
3	1	5	0	3	8	78	73.3654
3	1	5	0	3	10	98	93.6925
3	1	5	0	5	0	90	86.8393
3	1	5	0	5	2	78	85.6176
3	1	5	0	5	4	74	82.7071
3	1	5	0	5	6	78	81.9252
3	1	5	0	5	8	90	90.7864
3	1	5	0	5	10	110	118.4490
3	1	5	2	1	0	62	55.0849
3	1	5	2	1	2	50	51.6256
3	1	5	2	1	4	46	50.9112
3	1	5	2	1	6	50	53.5366
3	1	5	2	1	8	62	61.2660
3	1	5	2	1	10	82	76.6380
3	1	5	2	3	4	50	57.4487
3	1	5	2	3	6	54	60.7188
3	1	5	2	3	8	66	68.5868
3	1	5	2	3	10	86	86.4948
3	1	5	2	5	0	78	78.9541
3	1	5	2	5	2	66	72.7836
3	1	5	2	5	4	62	68.2979
3	1	5	2	5	6	66	69.2492

Table E.2 contd.

3	1	5	2	5	8	78	79.2977
3	1	5	2	5	10	98	105.5336
3	1	5	4	1	0	58	51.1586
3	1	5	4	1	2	46	50.0822
3	1	5	4	1	4	42	50.9634
3	1	5	4	1	6	46	54.5691
3	1	5	4	1	8	58	62.3622
3	1	5	4	1	10	78	76.5066
3	1	5	4	3	4	46	57.1813
3	1	5	4	3	6	50	60.1427
3	1	5	4	3	8	62	66.9435
3	1	5	4	3	10	82	83.0244
3	1	5	4	5	0	74	69.3352
3	1	5	4	5	2	62	63.7740
3	1	5	4	5	4	58	61.9938
3	1	5	4	5	6	62	64.6639
3	1	5	4	5	8	74	74.5798
3	1	5	4	5	10	94	98.8483
3	1	5	6	1	0	62	49.2346
3	1	5	6	1	2	50	49.3439
3	1	5	6	1	4	46	51.7429
3	1	5	6	1	6	50	56.2204
3	1	5	6	1	8	62	63.7521
3	1	5	6	1	10	82	76.7234
3	1	5	6	3	4	50	58.0757
3	1	5	6	3	6	54	60.3323
3	1	5	6	3	8	66	66.2261
3	1	5	6	3	10	86	80.9150
3	1	5	6	5	0	78	63.2728
3	1	5	6	5	2	66	60.4296
3	1	5	6	5	4	62	60.0901
3	1	5	6	5	6	66	62.9861
3	1	5	6	5	8	78	72.1309
3	1	5	6	5	10	98	94.4826
3	3	1	0	1	0	58	44.5722
3	3	1	0	1	2	46	40.8862
3	3	1	0	1	4	42	40.1606
3	3	1	0	1	6	46	43.6067
3	3	1	0	1	8	58	52.1903
3	3	1	0	1	10	78	65.7163
3	3	1	0	3	4	46	49.3179
3	3	1	0	3	6	50	49.8575
3	3	1	0	3	8	62	55.5250
3	3	1	0	3	10	82	70.1075
3	3	1	0	5	0	74	62.2080
3	3	1	0	5	2	62	63.4175
3	3	1	0	5	4	58	62.7146
3	3	1	0	5	6	62	61.9002
3	3	1	0	5	8	74	66.6086
3	3	1	0	5	10	94	86.0998
3	3	1	2	1	0	46	36.3670
3	3	1	2	1	2	34	34.8755
3	3	1	2	1	4	30	35.5258
3	3	1	2	1	6	34	38.9275
3	3	1	2	1	8	46	46.5608
3	3	1	2	1	10	66	59.2365
3	3	1	2	3	4	34	38.8213
3	3	1	2	3	6	38	42.0224
3	3	1	2	3	8	50	49.6626
3	3	1	2	3	10	70	64.8354
3	3	1	2	5	0	62	53.1646
3	3	1	2	5	2	50	52.3707
3	3	1	2	5	4	46	50.3208
3	3	1	2	5	6	50	50.4794
3	3	1	2	5	8	62	57.6856
3	3	1	2	5	10	82	79.1926
3	3	1	4	1	0	42	32.6191
3	3	1	4	1	2	30	32.5128
3	3	1	4	1	4	26	33.5903
3	3	1	4	1	6	30	36.6492
3	3	1	4	1	8	42	43.1555
3	3	1	4	1	10	62	54.1715
3	3	1	4	3	4	30	33.7571
3	3	1	4	3	6	34	37.3726
3	3	1	4	3	8	46	45.3630

Table E.2 contd.

3	3	1	4	3	10	66	60.8627
3	3	1	4	5	0	58	42.5822
3	3	1	4	5	2	46	40.6604
3	3	1	4	5	4	42	40.9855
3	3	1	4	5	6	46	44.5722
3	3	1	4	5	8	58	54.1839
3	3	1	4	5	10	78	76.9353
3	3	1	6	1	0	46	31.8660
3	3	1	6	1	2	34	32.1888
3	3	1	6	1	4	30	33.3547
3	3	1	6	1	6	34	36.2316
3	3	1	6	1	8	46	42.1317
3	3	1	6	1	10	66	51.8405
3	3	1	6	3	4	34	32.1242
3	3	1	6	3	6	38	35.2309
3	3	1	6	3	8	50	42.6056
3	3	1	6	3	10	70	58.0885
3	3	1	6	5	0	62	34.7913
3	3	1	6	5	2	50	34.3039
3	3	1	6	5	4	46	36.5925
3	3	1	6	5	6	50	42.1549
3	3	1	6	5	8	62	53.5084
3	3	1	6	5	10	82	77.3130
3	3	3	4	1	0	46	32.9658
3	3	3	4	1	2	34	33.9246
3	3	3	4	1	4	30	37.5907
3	3	3	4	1	6	34	44.9458
3	3	3	4	1	8	46	55.1431
3	3	3	4	1	10	66	65.8875
3	3	3	4	3	4	34	40.9802
3	3	3	4	3	6	38	46.7122
3	3	3	4	3	8	50	56.7691
3	3	3	4	3	10	70	72.4485
3	3	3	4	5	0	62	50.9566
3	3	3	4	5	2	50	47.8519
3	3	3	4	5	4	46	46.6895
3	3	3	4	5	6	50	50.5734
3	3	3	4	5	8	62	63.3632
3	3	3	4	5	10	82	90.6890
3	3	3	6	1	0	50	31.5960
3	3	3	6	1	2	38	32.7713
3	3	3	6	1	4	34	35.9621
3	3	3	6	1	6	38	42.3614
3	3	3	6	1	8	50	51.8601
3	3	3	6	1	10	70	62.9239
3	3	3	6	3	4	38	37.7954
3	3	3	6	3	6	42	44.5958
3	3	3	6	3	8	54	55.7576
3	3	3	6	3	10	74	72.5059
3	3	3	6	5	0	66	41.1652
3	3	3	6	5	2	54	41.1196
3	3	3	6	5	4	50	43.5078
3	3	3	6	5	6	54	49.9568
3	3	3	6	5	8	66	64.2073
3	3	3	6	5	10	86	91.4674
3	3	5	0	1	0	74	55.8651
3	3	5	0	1	2	62	55.2585
3	3	5	0	1	4	58	57.3679
3	3	5	0	1	6	62	62.0818
3	3	5	0	1	8	74	67.0732
3	3	5	0	1	10	94	72.6550
3	3	5	0	3	4	62	62.5522
3	3	5	0	3	6	66	64.9640
3	3	5	0	3	8	78	70.8954
3	3	5	0	3	10	98	81.8189
3	3	5	0	5	0	90	65.7495
3	3	5	0	5	2	78	66.7992
3	3	5	0	5	4	74	69.4031
3	3	5	0	5	6	78	75.3019
3	3	5	0	5	8	90	87.3040
3	3	5	0	5	10	110	110.4691
3	3	5	2	1	0	62	44.7618
3	3	5	2	1	2	50	46.6861
3	3	5	2	1	4	46	52.0836
3	3	5	2	1	6	50	59.1168

Table E.2 contd.

3	3	5	2	1	8	62	65.3061
3	3	5	2	1	10	82	71.3067
3	3	5	2	3	4	50	53.3031
3	3	5	2	3	6	54	58.3460
3	3	5	2	3	8	66	67.0498
3	3	5	2	3	10	86	79.8118
3	3	5	2	5	0	78	62.8259
3	3	5	2	5	2	66	61.5869
3	3	5	2	5	4	62	61.0567
3	3	5	2	5	6	66	65.1842
3	3	5	2	5	8	78	78.6638
3	3	5	2	5	10	98	104.7618
3	3	5	4	1	0	58	37.7882
3	3	5	4	1	2	46	41.9969
3	3	5	4	1	4	42	49.3343
3	3	5	4	1	6	46	57.6833
3	3	5	4	1	8	58	64.6046
3	3	5	4	1	10	78	70.9319
3	3	5	4	3	4	46	49.1474
3	3	5	4	3	6	50	56.4264
3	3	5	4	3	8	62	66.5949
3	3	5	4	3	10	82	80.4560
3	3	5	4	5	0	74	56.8265
3	3	5	4	5	2	62	53.3868
3	3	5	4	5	4	58	53.3175
3	3	5	4	5	6	62	60.4348
3	3	5	4	5	8	74	76.9433
3	3	5	4	5	10	94	104.0400
3	3	5	6	1	0	62	34.5410
3	3	5	6	1	2	50	38.8612
3	3	5	6	1	4	46	46.5804
3	3	5	6	1	6	50	55.8172
3	3	5	6	1	8	62	63.6646
3	3	5	6	1	10	82	70.6731
3	3	5	6	3	4	50	47.5430
3	3	5	6	3	6	54	56.1220
3	3	5	6	3	8	66	67.3948
3	3	5	6	3	10	86	82.2352
3	3	5	6	5	0	78	48.2641
3	3	5	6	5	2	66	47.0277
3	3	5	6	5	4	62	50.4132
3	3	5	6	5	6	66	60.2422
3	3	5	6	5	8	78	77.7859
3	3	5	6	5	10	98	103.9789
5	1	1	0	1	0	258	256.8155
5	1	1	0	1	2	246	244.9671
5	1	1	0	1	4	242	236.2651
5	1	1	0	1	6	246	233.7243
5	1	1	0	1	8	258	241.7013
5	1	1	0	1	10	278	261.2602
5	1	1	0	3	4	246	239.0324
5	1	1	0	3	6	250	244.1660
5	1	1	0	3	8	262	260.4919
5	1	1	0	3	10	282	278.8071
5	1	1	0	5	0	274	276.3268
5	1	1	0	5	2	262	262.5026
5	1	1	0	5	4	258	254.8820
5	1	1	0	5	6	262	260.6749
5	1	1	0	5	8	274	273.6340
5	1	1	0	5	10	294	282.0692
5	1	1	2	1	0	246	254.1313
5	1	1	2	1	2	234	243.9737
5	1	1	2	1	4	230	235.1443
5	1	1	2	1	6	234	232.1126
5	1	1	2	1	8	246	240.9159
5	1	1	2	1	10	266	262.0657
5	1	1	2	3	4	234	238.1681
5	1	1	2	3	6	238	246.2460
5	1	1	2	3	8	250	262.3797
5	1	1	2	3	10	270	279.5304
5	1	1	2	5	0	262	262.4191
5	1	1	2	5	2	250	251.9964
5	1	1	2	5	4	246	249.1890
5	1	1	2	5	6	250	257.6856
5	1	1	2	5	8	262	271.6124

Table E.2 contd.

5	1	1	2	5	10	282	280.1120
5	1	1	4	1	0	242	251.0009
5	1	1	4	1	2	230	241.8478
5	1	1	4	1	4	226	233.3966
5	1	1	4	1	6	230	229.9207
5	1	1	4	1	8	242	237.4015
5	1	1	4	1	10	262	257.4215
5	1	1	4	3	4	230	234.8381
5	1	1	4	3	6	234	242.4275
5	1	1	4	3	8	246	258.3457
5	1	1	4	3	10	266	275.7456
5	1	1	4	5	0	258	247.3576
5	1	1	4	5	2	246	242.3123
5	1	1	4	5	4	242	243.0312
5	1	1	4	5	6	246	252.7918
5	1	1	4	5	8	258	267.1220
5	1	1	4	5	10	278	275.5971
5	1	1	6	1	0	246	250.3558
5	1	1	6	1	2	234	242.5779
5	1	1	6	1	4	230	235.5504
5	1	1	6	1	6	234	232.4545
5	1	1	6	1	8	246	237.6524
5	1	1	6	1	10	266	253.8035
5	1	1	6	3	4	234	234.4460
5	1	1	6	3	6	238	238.6267
5	1	1	6	3	8	250	253.2023
5	1	1	6	3	10	270	271.6487
5	1	1	6	5	0	262	240.4772
5	1	1	6	5	2	250	235.8493
5	1	1	6	5	4	246	237.5915
5	1	1	6	5	6	250	248.6143
5	1	1	6	5	8	262	263.9720
5	1	1	6	5	10	282	272.9563
5	1	3	4	1	0	246	243.8967
5	1	3	4	1	2	234	238.3291
5	1	3	4	1	4	230	235.8243
5	1	3	4	1	6	234	238.3147
5	1	3	4	1	8	246	247.7528
5	1	3	4	1	10	266	265.6936
5	1	3	4	3	4	234	243.4574
5	1	3	4	3	6	238	245.9789
5	1	3	4	3	8	250	257.8869
5	1	3	4	3	10	270	275.5836
5	1	3	4	5	0	262	258.8729
5	1	3	4	5	2	250	247.3023
5	1	3	4	5	4	246	243.0731
5	1	3	4	5	6	250	251.4281
5	1	3	4	5	8	262	268.0734
5	1	3	4	5	10	282	281.0395
5	1	3	6	1	0	250	245.4319
5	1	3	6	1	2	238	240.1730
5	1	3	6	1	4	234	237.3482
5	1	3	6	1	6	238	239.0238
5	1	3	6	1	8	250	247.5382
5	1	3	6	1	10	270	264.6335
5	1	3	6	3	4	238	243.7006
5	1	3	6	3	6	242	246.0529
5	1	3	6	3	8	254	257.0010
5	1	3	6	3	10	274	273.8844
5	1	3	6	5	0	266	253.8869
5	1	3	6	5	2	254	245.8540
5	1	3	6	5	4	250	242.4843
5	1	3	6	5	6	254	249.9876
5	1	3	6	5	8	266	265.6957
5	1	3	6	5	10	286	278.2469
5	1	5	0	1	0	274	272.2026
5	1	5	0	1	2	262	261.9296
5	1	5	0	1	4	258	254.5034
5	1	5	0	1	6	262	253.1207
5	1	5	0	1	8	274	262.3667
5	1	5	0	1	10	294	283.2439
5	1	5	0	3	4	262	260.9696
5	1	5	0	3	6	266	264.1317
5	1	5	0	3	8	278	278.4962
5	1	5	0	3	10	298	297.0518

Table E.2 contd.

5	1	5	0	5	0	290	297.3549
5	1	5	0	5	2	278	285.3027
5	1	5	0	5	4	274	276.8576
5	1	5	0	5	6	278	281.0171
5	1	5	0	5	8	290	294.0625
5	1	5	0	5	10	310	303.8376
5	1	5	2	1	0	262	259.4620
5	1	5	2	1	2	250	254.5259
5	1	5	2	1	4	246	250.4660
5	1	5	2	1	6	250	250.6769
5	1	5	2	1	8	262	259.4340
5	1	5	2	1	10	282	279.0010
5	1	5	2	3	4	250	254.3203
5	1	5	2	3	6	254	257.4896
5	1	5	2	3	8	266	271.4974
5	1	5	2	3	10	286	290.9329
5	1	5	2	5	0	278	281.3958
5	1	5	2	5	2	266	265.5590
5	1	5	2	5	4	262	259.4098
5	1	5	2	5	6	266	268.1578
5	1	5	2	5	8	278	285.2465
5	1	5	2	5	10	298	297.8822
5	1	5	4	1	0	258	253.3949
5	1	5	4	1	2	246	252.6481
5	1	5	4	1	4	242	251.8079
5	1	5	4	1	6	246	252.7094
5	1	5	4	1	8	258	259.6795
5	1	5	4	1	10	278	277.1661
5	1	5	4	3	4	246	253.7020
5	1	5	4	3	6	250	255.1960
5	1	5	4	3	8	262	267.8732
5	1	5	4	3	10	282	287.4665
5	1	5	4	5	0	274	268.2526
5	1	5	4	5	2	262	255.6224
5	1	5	4	5	4	258	252.2932
5	1	5	4	5	6	262	262.3561
5	1	5	4	5	8	274	280.7286
5	1	5	4	5	10	294	294.9562
5	1	5	6	1	0	262	250.5934
5	1	5	6	1	2	250	252.0291
5	1	5	6	1	4	246	253.6057
5	1	5	6	1	6	250	254.7610
5	1	5	6	1	8	262	260.0176
5	1	5	6	1	10	282	275.6274
5	1	5	6	3	4	250	254.3294
5	1	5	6	3	6	254	254.1474
5	1	5	6	3	8	266	265.2538
5	1	5	6	3	10	286	284.5544
5	1	5	6	5	0	278	263.1064
5	1	5	6	5	2	266	252.8183
5	1	5	6	5	4	262	249.6480
5	1	5	6	5	6	266	258.9282
5	1	5	6	5	8	278	277.2672
5	1	5	6	5	10	298	292.4671

Table E.3: Training data with corresponding target and calculated outputs by ANN model for Optimum fuel allocation problem

Input variables			Target	Calculated
X_1	X_3	X_4	T	Y
25	0	0.6000	3.8705	3.7785
25	0	0.8000	5.1606	4.6614
25	0	1.0000	6.4508	5.8049
25	0.2000	0.4000	3.8130	3.9874
25	0.2000	0.6000	5.1032	4.9290
25	0.2000	0.8000	6.3934	6.0642
25	0.2000	1.0000	7.6835	7.2262
25	0.4000	0.2000	3.7556	4.1927
25	0.4000	0.4000	5.0458	5.2772
25	0.4000	0.6000	6.3359	6.5192
25	0.4000	0.8000	7.6261	7.6070
25	0.4000	1.0000	8.9163	8.6168
25	0.6000	0	3.6982	4.4826
25	0.6000	0.2000	4.9883	5.7595
25	0.6000	0.4000	6.2785	7.0853
25	0.6000	0.6000	7.5687	8.0502
25	0.6000	0.8000	8.8588	8.7210
25	0.6000	1.0000	10.1490	9.3333
25	0.8000	0	4.9309	6.3445
25	0.8000	0.2000	6.2211	7.6352
25	0.8000	0.4000	7.5112	8.4035
25	0.8000	0.6000	8.8014	8.8333
25	0.8000	0.8000	10.0916	9.1523
25	0.8000	1.0000	11.3817	9.5750
25	1.0000	0	6.1636	8.0688
25	1.0000	0.2000	7.4538	8.6372
25	1.0000	0.4000	8.7440	8.9066
25	1.0000	0.6000	10.0341	9.0806
25	1.0000	0.8000	11.3243	9.2855
25	1.0000	1.0000	12.6144	9.6660
26	0	0.6000	3.7217	3.6195
26	0	0.8000	4.9623	4.5414
26	0	1.0000	6.2028	6.0611
26	0.2000	0.4000	3.7590	3.7721
26	0.2000	0.6000	4.9996	4.6115
26	0.2000	0.8000	6.2401	5.9058
26	0.2000	1.0000	7.4807	7.4166
26	0.4000	0.2000	3.7963	3.8544
26	0.4000	0.4000	5.0369	4.7072
26	0.4000	0.6000	6.2775	5.9567
26	0.4000	0.8000	7.5180	7.3235
26	0.4000	1.0000	8.7586	8.9733
26	0.6000	0	3.8337	3.9701
26	0.6000	0.2000	5.0742	4.9187
26	0.6000	0.4000	6.3148	6.2333
26	0.6000	0.6000	7.5554	7.5537
26	0.6000	0.8000	8.7959	8.8922
26	0.6000	1.0000	10.0365	10.3424
26	0.8000	0	5.1116	5.2835
26	0.8000	0.2000	6.3521	6.6993
26	0.8000	0.4000	7.5927	7.9192
26	0.8000	0.6000	8.8333	8.9239
26	0.8000	0.8000	10.0738	9.8753
26	0.8000	1.0000	11.3144	10.9616
26	1.0000	0	6.3895	7.2448
26	1.0000	0.2000	7.6300	8.2695
26	1.0000	0.4000	8.8706	8.9529
26	1.0000	0.6000	10.1112	9.5511
26	1.0000	0.8000	11.3517	10.2536
26	1.0000	1.0000	12.5923	11.1878
27	0	0.6000	3.5740	3.8598
27	0	0.8000	4.7653	4.8529
27	0	1.0000	5.9567	6.4909
27	0.2000	0.4000	3.7063	3.9896
27	0.2000	0.6000	4.8976	4.9216
27	0.2000	0.8000	6.0890	6.4075
27	0.2000	1.0000	7.2803	8.0321
27	0.4000	0.2000	3.8386	3.9217
27	0.4000	0.4000	5.0299	4.7977
27	0.4000	0.6000	6.2213	6.2428
27	0.4000	0.8000	7.4126	7.7592

Table E.3 contd.

27	0.4000	1.0000	8.6039	9.3189
27	0.6000	0	3.9709	3.8561
27	0.6000	0.2000	5.1622	4.6971
27	0.6000	0.4000	6.3536	6.1325
27	0.6000	0.6000	7.5449	7.7152
27	0.6000	0.8000	8.7362	9.2806
27	0.6000	1.0000	9.9276	10.9799
27	0.8000	0	5.2945	4.7274
27	0.8000	0.2000	6.4859	6.1804
27	0.8000	0.4000	7.6772	7.8213
27	0.8000	0.6000	8.8685	9.4234
27	0.8000	0.8000	10.0599	10.8496
27	0.8000	1.0000	11.2512	12.1600
27	1.0000	0	6.6182	6.4474
27	1.0000	0.2000	7.8095	8.0297
27	1.0000	0.4000	9.0008	9.4704
27	1.0000	0.6000	10.1922	10.6877
27	1.0000	0.8000	11.3835	11.6521
27	1.0000	1.0000	12.5748	12.5890
28	0	0.6000	3.4274	3.9651
28	0	0.8000	4.5699	4.6923
28	0	1.0000	5.7123	6.0179
28	0.2000	0.4000	3.6549	4.4529
28	0.2000	0.6000	4.7974	5.1580
28	0.2000	0.8000	5.9398	6.4172
28	0.2000	1.0000	7.0823	7.9746
28	0.4000	0.2000	3.8824	4.4967
28	0.4000	0.4000	5.0248	5.2558
28	0.4000	0.6000	6.1673	6.4676
28	0.4000	0.8000	7.3098	7.8597
28	0.4000	1.0000	8.4523	9.1309
28	0.6000	0	4.1099	4.3055
28	0.6000	0.2000	5.2523	5.1424
28	0.6000	0.4000	6.3948	6.4654
28	0.6000	0.6000	7.5373	7.8220
28	0.6000	0.8000	8.6797	9.0299
28	0.6000	1.0000	9.8222	10.4609
28	0.8000	0	5.4798	4.9579
28	0.8000	0.2000	6.6223	6.4233
28	0.8000	0.4000	7.7648	7.9632
28	0.8000	0.6000	8.9072	9.2598
28	0.8000	0.8000	10.0497	10.6050
28	0.8000	1.0000	11.1922	12.1035
28	1.0000	0	6.8498	6.3941
28	1.0000	0.2000	7.9922	8.1750
28	1.0000	0.4000	9.1347	9.7014
28	1.0000	0.6000	10.2772	10.9302
28	1.0000	0.8000	11.4197	11.9797
28	1.0000	1.0000	12.5621	13.0295
29	0	0.6000	3.2819	3.5797
29	0	0.8000	4.3759	4.1274
29	0	1.0000	5.4699	5.0650
29	0.2000	0.4000	3.6048	4.3371
29	0.2000	0.6000	4.6988	4.8126
29	0.2000	0.8000	5.7927	5.8688
29	0.2000	1.0000	6.8867	7.2776
29	0.4000	0.2000	3.9277	4.8055
29	0.4000	0.4000	5.0217	5.2339
29	0.4000	0.6000	6.1156	6.1463
29	0.4000	0.8000	7.2096	7.5889
29	0.4000	1.0000	8.3036	8.7340
29	0.6000	0	4.2506	4.8741
29	0.6000	0.2000	5.3445	5.3598
29	0.6000	0.4000	6.4385	6.2795
29	0.6000	0.6000	7.5325	7.5203
29	0.6000	0.8000	8.6265	8.7131
29	0.6000	1.0000	9.7204	9.7371
29	0.8000	0	5.6674	5.3477
29	0.8000	0.2000	6.7614	6.3822
29	0.8000	0.4000	7.8554	7.6224
29	0.8000	0.6000	8.9493	8.6560
29	0.8000	0.8000	10.0433	9.8561
29	0.8000	1.0000	11.1373	11.2302
29	1.0000	0	7.0843	6.4693
29	1.0000	0.2000	8.1783	7.8588

Table E.3 contd.

29	1.0000	0.4000	9.2722	8.9927
29	1.0000	0.6000	10.3662	10.1038
29	1.0000	0.8000	11.4602	11.4846
29	1.0000	1.0000	12.5542	12.7698
30	0	0.6000	3.1375	3.2551
30	0	0.8000	4.1834	3.6869
30	0	1.0000	5.2292	4.2914
30	0.2000	0.4000	3.5560	3.8115
30	0.2000	0.6000	4.6019	4.2381
30	0.2000	0.8000	5.6477	5.0929
30	0.2000	1.0000	6.6935	6.2688
30	0.4000	0.2000	3.9745	4.5801
30	0.4000	0.4000	5.0204	4.8622
30	0.4000	0.6000	6.0662	5.6037
30	0.4000	0.8000	7.1120	6.9619
30	0.4000	1.0000	8.1579	8.0255
30	0.6000	0	4.3930	4.9528
30	0.6000	0.2000	5.4389	5.2064
30	0.6000	0.4000	6.4847	5.8548
30	0.6000	0.6000	7.5305	7.0991
30	0.6000	0.8000	8.5764	8.3971
30	0.6000	1.0000	9.6222	9.0913
30	0.8000	0	5.8574	5.3248
30	0.8000	0.2000	6.9032	6.0122
30	0.8000	0.4000	7.9490	7.1422
30	0.8000	0.6000	8.9949	8.2676
30	0.8000	0.8000	10.0407	9.3216
30	0.8000	1.0000	11.0866	10.1773
30	1.0000	0	7.3217	6.1375
30	1.0000	0.2000	8.3675	7.3006
30	1.0000	0.4000	9.4134	8.3055
30	1.0000	0.6000	10.4592	9.2840
30	1.0000	0.8000	11.5051	10.6097
30	1.0000	1.0000	12.5509	11.8401

Table E.4: Training data with corresponding target and calculated outputs by ANN model for Drying problem

Input variables		Target	Calculated
X_1	X_2	T	Y
550	0.1100	40.5341	43.8105
550	0.2100	66.1878	62.3471
550	0.3100	84.1429	80.4784
550	0.4100	96.7389	95.8044
550	0.5100	105.6957	107.4461
550	0.6100	112.1949	115.6555
550	0.7100	117.0201	121.1459
550	0.8100	120.6847	124.665
550	0.9100	123.5269	126.8262
550	1.0100	125.7729	128.0801
550	1.1100	127.5770	128.7393
550	1.2100	129.0470	129.0131
550	1.3100	130.2596	129.039
550	1.4100	131.2709	128.9065
550	1.5100	132.1223	128.6731
600	0.1100	42.9309	46.4596
600	0.2100	70.4669	66.2873
600	0.3100	89.9872	85.7314
600	0.4100	103.8401	102.1762
600	0.5100	113.7853	114.6614
600	0.6100	121.0560	123.4598
600	0.7100	126.4847	129.3423
600	0.8100	130.6256	133.1141
600	0.9100	133.8483	135.4337
600	1.0100	136.4021	136.7836
600	1.1100	138.4582	137.4978
600	1.2100	140.1371	137.7998
650	0.1100	45.2522	49.1893
650	0.2100	74.6233	70.3453
650	0.3100	95.6831	91.1372
650	0.4100	110.7851	108.7249
650	0.5100	121.7223	122.0665
650	0.6100	129.7740	131.4594
650	0.7100	135.8181	137.7365
650	0.8100	140.4475	141.7628
650	0.9100	144.0620	144.2424
650	1.0100	146.9337	145.6902
650	1.1100	149.2510	146.4616
700	0.1100	47.5058	51.9876
700	0.2100	78.6688	74.5019
700	0.3100	101.2438	96.6691
700	0.4100	117.5861	115.4161
700	0.5100	129.5174	129.6201
700	0.6100	138.3580	139.608
700	0.7100	145.0280	146.2785
700	0.8100	150.1567	150.5583
700	0.9100	154.1732	153.1981
750	0.1100	49.6982	54.8412
750	0.2100	82.6133	78.7362
750	0.3100	106.6802	102.2976
750	0.4100	124.2537	122.2123
750	0.5100	137.1799	137.278
750	0.6100	146.8160	147.8562
750	0.7100	154.1208	154.9152
750	0.8100	159.7587	159.4451
800	0.1100	51.8348	57.7354
800	0.2100	86.4653	83.0253
800	0.3100	112.0020	107.9913
800	0.4100	130.7974	129.074
800	0.5100	144.7183	144.994
800	0.6100	155.1551	156.1528
800	0.7100	163.1029	163.5919
850	0.1100	53.9204	60.6541
850	0.2100	90.2321	87.3448
850	0.3100	117.2175	113.7171
850	0.4100	137.2254	135.9605
850	0.5100	152.1399	152.7211
850	0.6100	163.3817	164.4462
900	0.1100	55.9590	63.5809
900	0.2100	93.9202	91.6696
900	0.3100	122.3342	119.4414

Table E.4 contd.

900	0.4100	143.5448	142.8308
900	0.5100	159.4514	160.4129
950	0.1100	57.9540	66.4983
950	0.2100	97.5350	95.9739
950	0.3100	127.3584	125.1305
950	0.4100	149.7623	149.6447
950	0.5100	166.6589	168.0243
1000	0.1100	59.9087	69.3888
1000	0.2100	101.0816	100.2321
1000	0.3100	132.2961	130.7513
1000	0.4100	155.8838	156.3636

Table E.5: Input data with corresponding outputs and predicted values by ANN model for Alkylation process problem (a)

Input variables			Target	Predicted
X_1	X_7	X_8	T	Y
150	92.8000	6.2000	14.18803	60.94547
150	92.8000	6.6000	34.1136	72.59843
150	92.8000	7.0000	49.35875	90.33539
150	92.8000	7.4000	61.24333	113.524
150	92.8000	7.8000	70.57528	138.3534
150	92.8000	8.2000	77.8758	159.8185
150	92.8000	8.6000	83.4975	175.0294
150	92.8000	9.0000	87.68628	183.8997
150	92.8000	9.4000	90.61922	187.6692
150	92.8000	9.8000	92.42725	187.5891
150	92.8000	10.2000	93.20882	184.4505
150	92.8000	10.6000	93.03999	178.608
150	92.8000	11.0000	91.9796	170.1721
150	92.8000	11.4000	90.07452	159.2391
150	92.8000	11.8000	87.36192	146.1106
150	93.0000	6.6000	27.87579	64.92143
150	93.0000	7.0000	45.43202	78.22981
150	93.0000	7.4000	58.89716	97.76957
150	93.0000	7.8000	69.35583	121.9994
150	93.0000	8.2000	77.48546	146.2746
150	93.0000	8.6000	83.73264	165.6488
150	93.0000	9.0000	88.40217	177.85
150	93.0000	9.4000	91.70984	183.1949
150	93.0000	9.8000	93.81286	182.9501
150	93.0000	10.2000	94.82783	178.2529
150	93.0000	10.6000	94.84382	169.8799
150	93.0000	11.0000	93.92896	158.4349
150	93.0000	11.4000	92.1371	144.6208
150	93.0000	11.8000	89.51076	129.4102
150	93.2000	6.6000	19.45156	60.3834
150	93.2000	7.0000	40.0675	69.79473
150	93.2000	7.4000	55.5567	84.81119
150	93.2000	7.8000	67.42111	105.8803
150	93.2000	8.2000	76.56415	130.3217
150	93.2000	8.6000	83.56313	152.6984
150	93.2000	9.0000	88.80306	168.2958
150	93.2000	9.4000	92.5506	175.5197
150	93.2000	9.8000	94.99689	175.1485
150	93.2000	10.2000	96.28171	168.73
150	93.2000	10.6000	96.51031	157.8087
150	93.2000	11.0000	95.76258	143.8685
150	93.2000	11.4000	94.10088	128.4368
150	93.2000	11.8000	91.57386	113.0315
150	93.4000	7.0000	32.75785	64.79782
150	93.4000	7.4000	50.91083	75.49926
150	93.4000	7.8000	64.56915	91.95302
150	93.4000	8.2000	74.97465	113.6407
150	93.4000	8.6000	82.89232	136.5089
150	93.4000	9.0000	88.81831	154.5351
150	93.4000	9.4000	93.08852	163.6793
150	93.4000	9.8000	95.93869	163.5079
150	93.4000	10.2000	97.53858	155.89
150	93.4000	10.6000	98.01419	143.4178
150	93.4000	11.0000	97.45988	128.6285
150	93.4000	11.4000	95.94858	113.6716
150	93.4000	11.8000	93.53682	100.0917
150	93.6000	7.4000	44.50439	69.83974
150	93.6000	7.8000	60.51474	81.43566
150	93.6000	8.2000	72.52802	98.31478
150	93.6000	8.6000	81.58964	118.5752
150	93.6000	9.0000	88.3548	136.8947
150	93.6000	9.4000	93.25492	147.5719
150	93.6000	9.8000	96.58626	148.4491
150	93.6000	10.2000	98.55814	141.2452
150	93.6000	10.6000	99.32334	129.3778
150	93.6000	11.0000	98.99483	116.0889
150	93.6000	11.4000	97.65891	103.6283
150	93.6000	11.8000	95.38152	93.16228
150	93.8000	7.8000	54.8399	74.6634
150	93.8000	8.2000	68.95721	86.02929

Table E.5 contd.

150	93.8000	8.6000	79.47544	101.3451
150	93.8000	9.0000	87.28626	117.4244
150	93.8000	9.4000	92.95819	128.8975
150	93.8000	9.8000	96.87113	132.1913
150	93.8000	10.2000	99.28786	127.9497
150	93.8000	10.6000	100.3966	119.409
150	93.8000	11.0000	100.3346	109.7946
150	93.8000	11.4000	99.20477	101.1299
150	93.8000	11.8000	97.0855	94.27324
150	94.0000	8.2000	63.87054	77.74318
150	94.0000	8.6000	76.29418	87.42603
150	94.0000	9.0000	85.43834	99.6933
150	94.0000	9.4000	92.07443	111.274
150	94.0000	9.8000	96.70232	118.3636
150	94.0000	10.2000	99.65899	119.7261
150	94.0000	10.6000	101.1806	117.0211
150	94.0000	11.0000	101.4364	112.7353
150	94.0000	11.4000	100.5517	108.6627
150	94.0000	11.8000	98.6199	105.6638
150	94.2000	8.6000	71.6712	78.76797
150	94.2000	9.0000	82.56187	87.23887
150	94.2000	9.4000	90.43063	98.58155
150	94.2000	9.8000	95.95506	110.0931
150	94.2000	10.2000	99.57861	118.9072
150	94.2000	10.6000	101.6043	124.1127
150	94.2000	11.0000	102.2448	126.6685
150	94.2000	11.4000	101.6546	128.0147
150	94.2000	11.8000	99.94775	129.2019
150	94.4000	9.0000	78.28967	81.58889
150	94.4000	9.4000	87.77902	92.41127
150	94.4000	9.8000	94.45484	107.6297
150	94.4000	10.2000	98.91924	124.1924
150	94.4000	10.6000	101.5713	138.7068
150	94.4000	11.0000	102.6846	149.6393
150	94.4000	11.4000	102.4539	157.3867
150	94.4000	11.8000	101.0202	163.0605
150	94.6000	9.4000	83.75025	91.10335
150	94.6000	9.8000	91.94844	107.8775
150	94.6000	10.2000	97.49979	130.157
150	94.6000	10.6000	100.9475	153.6712
150	94.6000	11.0000	102.6535	174.0424
150	94.6000	11.4000	102.8688	189.3964
150	94.6000	11.8000	101.7716	200.2158
150	94.8000	10.2000	95.0556	131.792
150	94.8000	10.6000	99.54092	161.5198
150	94.8000	11.0000	102.0068	191.06
150	94.8000	11.4000	102.7868	215.4208
150	94.8000	11.8000	102.1123	233.0365
150	95.0000	10.6000	97.06561	160.3122
150	95.0000	11.0000	100.5342	196.4475
150	95.0000	11.4000	102.0474	230.2363
150	95.0000	11.8000	101.9149	256.7215
350	92.8000	6.2000	33.1056	69.66626
350	92.8000	6.6000	79.59852	83.83635
350	92.8000	7.0000	115.1702	104.8984
350	92.8000	7.4000	142.9013	131.5258
350	92.8000	7.8000	164.6754	159.0284
350	92.8000	8.2000	181.7103	182.1844
350	92.8000	8.6000	194.8276	198.5534
350	92.8000	9.0000	204.6011	208.5672
350	92.8000	9.4000	211.4447	213.783
350	92.8000	9.8000	215.6635	215.6202
350	92.8000	10.2000	217.4873	214.966
350	92.8000	10.6000	217.0933	212.2092
350	92.8000	11.0000	214.619	207.3833
350	92.8000	11.4000	210.1739	200.3157
350	92.8000	11.8000	203.8446	190.7813
350	93.0000	6.6000	65.04363	76.87739
350	93.0000	7.0000	106.0078	93.61522
350	93.0000	7.4000	137.4269	117.575
350	93.0000	7.8000	161.83	146.3456
350	93.0000	8.2000	180.7995	174.3643
350	93.0000	8.6000	195.3763	196.5685
350	93.0000	9.0000	206.2715	211.1937
350	93.0000	9.4000	213.9894	219.0772

Table E.5 contd.

350	93.0000	9.8000	218.8966	221.7841
350	93.0000	10.2000	221.265	220.5391
350	93.0000	10.6000	221.3022	215.9981
350	93.0000	11.0000	219.1675	208.3903
350	93.0000	11.4000	214.9866	197.7658
350	93.0000	11.8000	208.8585	184.2619
350	93.2000	6.6000	45.38709	73.82723
350	93.2000	7.0000	93.49062	86.45594
350	93.2000	7.4000	129.6325	106.2666
350	93.2000	7.8000	157.3157	133.3996
350	93.2000	8.2000	178.6498	164.1008
350	93.2000	8.6000	194.9808	191.9709
350	93.2000	9.0000	207.2069	212.234
350	93.2000	9.4000	215.9512	223.7647
350	93.2000	9.8000	221.6593	227.694
350	93.2000	10.2000	224.6574	225.5515
350	93.2000	10.6000	225.1907	218.4703
350	93.2000	11.0000	223.446	207.198
350	93.2000	11.4000	219.5687	192.4102
350	93.2000	11.8000	213.6724	175.0222
350	93.4000	7.0000	76.43478	83.61358
350	93.4000	7.4000	118.7921	99.00293
350	93.4000	7.8000	150.6611	122.3063
350	93.4000	8.2000	174.9409	152.5156
350	93.4000	8.6000	193.4155	184.2156
350	93.4000	9.0000	207.2425	210.2401
350	93.4000	9.4000	217.2063	226.2648
350	93.4000	9.8000	223.8568	231.8743
350	93.4000	10.2000	227.5901	228.6506
350	93.4000	10.6000	228.6997	218.5223
350	93.4000	11.0000	227.4064	203.2996
350	93.4000	11.4000	223.88	184.8003
350	93.4000	11.8000	218.2527	164.9583
350	93.6000	7.4000	103.8438	96.33423
350	93.6000	7.8000	141.2008	114.8346
350	93.6000	8.2000	169.2321	141.542
350	93.6000	8.6000	190.3759	173.6618
350	93.6000	9.0000	206.161	203.8449
350	93.6000	9.4000	217.5946	224.5124
350	93.6000	9.8000	225.3678	232.3904
350	93.6000	10.2000	229.9691	228.4897
350	93.6000	10.6000	231.7544	215.8104
350	93.6000	11.0000	230.9879	197.7669
350	93.6000	11.4000	227.8708	177.5595
350	93.6000	11.8000	222.557	157.8365
350	93.8000	7.8000	127.9595	111.9665
350	93.8000	8.2000	160.9003	133.2539
350	93.8000	8.6000	185.4429	161.9296
350	93.8000	9.0000	203.6677	192.7838
350	93.8000	9.4000	216.9023	216.9322
350	93.8000	9.8000	226.0325	227.8358
350	93.8000	10.2000	231.6718	224.9547
350	93.8000	10.6000	234.2587	212.0762
350	93.8000	11.0000	234.1139	194.1667
350	93.8000	11.4000	231.4778	175.4198
350	93.8000	11.8000	226.5329	158.5594
350	94.0000	8.2000	149.0314	129.263
350	94.0000	8.6000	178.0199	151.6029
350	94.0000	9.0000	199.3559	178.947
350	94.0000	9.4000	214.8402	204.1251
350	94.0000	9.8000	225.6386	218.931
350	94.0000	10.2000	232.5377	220.3917
350	94.0000	10.6000	236.0882	211.6268
350	94.0000	11.0000	236.685	198.0368
350	94.0000	11.4000	234.6207	184.0584
350	94.0000	11.8000	230.1132	172.1944
350	94.2000	8.6000	167.2329	145.3009
350	94.2000	9.0000	192.6441	166.0835
350	94.2000	9.4000	211.0046	189.5256
350	94.2000	9.8000	223.895	208.7586
350	94.2000	10.2000	232.3502	218.5041
350	94.2000	10.6000	237.0766	218.9501
350	94.2000	11.0000	238.5711	213.9612
350	94.2000	11.4000	237.1941	207.502
350	94.2000	11.8000	233.2115	202.0122

Table E.5 contd.

350	94.4000	9.0000	182.6757	157.1529
350	94.4000	9.4000	204.8175	176.6846
350	94.4000	9.8000	220.3945	199.4327
350	94.4000	10.2000	230.8116	219.558
350	94.4000	10.6000	236.9996	233.0705
350	94.4000	11.0000	239.5974	240.1146
350	94.4000	11.4000	239.0592	243.1694
350	94.4000	11.8000	235.7138	244.6701
350	94.6000	9.4000	195.4171	165.5374
350	94.6000	9.8000	214.5463	188.894
350	94.6000	10.2000	227.4996	217.4086
350	94.6000	10.6000	235.5441	244.4869
350	94.6000	11.0000	239.5248	265.3919
350	94.6000	11.4000	240.0272	279.4491
350	94.6000	11.8000	237.4672	288.4366
350	94.8000	10.2000	221.7965	206.3172
350	94.8000	10.6000	232.2621	243.1622
350	94.8000	11.0000	238.0158	277.3611
350	94.8000	11.4000	239.8359	303.6863
350	94.8000	11.8000	238.262	321.573
350	95.0000	10.6000	226.4864	228.132
350	95.0000	11.0000	234.5797	271.7824
350	95.0000	11.4000	238.1105	310.3259
350	95.0000	11.8000	237.8015	338.9728
550	92.8000	6.2000	52.02279	89.01586
550	92.8000	6.6000	125.0833	107.8763
550	92.8000	7.0000	180.982	135.1444
550	92.8000	7.4000	224.559	168.3022
550	92.8000	7.8000	258.776	201.1645
550	92.8000	8.2000	285.545	227.9585
550	92.8000	8.6000	306.1576	246.6641
550	92.8000	9.0000	321.5161	258.3482
550	92.8000	9.4000	332.2704	265.0276
550	92.8000	9.8000	338.8997	268.4036
550	92.8000	10.2000	341.7659	269.5487
550	92.8000	10.6000	341.1466	268.9892
550	92.8000	11.0000	337.2585	266.8563
550	92.8000	11.4000	330.2733	263.0055
550	92.8000	11.8000	320.3274	257.1064
550	93.0000	6.6000	102.2113	102.9434
550	93.0000	7.0000	166.584	126.1605
550	93.0000	7.4000	215.9564	158.3681
550	93.0000	7.8000	254.3046	195.5235
550	93.0000	8.2000	284.1138	230.3707
550	93.0000	8.6000	307.0198	257.4082
550	93.0000	9.0000	324.1411	275.4436
550	93.0000	9.4000	336.2693	286.0524
550	93.0000	9.8000	343.9803	291.34
550	93.0000	10.2000	347.7023	292.8691
550	93.0000	10.6000	347.7606	291.4812
550	93.0000	11.0000	344.4061	287.4377
550	93.0000	11.4000	337.836	280.6125
550	93.0000	11.8000	328.2065	270.6797
550	93.2000	6.6000	71.32244	102.8327
550	93.2000	7.0000	146.9141	121.5064
550	93.2000	7.4000	203.7081	150.1644
550	93.2000	7.8000	247.2107	188.1101
550	93.2000	8.2000	280.7357	229.4309
550	93.2000	8.6000	306.3982	265.9094
550	93.2000	9.0000	325.611	292.5384
550	93.2000	9.4000	339.3521	308.9579
550	93.2000	9.8000	348.3217	317.1252
550	93.2000	10.2000	353.0332	319.0974
550	93.2000	10.6000	353.8711	316.1885
550	93.2000	11.0000	351.1294	308.9632
550	93.2000	11.4000	345.0365	297.5177
550	93.2000	11.8000	335.7712	281.8345
550	93.4000	7.0000	120.112	121.7588
550	93.4000	7.4000	186.6732	145.5263
550	93.4000	7.8000	236.7535	180.6566
550	93.4000	8.2000	274.9075	224.7245
550	93.4000	8.6000	303.9386	269.6236
550	93.4000	9.0000	325.6669	306.3913
550	93.4000	9.4000	341.3244	330.7309
550	93.4000	9.8000	351.775	343.0617

Table E.5 contd.

550	93.4000	10.2000	357.6417	345.5839
550	93.4000	10.6000	359.3853	340.2643
550	93.4000	11.0000	357.3528	328.3499
550	93.4000	11.4000	351.8114	310.6667
550	93.4000	11.8000	342.9687	288.128
550	93.6000	7.4000	163.1829	145.4455
550	93.6000	7.8000	221.8873	175.276
550	93.6000	8.2000	265.9365	217.3769
550	93.6000	8.6000	299.1621	266.8062
550	93.6000	9.0000	323.9674	313.0578
550	93.6000	9.4000	341.9345	346.8082
550	93.6000	9.8000	354.1494	364.7466
550	93.6000	10.2000	361.3801	368.1657
550	93.6000	10.6000	364.1855	359.7843
550	93.6000	11.0000	362.9809	342.2048
550	93.6000	11.4000	358.0827	317.8467
550	93.6000	11.8000	349.7326	289.302
550	93.8000	7.8000	201.0796	173.6962
550	93.8000	8.2000	252.8436	209.9435
550	93.8000	8.6000	291.4101	258.0937
550	93.8000	9.0000	320.0494	309.8373
550	93.8000	9.4000	340.8466	352.4633
550	93.8000	9.8000	355.1939	377.1688
550	93.8000	10.2000	364.0558	382.5038
550	93.8000	10.6000	368.1209	371.6801
550	93.8000	11.0000	367.8933	349.4155
550	93.8000	11.4000	363.7508	320.553
550	93.8000	11.8000	355.9805	289.5794
550	94.0000	8.2000	234.1924	205.5636
550	94.0000	8.6000	279.7454	247.0633
550	94.0000	9.0000	313.2737	297.7225
550	94.0000	9.4000	337.6061	345.5646
550	94.0000	9.8000	354.5749	377.1846
550	94.0000	10.2000	365.4165	386.6354
550	94.0000	10.6000	370.9956	376.5307
550	94.0000	11.0000	371.9335	353.7032
550	94.0000	11.4000	368.6896	325.3607
550	94.0000	11.8000	361.6067	297.177
550	94.2000	8.6000	262.7945	238.782
550	94.2000	9.0000	302.7266	281.7497
550	94.2000	9.4000	331.5788	328.514
550	94.2000	9.8000	351.835	365.3498
550	94.2000	10.2000	365.1218	382.0131
550	94.2000	10.6000	372.5489	378.4111
550	94.2000	11.0000	374.8975	361.6301
550	94.2000	11.4000	372.7335	339.86
550	94.2000	11.8000	366.4754	318.9787
550	94.4000	9.0000	287.0619	266.8592
550	94.4000	9.4000	321.8563	305.5179
550	94.4000	9.8000	346.3342	343.3841
550	94.4000	10.2000	362.7041	369.1536
550	94.4000	10.6000	372.428	378.1111
550	94.4000	11.0000	376.5101	373.9588
550	94.4000	11.4000	375.6644	363.549
550	94.4000	11.8000	370.4077	352.3359
550	94.6000	9.4000	307.0841	277.1363
550	94.6000	9.8000	337.144	310.2889
550	94.6000	10.2000	357.4995	343.3458
550	94.6000	10.6000	370.1407	367.7161
550	94.6000	11.0000	376.3961	380.5367
550	94.6000	11.4000	377.1856	384.5542
550	94.6000	11.8000	373.1629	384.0631
550	94.8000	10.2000	348.5375	302.7586
550	94.8000	10.6000	364.9833	340.0128
550	94.8000	11.0000	374.0247	370.5429
550	94.8000	11.4000	376.885	390.8307
550	94.8000	11.8000	374.412	402.387
550	95.0000	10.6000	355.9072	298.9641
550	95.0000	11.0000	368.6252	343.1172
550	95.0000	11.4000	374.1736	379.3894
550	95.0000	11.8000	373.6883	404.3952
750	92.8000	6.2000	70.94022	132.3378
750	92.8000	6.6000	170.5683	160.391
750	92.8000	7.0000	246.7932	199.5093
750	92.8000	7.4000	306.2167	244.7585

Table E.5 contd.

750	92.8000	7.8000	352.8761	287.2857
750	92.8000	8.2000	389.3796	320.4286
750	92.8000	8.6000	417.4876	342.8736
750	92.8000	9.0000	438.4308	356.7487
750	92.8000	9.4000	453.0962	364.8654
750	92.8000	9.8000	462.1358	369.413
750	92.8000	10.2000	466.0443	371.756
750	92.8000	10.6000	465.1996	372.611
750	92.8000	11.0000	459.8976	372.2503
750	92.8000	11.4000	450.3727	370.643
750	92.8000	11.8000	436.8098	367.533
750	93.0000	6.6000	139.3793	157.33
750	93.0000	7.0000	227.1596	192.4331
750	93.0000	7.4000	294.4858	239.1423
750	93.0000	7.8000	346.7789	290.2366
750	93.0000	8.2000	387.4279	335.7116
750	93.0000	8.6000	418.6634	369.6107
750	93.0000	9.0000	442.0103	391.7971
750	93.0000	9.4000	458.5493	405.0547
750	93.0000	9.8000	469.0638	412.3483
750	93.0000	10.2000	474.1394	415.7766
750	93.0000	10.6000	474.2188	416.5162
750	93.0000	11.0000	469.6444	415.042
750	93.0000	11.4000	460.6856	411.3375
750	93.0000	11.8000	447.554	405.0416
750	93.2000	6.6000	97.25812	158.8046
750	93.2000	7.0000	200.337	187.8299
750	93.2000	7.4000	277.7835	231.1201
750	93.2000	7.8000	337.1053	285.8373
750	93.2000	8.2000	382.8214	342.2609
750	93.2000	8.6000	417.8158	389.7236
750	93.2000	9.0000	444.0148	423.4025
750	93.2000	9.4000	462.7531	444:3524
750	93.2000	9.8000	474.984	455.8488
750	93.2000	10.2000	481.4088	460.8318
750	93.2000	10.6000	482.5512	461.1396
750	93.2000	11.0000	478.8126	457.5894
750	93.2000	11.4000	470.5045	450.2586
750	93.2000	11.8000	457.8695	438.7564
750	93.4000	7.0000	163.7887	186.7704
750	93.4000	7.4000	254.5542	223.2351
750	93.4000	7.8000	322.8455	275.3548
750	93.4000	8.2000	374.8738	337.6195
750	93.4000	8.6000	414.4618	397.9003
750	93.4000	9.0000	444.0911	445.5453
750	93.4000	9.4000	465.4427	477.1836
750	93.4000	9.8000	479.693	494.9056
750	*93.4000*	*10.2000*	*487.6932*	*502.1897*
750	93.4000	10.6000	490.0706	501.6921
750	93.4000	11.0000	487.2991	494.8018
750	93.4000	11.4000	479.743	481.9227
750	93.4000	11.8000	467.6843	462.9447
750	93.6000	7.4000	222.522	217.7571
750	93.6000	7.8000	302.5734	262.3858
750	93.6000	8.2000	362.6407	323.163
750	93.6000	8.6000	407.9484	391.3527
750	93.6000	9.0000	441.7735	452.8459
750	93.6000	9.4000	466.2747	497.7081
750	93.6000	9.8000	482.9308	524.1059
750	93.6000	10.2000	492.7909	534.8458
750	93.6000	10.6000	496.6164	533.3141
750	93.6000	11.0000	494.9738	521.8244
750	93.6000	11.4000	488.2946	501.6498
750	93.6000	11.8000	476.9078	473.6842
750	93.8000	7.8000	274.1992	251.1588
750	93.8000	8.2000	344.7867	304.1305
750	93.8000	8.6000	397.3774	372.298
750	93.8000	9.0000	436.4308	443.3519
750	93.8000	9.4000	464.7911	501.8762
750	93.8000	9.8000	484.3552	539.2357
750	93.8000	10.2000	496.4396	555.0986
750	93.8000	10.6000	501.9827	552.8989
750	93.8000	11.0000	501.6725	536.3278
750	93.8000	11.4000	496.0239	508.4835
750	93.8000	11.8000	485.4277	472.3587

Table E.5 contd.

750	94.0000	8.2000	319.3533	287.4354
750	94.0000	8.6000	381.4711	348.1264
750	94.0000	9.0000	427.1912	420.8896
750	94.0000	9.4000	460.3723	489.7644
750	94.0000	9.8000	483.5111	538.9861
750	94.0000	10.2000	498.2952	562.0402
750	94.0000	10.6000	505.9029	560.7902
750	94.0000	11.0000	507.1819	540.5858
750	94.0000	11.4000	502.7586	507.4391
750	94.0000	11.8000	493.0997	467.2867
750	94.2000	8.6000	358.3562	328.4073
750	94.2000	9.0000	412.8088	394.4722
750	94.2000	9.4000	452.1532	466.3214
750	94.2000	9.8000	479.7748	525.1262
750	94.2000	10.2000	497.8933	556.9744
750	94.2000	10.6000	508.021	559.8141
750	94.2000	11.0000	511.2236	540.0346
750	94.2000	11.4000	508.273	506.8241
750	94.2000	11.8000	499.7389	468.7847
750	94.4000	9.0000	391.4478	372.5271
750	94.4000	9.4000	438.8952	437.1312
750	94.4000	9.8000	472.2738	498.2564
750	94.4000	10.2000	494.5964	537.9223
750	94.4000	10.6000	507.8561	548.5777
750	94.4000	11.0000	513.4227	535.2431
750	94.4000	11.4000	512.2697	508.518
750	94.4000	11.8000	505.101	478.1382
750	94.6000	9.4000	418.7514	400.7053
750	94.6000	9.8000	459.7417	452.7371
750	94.6000	10.2000	487.4992	495.3175
750	94.6000	10.6000	504.7371	516.7283
750	94.6000	11.0000	513.2673	516.7184
750	94.6000	11.4000	514.3441	503.171
750	94.6000	11.8000	508.8582	484.7478
750	94.8000	10.2000	475.2783	425.181
750	94.8000	10.6000	497.7043	458.0505
750	94.8000	11.0000	510.0335	476.8231
750	94.8000	11.4000	513.9341	482.1739
750	94.8000	11.8000	510.5615	479.2813
750	95.0000	10.6000	485.3277	383.587
750	95.0000	11.0000	502.6705	420.5483
750	95.0000	11.4000	510.2369	446.6897
750	95.0000	11.8000	509.5746	461.3831
950	92.8000	6.2000	89.85741	215.2459
950	92.8000	6.6000	216.0529	259.2934
950	92.8000	7.0000	312.6051	317.3892
950	92.8000	7.4000	387.8743	380.0109
950	92.8000	7.8000	446.9763	434.5702
950	92.8000	8.2000	493.2137	474.0451
950	92.8000	8.6000	528.8176	498.8365
950	92.8000	9.0000	555.346	512.8749
950	92.8000	9.4000	573.9216	520.1771
950	92.8000	9.8000	585.3723	523.6189
950	92.8000	10.2000	590.3231	524.94
950	92.8000	10.6000	589.2531	525.0702
950	92.8000	11.0000	582.5374	524.4215
950	92.8000	11.4000	570.4719	523.0775
950	92.8000	11.8000	553.2925	520.8949
950	93.0000	6.6000	176.5468	249.829
950	93.0000	7.0000	287.7359	302.4719
950	93.0000	7.4000	373.0152	368.4541
950	93.0000	7.8000	439.2531	435.559
950	93.0000	8.2000	490.7415	490.9658
950	93.0000	8.6000	530.3068	529.4124
950	93.0000	9.0000	559.8799	552.8338
950	93.0000	9.4000	580.8288	565.7338
950	93.0000	9.8000	594.1479	572.1245
950	93.0000	10.2000	600.5768	574.6904
950	93.0000	10.6000	600.6772	574.95
950	93.0000	11.0000	594.8834	573.6101
950	93.0000	11.4000	583.5349	570.8408
950	93.0000	11.8000	566.9018	566.4415
950	93.2000	6.6000	123.1933	243.0739
950	93.2000	7.0000	253.7605	285.9303
950	93.2000	7.4000	351.8589	347.186

Table E.5 contd.

950	93.2000	7.8000	426.9998	419.9224
950	93.2000	8.2000	484.9065	489.702
950	93.2000	8.6000	529.2332	544.3971
950	93.2000	9.0000	562.4189	580.8009
950	93.2000	9.4000	586.1536	602.1484
950	93.2000	9.8000	601.6467	613.2349
950	93.2000	10.2000	609.7847	617.8757
950	93.2000	10.6000	611.2317	618.4203
950	93.2000	11.0000	606.4963	616.0256
950	93.2000	11.4000	595.9721	611.0138
950	93.2000	11.8000	579.9681	603.1325
950	93.4000	7.0000	207.4661	271.5388
950	93.4000	7.4000	322.4351	322.2798
950	93.4000	7.8000	408.9374	391.5977
950	93.4000	8.2000	474.8397	469.3889
950	93.4000	8.6000	524.9847	539.7936
950	93.4000	9.0000	562.5156	592.1905
950	93.4000	9.4000	589.5604	625.4396
950	93.4000	9.8000	607.6115	643.7047
950	93.4000	10.2000	617.745	651.7417
950	93.4000	10.6000	620.7562	652.928
950	93.4000	11.0000	617.2459	649.0723
950	93.4000	11.4000	607.6743	640.7714
950	93.4000	11.8000	592.4002	627.8005
950	93.6000	7.4000	281.861	299.6024
950	93.6000	7.8000	383.2595	358.3339
950	93.6000	8.2000	459.3444	434.7138
950	93.6000	8.6000	516.7344	515.5391
950	93.6000	9.0000	559.58	584.4644
950	93.6000	9.4000	590.6143	632.8997
950	93.6000	9.8000	611.7127	661.5027
950	93.6000	10.2000	624.2021	674.8878
950	93.6000	10.6000	629.0475	677.3985
950	93.6000	11.0000	626.9672	671.7131
950	93.6000	11.4000	618.5063	658.9422
950	93.6000	11.8000	604.0833	639.1579
950	93.8000	7.8000	347.3189	328.1523
950	93.8000	8.2000	436.7293	394.8405
950	93.8000	8.6000	503.3445	477.0246
950	93.8000	9.0000	552.8126	558.7122
950	93.8000	9.4000	588.735	623.9167
950	93.8000	9.8000	613.5169	666.2287
950	93.8000	10.2000	628.8238	687.6179
950	93.8000	10.6000	635.845	692.6587
950	93.8000	11.0000	635.4522	685.0383
950	93.8000	11.4000	628.2968	666.8273
950	93.8000	11.8000	614.8752	639.0305
950	94.0000	8.2000	404.5137	360.0036
950	94.0000	8.6000	483.1965	434.6711
950	94.0000	9.0000	541.109	521.2367
950	94.0000	9.4000	583.1379	601.2654
950	94.0000	9.8000	612.4478	659.7136
950	94.0000	10.2000	631.1742	692.2257
950	94.0000	10.6000	640.8105	701.6828
950	94.0000	11.0000	642.4308	692.6207
950	94.0000	11.4000	636.8273	668.7901
950	94.0000	11.8000	624.5931	633.2342
950	94.2000	8.6000	453.9176	400.7262
950	94.2000	9.0000	522.8914	483.1702
950	94.2000	9.4000	572.7272	571.6693
950	94.2000	9.8000	607.7151	645.6573
950	94.2000	10.2000	630.6652	691.6411
950	94.2000	10.6000	643.4934	707.6554
950	94.2000	11.0000	647.5503	698.2824
950	94.2000	11.4000	643.8123	669.8139
950	94.2000	11.8000	633.0027	628.6368
950	94.4000	9.0000	495.8341	455.3665
950	94.4000	9.4000	555.9336	542.0402
950	94.4000	9.8000	598.2138	624.9625
950	94.4000	10.2000	626.4891	682.9103
950	94.4000	10.6000	643.2845	706.2706
950	94.4000	11.0000	650.3358	698.0077
950	94.4000	11.4000	648.8748	667.2654
950	94.4000	11.8000	639.7947	624.3436
950	94.6000	9.4000	530.4181	510.3775

Table E.5 contd.

950	94.6000	9.8000	582.3399	587.8547
950	94.6000	10.2000	617.4993	649.4164
950	94.6000	10.6000	639.3338	678.7135
950	94.6000	11.0000	650.1389	674.7035
950	94.6000	11.4000	651.5024	647.4006
950	94.6000	11.8000	644.5538	609.3281
950	94.8000	10.2000	602.0194	573.6666
950	94.8000	10.6000	630.4255	607.6189
950	94.8000	11.0000	646.0428	615.0174
950	94.8000	11.4000	650.983	601.3654
950	94.8000	11.8000	646.7113	577.1805
950	95.0000	10.6000	614.7487	503.1693
950	95.0000	11.0000	636.7164	528.104
950	95.0000	11.4000	646.2999	536.8371
950	95.0000	11.8000	645.4614	533.4717
1150	92.8000	6.2000	108.7748	323.9915
1150	92.8000	6.6000	261.5382	389.4791
1150	92.8000	7.0000	378.4165	470.3152
1150	92.8000	7.4000	469.5321	550.9975
1150	92.8000	7.8000	541.0763	615.8385
1150	92.8000	8.2000	597.048	658.8594
1150	92.8000	8.6000	640.1475	682.9177
1150	92.8000	9.0000	672.2609	693.8886
1150	92.8000	9.4000	694.7471	696.9756
1150	92.8000	9.8000	708.6086	695.7514
1150	92.8000	10.2000	714.6015	692.3857
1150	92.8000	10.6000	713.3062	688.0942
1150	92.8000	11.0000	705.1769	683.4951
1150	92.8000	11.4000	690.5714	678.8364
1150	92.8000	11.8000	669.7754	674.1245
1150	93.0000	6.6000	213.715	359.7586
1150	93.0000	7.0000	348.3116	432.8964
1150	93.0000	7.4000	451.5448	518,3781
1150	93.0000	7.8000	531.7271	598.6995
1150	93.0000	8.2000	594.0554	659.748
1150	93.0000	8.6000	641.9501	698.349
1150	93.0000	9.0000	677.7494	718.9179
1150	93.0000	9.4000	703.1085	727.591
1150	93.0000	9.8000	719.2317	729.2923
1150	93.0000	10.2000	727.0139	727.2287
1150	93.0000	10.6000	727.1356	723.2567
1150	93.0000	11.0000	720.1221	718.336
1150	93.0000	11.4000	706.3845	712.8577
1150	93.0000	11.8000	686.2498	706.8409
1150	93.2000	6.6000	149.1293	333.9842
1150	93.2000	7.0000	307.1837	393.0267
1150	93.2000	7.4000	425.9346	472.649
1150	93.2000	7.8000	516.8942	560.6556
1150	93.2000	8.2000	586.992	638.6732
1150	93.2000	8.6000	640.6505	694.9867
1150	93.2000	9.0000	680.8229	729.0382
1150	93.2000	9.4000	709.5543	746.3034
1150	93.2000	9.8000	728.3093	752.8556
1150	93.2000	10.2000	738.1603	753.1872
1150	93.2000	10.6000	739.912	750.0588
1150	93.2000	11.0000	734.1798	744.9424
1150	93.2000	11.4000	721.4401	738.4532
1150	93.2000	11.8000	702.067	730.6371
1150	93.4000	7.0000	251.143	357.9041
1150	93.4000	7.4000	390.3163	423.7683
1150	93.4000	7.8000	495.0292	508.5616
1150	93.4000	8.2000	574.8058	597.1609
1150	93.4000	8.6000	635.5076	671.4627
1150	93.4000	9.0000	680.9399	722.5839
1150	93.4000	9.4000	713.6783	752.1703
1150	93.4000	9.8000	735.5297	766.2708
1150	93.4000	10.2000	747.7964	770.622
1150	93.4000	10.6000	751.4417	769.144
1150	93.4000	11.0000	747.1925	764.0412
1150	93.4000	11.4000	735.6058	756.2729
1150	93.4000	11.8000	717.1163	745.9496
1150	93.6000	7.4000	341.2003	380.4325
1150	93.6000	7.8000	463.9455	452.8053
1150	93.6000	8.2000	556.0483	541.5124
1150	93.6000	8.6000	625.5205	629.1504

				Table E.5 contd.
1150	93.6000	9.0000	677.3862	698.8537
1150	93.6000	9.4000	714.9541	744.6351
1150	93.6000	9.8000	740.4944	769.8068
1150	93.6000	10.2000	755.6129	780.5339
1150	93.6000	10.6000	761.4786	781.9583
1150	93.6000	11.0000	758.9603	777.2522
1150	93.6000	11.4000	748.7183	767.8932
1150	93.6000	11.8000	731.259	754.1524
1150	93.8000	7.8000	420.4384	403.4429
1150	93.8000	8.2000	528.6722	482.1949
1150	93.8000	8.6000	609.3116	573.8647
1150	93.8000	9.0000	669.1942	659.5068
1150	93.8000	9.4000	712.6792	724.1642
1150	93.8000	9.8000	742.6784	764.4961
1150	93.8000	10.2000	761.2076	784.8752
1150	93.8000	10.6000	769.707	791.1369
1150	93.8000	11.0000	769.2317	787.5515
1150	93.8000	11.4000	760.5699	776.3529
1150	93.8000	11.8000	744.3228	758.2013
1150	94.0000	8.2000	489.6744	430.7508
1150	94.0000	8.6000	584.9219	516.3025
1150	94.0000	9.0000	655.0267	610.5771
1150	94.0000	9.4000	705.9037	693.7182
1150	94.0000	9.8000	741.3842	752.969
1150	94.0000	10.2000	764.0529	787.1375
1150	94.0000	10.6000	775.7179	801.0228
1150	94.0000	11.0000	777.6794	799.7368
1150	94.0000	11.4000	770.8964	786.5603
1150	94.0000	11.8000	756.0866	763.0295
1150	94.2000	8.6000	549.4791	469.6543
1150	94.2000	9.0000	632.9738	563.1865
1150	94.2000	9.4000	693.3011	660.018
1150	94.2000	9.8000	735.6552	739.7092
1150	94.2000	10.2000	763.4366	791.6608
1150	94.2000	10.6000	778.9656	816.28
1150	94.2000	11.0000	783.8767	818.4322
1150	94.2000	11.4000	779.3519	802.7458
1150	94.2000	11.8000	766.2667	772.5634
1150	94.4000	9.0000	600.2202	529.7573
1150	94.4000	9.4000	672.9722	631.0647
1150	94.4000	9.8000	724.1536	727.2038
1150	94.4000	10.2000	758.3814	797.5599
1150	94.4000	10.6000	778.7128	834.0708
1150	94.4000	11.0000	787.2487	839.1761
1150	94.4000	11.4000	785.4801	819.1189
1150	94.4000	11.8000	774.4886	780.5335
1150	94.6000	9.4000	642.085	606.9122
1150	94.6000	9.8000	704.9378	706.3679
1150	94.6000	10.2000	747.499	787.5449
1150	94.6000	10.6000	773.9304	832.4565
1150	94.6000	11.0000	787.0104	838.4287
1150	94.6000	11.4000	788.6608	813.0264
1150	94.6000	11.8000	780.2497	767.2707
1150	94.8000	10.2000	728.7602	732.7505
1150	94.8000	10.6000	763.1467	779.7297
1150	94.8000	11.0000	782.052	787.9188
1150	94.8000	11.4000	788.0322	763.5405
1150	94.8000	11.8000	782.8613	720.0261
1150	95.0000	10.6000	744.1694	669.0234
1150	95.0000	11.0000	770.762	686.0795
1150	95.0000	11.4000	782.3631	675.9968
1150	95.0000	11.8000	781.3482	648.7296
1350	92.8000	6.6000	307.0224	498.7138
1350	92.8000	7.0000	444.2281	597.7045
1350	92.8000	7.4000	551.1906	691.3554
1350	92.8000	7.8000	635.1766	763.3958
1350	92.8000	8.2000	700.8834	809.8311
1350	92.8000	8.6000	751.4777	835.2623
1350	92.8000	9.0000	789.1751	846.327
1350	92.8000	9.4000	815.5729	848.4346
1350	92.8000	9.8000	831.8445	845.2105
1350	92.8000	10.2000	838.8796	838.9
1350	92.8000	10.6000	837.3591	830.8646
1350	92.8000	11.0000	827.817	821.9364
1350	92.8000	11.4000	810.6708	812.6277

Table E.5 contd.

1350	92.8000	11.8000	786.2584	803.2405
1350	93.0000	7.0000	408.8877	537.2086
1350	93.0000	7.4000	530.0751	637.9451
1350	93.0000	7.8000	624.2015	727.7886
1350	93.0000	8.2000	697.3704	793.2446
1350	93.0000	8.6000	753.5939	833.2949
1350	93.0000	9.0000	795.6181	853.7666
1350	93.0000	9.4000	825.3885	861.3049
1350	93.0000	9.8000	844.3151	860.9615
1350	93.0000	10.2000	853.4507	856.0235
1350	93.0000	10.6000	853.5936	848.4941
1350	93.0000	11.0000	845.3611	839.5614
1350	93.0000	11.4000	829.234	829.9117
1350	93.0000	11.8000	805.5978	819.9058
1350	93.2000	7.0000	360.607	476.362
1350	93.2000	7.4000	500.011	572.1472
1350	93.2000	7.8000	606.789	672.4381
1350	93.2000	8.2000	689.0786	756.8837
1350	93.2000	8.6000	752.0683	815.2563
1350	93.2000	9.0000	799.2261	849.1006
1350	93.2000	9.4000	832.9552	865.0091
1350	93.2000	9.8000	854.9713	869.4224
1350	93.2000	10.2000	866.5356	866.9677
1350	93.2000	10.6000	868.592	860.5642
1350	93.2000	11.0000	861.8638	851.9332
1350	93.2000	11.4000	846.908	842.0267
1350	93.2000	11.8000	824.1658	831.2967
1350	93.4000	7.4000	458.1981	504.8163
1350	93.4000	7.8000	581.1214	603.8922
1350	93.4000	8.2000	674.773	701.8535
1350	93.4000	8.6000	746.031	779.8687
1350	93.4000	9.0000	799.3635	831.1105
1350	93.4000	9.4000	837.7965	859.1725
1350	93.4000	9.8000	863.4476	871.0118
1350	93.4000	10.2000	877.8475	872.6949
1350	93.4000	10.6000	882.1269	868.3571
1350	93.4000	11.0000	877.1395	860.4901
1350	93.4000	11.4000	863.5372	850.4473
1350	93.4000	11.8000	841.8325	838.8239
1350	93.6000	7.8000	544.6317	532.5204
1350	93.6000	8.2000	652.7534	633.3957
1350	93.6000	8.6000	734.3069	727.6172
1350	93.6000	9.0000	795.1919	798.7544
1350	93.6000	9.4000	839.2942	843.2167
1350	93.6000	9.8000	869.2756	866.0986
1350	93.6000	10.2000	887.0234	874.3709
1350	93.6000	10.6000	893.9092	873.5672
1350	93.6000	11.0000	890.954	867.2266
1350	93.6000	11.4000	878.9303	857.2907
1350	93.6000	11.8000	858.4348	844.582
1350	93.8000	7.8000	493.5582	469.0921
1350	93.8000	8.2000	620.6162	561.325
1350	93.8000	8.6000	715.2792	662.9216
1350	93.8000	9.0000	785.575	752.6885
1350	93.8000	9.4000	836.6237	817.1473
1350	93.8000	9.8000	871.8395	855.5529
1350	93.8000	10.2000	893.5911	873.9469
1350	93.8000	10.6000	903.5687	878.98
1350	93.8000	11.0000	903.0116	875.4542
1350	93.8000	11.4000	892.843	866.1301
1350	93.8000	11.8000	873.7706	852.191
1350	94.0000	8.2000	574.8362	497.6162
1350	94.0000	8.6000	686.6478	595.495
1350	94.0000	9.0000	768.9436	697.6059
1350	94.0000	9.4000	828.6699	783.1517
1350	94.0000	9.8000	870.3202	841.7879
1350	94.0000	10.2000	896.9312	875.0496
1350	94.0000	10.6000	910.625	889.3509
1350	94.0000	11.0000	912.9286	890.6474
1350	94.0000	11.4000	904.9653	882.7332
1350	94.0000	11.8000	887.5802	867.3604
1350	94.2000	8.6000	645.041	538.8825
1350	94.2000	9.0000	743.0554	643.8503
1350	94.2000	9.4000	813.8756	747.3399
1350	94.2000	9.8000	863.5948	829.506

Table E.5 çontd.

1350	94.2000	10.2000	896.2077	882.958
1350	94.2000	10.6000	914.4375	910.806
1350	94.2000	11.0000	920.2036	919.2263
1350	94.2000	11.4000	914.8914	913.106
1350	94.2000	11.8000	899.5308	895.1171
1350	94.4000	9.0000	704.6057	604.9757
1350	94.4000	9.4000	790.0111	718.4966
1350	94.4000	9.8000	850.0929	822.9168
1350	94.4000	10.2000	890.2734	899.6764
1350	94.4000	10.6000	914.1407	944.061
1350	94.4000	11.0000	924.162	960.1935
1350	94.4000	11.4000	922.0853	953.8234
1350	94.4000	11.8000	909.1826	929.2888
1350	94.6000	9.4000	753.7522	699.5866
1350	94.6000	9.8000	827.5353	816.0137
1350	94.6000	10.2000	877.4984	911.6486
1350	94.6000	10.6000	908.5267	970.5427
1350	94.6000	11.0000	923.8823	991.3107
1350	94.6000	11.4000	925.8193	979.908
1350	94.6000	11.8000	915.9455	943.6214
1350	94.8000	10.2000	855.5007	887.067
1350	94.8000	10.6000	895.8674	953.0082
1350	94.8000	11.0000	918.0615	974.8963
1350	94.8000	11.4000	925.0813	957.0468
1350	94.8000	11.8000	919.0114	910.2843
1350	95.0000	10.6000	873.5898	866.6059
1350	95.0000	11.0000	904.8081	890.3937
1350	95.0000	11.4000	918.4263	873.8281
1350	95.0000	11.8000	917.2352	828.7033
1550	92.8000	7.4000	632.8478	774.3546
1550	92.8000	7.8000	729.2773	848.0393
1550	92.8000	8.2000	804.7169	895.5982
1550	92.8000	8.6000	862.8068	922.8477
1550	92.8000	9.0000	906.0906	936.5603
1550	92.8000	9.4000	936.3978	941.8483
1550	92.8000	9.8000	955.0809	941.9582
1550	92.8000	10.2000	963.1583	938.7849
1550	92.8000	10.6000	961.4124	933.3923
1550	92.8000	11.0000	950.4558	926.3759
1550	92.8000	11.4000	930.7702	918.0859
1550	93.0000	7.4000	608.604	710.0808
1550	93.0000	7.8000	716.6761	803.1894
1550	93.0000	8.2000	800.6833	869.8928
1550	93.0000	8.6000	865.2366	911.2895
1550	93.0000	9.0000	913.4882	933.961
1550	93.0000	9.4000	947.6675	944.4258
1550	93.0000	9.8000	969.3989	947.3313
1550	93.0000	10.2000	979.8881	945.5538
1550	93.0000	10.6000	980.0519	940.7562
1550	93.0000	11.0000	970.5992	933.875
1550	93.0000	11.4000	952.0835	925.442
1550	93.2000	7.8000	696.6841	740.4902
1550	93.2000	8.2000	791.1631	827.3844
1550	93.2000	8.6000	863.485	886.965
1550	93.2000	9.0000	917.6307	922.47
1550	93.2000	9.4000	956.3553	940.8646
1550	93.2000	9.8000	981.6339	948.29
1550	93.2000	10.2000	994.9116	948.9278
1550	93.2000	10.6000	997.2723	945.3116
1550	93.2000	11.0000	989.5467	938.8917
1550	93.2000	11.4000	972.3759	930.4779
1550	93.4000	7.8000	667.2139	664.8533
1550	93.4000	8.2000	774.7383	767.836
1550	93.4000	8.6000	856.5533	847.8513
1550	93.4000	9.0000	917.7885	900.4283
1550	93.4000	9.4000	961.9138	930.4244
1550	93.4000	9.8000	991.3658	944.8867
1550	93.4000	10.2000	1007.899	949.5014
1550	93.4000	10.6000	1012.812	947.992
1550	93.4000	11.0000	1007.085	942.5628
1550	93.4000	11.4000	991.4688	934.4437
1550	93.6000	8.2000	749.4564	694.7865
1550	93.6000	8.6000	843.0922	793.2089
1550	93.6000	9.0000	912.9988	866.119
1550	93.6000	9.4000	963.6333	912.092

Table E.5 contd.

1550	93.6000	9.8000	998.0573	937.1143
1550	93.6000	10.2000	1018.435	948.0808
1550	93.6000	10.6000	1026.34	950.1672
1550	93.6000	11.0000	1022.946	946.6303
1550	93.6000	11.4000	1009.142	939.3193
1550	93.8000	8.6000	821.2457	725.8292
1550	93.8000	9.0000	901.9572	819.1481
1550	93.8000	9.4000	960.5671	885.3002
1550	93.8000	9.8000	1001.001	925.4619
1550	93.8000	10.2000	1025.975	946.3055
1550	93.8000	10.6000	1037.431	954.38
1550	93.8000	11.0000	1036.791	954.2795
1550	93.8000	11.4000	1025.116	948.727
1550	94.0000	8.6000	788.3726	653.9819
1550	94.0000	9.0000	882.8619	762.7761
1550	94.0000	9.4000	951.435	851.426
1550	94.0000	9.8000	999.2565	911.9577
1550	94.0000	10.2000	1029.81	947.6263
1550	94.0000	10.6000	1045.532	965.3774
1550	94.0000	11.0000	1048.177	971.1995
1550	94.0000	11.4000	1039.034	968.9463
1550	94.2000	9.0000	853.1384	706.1169
1550	94.2000	9.4000	934.449	815.7389
1550	94.2000	9.8000	991.5348	901.2029
1550	94.2000	10.2000	1028.979	957.7623
1550	94.2000	10.6000	1049.91	990.1612
1550	94.2000	11.0000	1056.529	1005.154
1550	94.2000	11.4000	1050.431	1007.831
1550	94.4000	9.4000	907.049	787.2233
1550	94.4000	9.8000	976.0327	898.635
1550	94.4000	10.2000	1022.166	981.0715
1550	94.4000	10.6000	1049.569	1032.678
1550	94.4000	11.0000	1061.074	1059.05
1550	94.4000	11.4000	1058.691	1066.243
1550	94.6000	9.4000	865.4185	771.4285
1550	94.6000	9.8000	950.1332	901.64
1550	94.6000	10.2000	1007.498	1008.74
1550	94.6000	10.6000	1043.123	1079.86
1550	94.6000	11.0000	1060.753	1115.984
1550	94.6000	11.4000	1062.978	1123.261
1550	94.8000	10.2000	982.2419	1011.083
1550	94.8000	10.6000	1028.589	1096.098
1550	94.8000	11.0000	1054.07	1137.88
1550	94.8000	11.4000	1062.13	1140.487
1550	95.0000	10.6000	1003.01	1048.231
1550	95.0000	11.0000	1038.853	1093.379
1550	95.0000	11.4000	1054.49	1092.206
1750	92.8000	7.8000	823.3782	890.6698
1750	92.8000	8.2000	908.5516	936.2332
1750	93.0000	8.2000	903.9977	907.8267
1750	93.2000	8.2000	893.249	864.7073

Table E.6: Input data and corresponding outputs and predicted values by ANN model for Alkylation process problem (b)

Input variables							Target	Predicted
X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	T	Y
0	0	0.8700	0.9000	3	2.6000	1.4500	0	0.1093
0	0	0.8900	0.9000	3	2.6000	1.4500	0	-0.1361
0	0	0.9100	0.9000	3	2.6000	1.4500	0	-0.3661
0	0	0.9300	0.9000	3	2.6000	1.4500	0	-0.5204
0	0	0.8700	0.9000	3	2.6000	1.5350	0	-0.0921
0	0	0.8900	0.9000	3	2.6000	1.5350	0	-0.1798
0	0	0.9100	0.9000	3	2.6000	1.5350	0	-0.2218
0	0	0.9300	0.9000	3	2.6000	1.5350	0	-0.2436
0	0	0.8700	0.9000	3	4.0000	1.4500	0	-0.0394
0	0	0.8900	0.9000	3	4.0000	1.4500	0	-0.3262
0	0	0.9100	0.9000	3	4.0000	1.4500	0	-0.3165
0	0	0.9300	0.9000	3	4.0000	1.4500	0	-0.1323
0	0	0.8700	0.9000	3	4.0000	1.5350	0	0.4051
0	0	0.8900	0.9000	3	4.0000	1.5350	0	0.0979
0	0	0.9100	0.9000	3	4.0000	1.5350	0	-0.1488
0	0	0.9300	0.9000	3	4.0000	1.5350	0	-0.3129
0	0	0.8700	0.9000	6	2.6000	1.4500	0	0.0859
0	0	0.8900	0.9000	6	2.6000	1.4500	0	-0.1007
0	0	0.9100	0.9000	6	2.6000	1.4500	0	-0.3068
0	0	0.9300	0.9000	6	2.6000	1.4500	0	-0.4919
0	0	0.8700	0.9000	6	2.6000	1.5350	0	-0.2152
0	0	0.8900	0.9000	6	2.6000	1.5350	0	-0.3172
0	0	0.9100	0.9000	6	2.6000	1.5350	0	-0.3379
0	0	0.9300	0.9000	6	2.6000	1.5350	0	-0.3001
0	0	0.8700	0.9000	6	4.0000	1.4500	0	0.1753
0	0	0.8900	0.9000	6	4.0000	1.4500	0	-0.1885
0	0	0.9100	0.9000	6	4.0000	1.4500	0	-0.2976
0	0	0.9300	0.9000	6	4.0000	1.4500	0	-0.1937
0	0	0.8700	0.9000	6	4.0000	1.5350	0	0.2701
0	0	0.8900	0.9000	6	4.0000	1.5350	0	0.0518
0	0	0.9100	0.9000	6	4.0000	1.5350	0	-0.1396
0	0	0.9300	0.9000	6	4.0000	1.5350	0	-0.2895
0	0	0.8700	0.9000	9	2.6000	1.4500	0	0.1469
0	0	0.8900	0.9000	9	2.6000	1.4500	0	0.0184
0	0	0.9100	0.9000	9	2.6000	1.4500	0	-0.1605
0	0	0.9300	0.9000	9	2.6000	1.4500	0	-0.3639
0	0	0.8700	0.9000	9	2.6000	1.5350	0	-0.3711
0	0	0.8900	0.9000	9	2.6000	1.5350	0	-0.4376
0	0	0.9100	0.9000	9	2.6000	1.5350	0	-0.4082
0	0	0.9300	0.9000	9	2.6000	1.5350	0	-0.3070
0	0	0.8700	0.9000	9	4.0000	1.4500	0	0.3653
0	0	0.8900	0.9000	9	4.0000	1.4500	0	-0.0360
0	0	0.9100	0.9000	9	4.0000	1.4500	0	-0.2540
0	0	0.9300	0.9000	9	4.0000	1.4500	0	-0.2443
0	0	0.8700	0.9000	9	4.0000	1.5350	0	0.0947
0	0	0.8900	0.9000	9	4.0000	1.5350	0	-0.0327
0	0	0.9100	0.9000	9	4.0000	1.5350	0	-0.1662
0	0	0.9300	0.9000	9	4.0000	1.5350	0	-0.2973
0	0	0.8700	0.9000	12	2.6000	1.4500	0	0.2131
0	0	0.8900	0.9000	12	2.6000	1.4500	0	0.1423
0	0	0.9100	0.9000	12	2.6000	1.4500	0	0.0098
0	0	0.9300	0.9000	12	2.6000	1.4500	0	-0.1696
0	0	0.8700	0.9000	12	2.6000	1.5350	0	-0.5354
0	0	0.8900	0.9000	12	2.6000	1.5350	0	-0.5541
0	0	0.9100	0.9000	12	2.6000	1.5350	0	-0.4752
0	0	0.9300	0.9000	12	2.6000	1.5350	0	-0.3144
0	0	0.8700	0.9000	12	4.0000	1.4500	0	0.4559
0	0	0.8900	0.9000	12	4.0000	1.4500	0	0.0495
0	0	0.9100	0.9000	12	4.0000	1.4500	0	-0.2504
0	0	0.9300	0.9000	12	4.0000	1.4500	0	-0.3271
0	0	0.8700	0.9000	12	4.0000	1.5350	0	-0.1141
0	0	0.8900	0.9000	12	4.0000	1.5350	0	-0.1757
0	0	0.9100	0.9000	12	4.0000	1.5350	0	-0.2624
0	0	0.9300	0.9000	12	4.0000	1.5350	0	-0.3673
0	0	0.8700	0.9250	3	2.6000	1.4500	0	0.0339
0	0	0.8900	0.9250	3	2.6000	1.4500	0	-0.0371
0	0	0.9100	0.9250	3	2.6000	1.4500	0	-0.0950
0	0	0.9300	0.9250	3	2.6000	1.4500	0	-0.1918
0	0	0.8700	0.9250	3	2.6000	1.5350	0	-0.0766
0	0	0.8900	0.9250	3	2.6000	1.5350	0	-0.0514
0	0	0.9100	0.9250	3	2.6000	1.5350	0	0.0410

Table E.6 contd.

0	0	0.9300	0.9250	3	2.6000	1.5350	0	0.1148
0	0	0.8700	0.9250	3	4.0000	1.4500	0	0.2643
0	0	0.8900	0.9250	3	4.0000	1.4500	0	0.0144
0	0	0.9100	0.9250	3	4.0000	1.4500	0	-0.0914
0	0	0.9300	0.9250	3	4.0000	1.4500	0	-0.1230
0	0	0.8700	0.9250	3	4.0000	1.5350	0	0.6446
0	0	0.8900	0.9250	3	4.0000	1.5350	0	0.4741
0	0	0.9100	0.9250	3	4.0000	1.5350	0	0.1900
0	0	0.9300	0.9250	3	4.0000	1.5350	0	-0.1977
0	0	0.8700	0.9250	6	2.6000	1.4500	0	-0.1378
0	0	0.8900	0.9250	6	2.6000	1.4500	0	-0.1922
0	0	0.9100	0.9250	6	2.6000	1.4500	0	-0.2186
0	0	0.9300	0.9250	6	2.6000	1.4500	0	-0.2689
0	0	0.8700	0.9250	6	2.6000	1.5350	0	-0.2036
0	0	0.8900	0.9250	6	2.6000	1.5350	0	-0.2826
0	0	0.9100	0.9250	6	2.6000	1.5350	0	-0.2382
0	0	0.9300	0.9250	6	2.6000	1.5350	0	-0.1093
0	0	0.8700	0.9250	6	4.0000	1.4500	0	0.2836
0	0	0.8900	0.9250	6	4.0000	1.4500	0	-0.0005
0	0	0.9100	0.9250	6	4.0000	1.4500	0	-0.1756
0	0	0.9300	0.9250	6	4.0000	1.4500	0	-0.2625
0	0	0.8700	0.9250	6	4.0000	1.5350	0	0.3652
0	0	0.8900	0.9250	6	4.0000	1.5350	0	0.2678
0	0	0.9100	0.9250	6	4.0000	1.5350	0	0.0941
0	0	0.9300	0.9250	6	4.0000	1.5350	0	-0.2091
0	0	0.8700	0.9250	9	2.6000	1.4500	0	-0.1840
0	0	0.8900	0.9250	9	2.6000	1.4500	0	-0.2110
0	0	0.9100	0.9250	9	2.6000	1.4500	0	-0.2039
0	0	0.9300	0.9250	9	2.6000	1.4500	0	-0.1993
0	0	0.8700	0.9250	9	2.6000	1.5350	0	-0.3752
0	0	0.8900	0.9250	9	2.6000	1.5350	0	-0.4806
0	0	0.9100	0.9250	9	2.6000	1.5350	0	-0.4166
0	0	0.9300	0.9250	9	2.6000	1.5350	0	-0.2003
0	0	0.8700	0.9250	9	4.0000	1.4500	0	0.2957
0	0	0.8900	0.9250	9	4.0000	1.4500	0	0.0175
0	0	0.9100	0.9250	9	4.0000	1.4500	0	-0.1917
0	0	0.9300	0.9250	9	4.0000	1.4500	0	-0.3072
0	0	0.8700	0.9250	9	4.0000	1.5350	0	0.0140
0	0	0.8900	0.9250	9	4.0000	1.5350	0	0.0136
0	0	0.9100	0.9250	9	4.0000	1.5350	0	-0.0184
0	0	0.9300	0.9250	9	4.0000	1.5350	0	-0.1795
0	0	0.8700	0.9250	12	2.6000	1.4500	0	-0.1582
0	0	0.8900	0.9250	12	2.6000	1.4500	0	-0.1275
0	0	0.9100	0.9250	12	2.6000	1.4500	0	-0.0611
0	0	0.9300	0.9250	12	2.6000	1.4500	0	0.0163
0	0	0.8700	0.9250	12	2.6000	1.5350	0	-0.5724
0	0	0.8900	0.9250	12	2.6000	1.5350	0	-0.6332
0	0	0.9100	0.9250	12	2.6000	1.5350	0	-0.5022
0	0	0.9300	0.9250	12	2.6000	1.5350	0	-0.1889
0	0	0.8700	0.9250	12	4.0000	1.4500	0	0.2546
0	0	0.8900	0.9250	12	4.0000	1.4500	0	0.0324
0	0	0.9100	0.9250	12	4.0000	1.4500	0	-0.1524
0	0	0.9300	0.9250	12	4.0000	1.4500	0	-0.2524
0	0	0.8700	0.9250	12	4.0000	1.5350	0	-0.3463
0	0	0.8900	0.9250	12	4.0000	1.5350	0	-0.2471
0	0	0.9100	0.9250	12	4.0000	1.5350	0	-0.1346
0	0	0.9300	0.9250	12	4.0000	1.5350	0	-0.1256
0	0	0.8700	0.9500	3	2.6000	1.4500	0	-0.0315
0	0	0.8900	0.9500	3	2.6000	1.4500	0	0.0546
0	0	0.9100	0.9500	3	2.6000	1.4500	0	0.1507
0	0	0.9300	0.9500	3	2.6000	1.4500	0	0.1147
0	0	0.8700	0.9500	3	2.6000	1.5350	0	-0.0264
0	0	0.8900	0.9500	3	2.6000	1.5350	0	0.1195
0	0	0.9100	0.9500	3	2.6000	1.5350	0	0.3301
0	0	0.9300	0.9500	3	2.6000	1.5350	0	0.4621
0	0	0.8700	0.9500	3	4.0000	1.4500	0	0.4198
0	0	0.8900	0.9500	3	4.0000	1.4500	0	0.1711
0	0	0.9100	0.9500	3	4.0000	1.4500	0	-0.0932
0	0	0.9300	0.9500	3	4.0000	1.4500	0	-0.3392
0	0	0.8700	0.9500	3	4.0000	1.5350	0	0.7631
0	0	0.8900	0.9500	3	4.0000	1.5350	0	0.7056
0	0	0.9100	0.9500	3	4.0000	1.5350	0	0.3868
0	0	0.9300	0.9500	3	4.0000	1.5350	0	-0.1704
0	0	0.8700	0.9500	6	2.6000	1.4500	0	-0.3302
0	0	0.8900	0.9500	6	2.6000	1.4500	0	-0.2279

Table E.6 contd.

0	0	0.9100	0.9500	6	2.6000	1.4500	0	-0.0356
0	0	0.9300	0.9500	6	2.6000	1.4500	0	0.0943
0	0	0.8700	0.9500	6	2.6000	1.5350	0	-0.1515
0	0	0.8900	0.9500	6	2.6000	1.5350	0	-0.1341
0	0	0.9100	0.9500	6	2.6000	1.5350	0	0.0263
0	0	0.9300	0.9500	6	2.6000	1.5350	0	0.2463
0	0	0.8700	0.9500	6	4.0000	1.4500	0	0.2269
0	0	0.8900	0.9500	6	4.0000	1.4500	0	0.0485
0	0	0.9100	0.9500	6	4.0000	1.4500	0	-0.1663
0	0	0.9300	0.9500	6	4.0000	1.4500	0	-0.3949
0	0	0.8700	0.9500	6	4.0000	1.5350	0	0.3698
0	0	0.8900	0.9500	6	4.0000	1.5350	0	0.3969
0	0	0.9100	0.9500	6	4.0000	1.5350	0	0.2625
0	0	0.9300	0.9500	6	4.0000	1.5350	0	-0.1159
0	0	0.8700	0.9500	9	2.6000	1.4500	0	-0.4371
0	0	0.8900	0.9500	9	2.6000	1.4500	0	-0.2956
0	0	0.9100	0.9500	9	2.6000	1.4500	0	-0.0324
0	0	0.9300	0.9500	9	2.6000	1.4500	0	0.2217
0	0	0.8700	0.9500	9	2.6000	1.5350	0	-0.2652
0	0	0.8900	0.9500	9	2.6000	1.5350	0	-0.3000
0	0	0.9100	0.9500	9	2.6000	1.5350	0	-0.1340
0	0	0.9300	0.9500	9	2.6000	1.5350	0	0.1872
0	0	0.8700	0.9500	9	4.0000	1.4500	0	0.1479
0	0	0.8900	0.9500	9	4.0000	1.4500	0	0.0779
0	0	0.9100	0.9500	9	4.0000	1.4500	0	-0.0515
0	0	0.9300	0.9500	9	4.0000	1.4500	0	-0.2431
0	0	0.8700	0.9500	9	4.0000	1.5350	0	-0.0357
0	0	0.8900	0.9500	9	4.0000	1.5350	0	0.1305
0	0	0.9100	0.9500	9	4.0000	1.5350	0	0.2304
0	0	0.9300	0.9500	9	4.0000	1.5350	0	0.0925
0	0	0.8700	0.9500	12	2.6000	1.4500	0	-0.3948
0	0	0.8900	0.9500	12	2.6000	1.4500	0	-0.2058
0	0	0.9100	0.9500	12	2.6000	1.4500	0	0.1007
0	0	0.9300	0.9500	12	2.6000	1.4500	0	0.4332
0	0	0.8700	0.9500	12	2.6000	1.5350	0	-0.3994
0	0	0.8900	0.9500	12	2.6000	1.5350	0	-0.4177
0	0	0.9100	0.9500	12	2.6000	1.5350	0	-0.2081
0	0	0.9300	0.9500	12	2.6000	1.5350	0	0.2044
0	0	0.8700	0.9500	12	4.0000	1.4500	0	0.1608
0	0	0.8900	0.9500	12	4.0000	1.4500	0	0.2138
0	0	0.9100	0.9500	12	4.0000	1.4500	0	0.1964
0	0	0.9300	0.9500	12	4.0000	1.4500	0	0.0729
0	0	0.8700	0.9500	12	4.0000	1.5350	0	-0.3583
0	0	0.8900	0.9500	12	4.0000	1.5350	0	-0.0567
0	0	0.9100	0.9500	12	4.0000	1.5350	0	0.2545
0	0	0.9300	0.9500	12	4.0000	1.5350	0	0.3625
1	0	0.8700	0.9000	3	2.6000	1.4500	-0.8125	-0.4771
1	0.5000	0.8700	0.9000	3	2.6000	1.4500	0.0975	0.2118
1	0	0.8900	0.9000	3	2.6000	1.4500	-0.8640	-0.8899
1	0.5000	0.8900	0.9000	3	2.6000	1.4500	0.0460	-0.1586
1	0	0.9100	0.9000	3	2.6000	1.4500	-0.9449	-1.2580
1	0.5000	0.9100	0.9000	3	2.6000	1.4500	-0.0349	-0.4298
1	0	0.9300	0.9000	3	2.6000	1.4500	-1.0905	-1.4203
1	0.5000	0.9300	0.9000	3	2.6000	1.4500	-0.1805	-0.4923
1	0	0.8700	0.9000	3	2.6000	1.5350	-0.8125	-0.6397
1	0.5000	0.8700	0.9000	3	2.6000	1.5350	0.0975	-0.1122
1	0	0.8900	0.9000	3	2.6000	1.5350	-0.8640	-0.8947
1	0.5000	0.8900	0.9000	3	2.6000	1.5350	0.0460	-0.2191
1	0	0.9100	0.9000	3	2.6000	1.5350	-0.9449	-1.1625
1	0.5000	0.9100	0.9000	3	2.6000	1.5350	-0.0349	-0.3417
1	0	0.9300	0.9000	3	2.6000	1.5350	-1.0905	-1.3960
1	0.5000	0.9300	0.9000	3	2.6000	1.5350	-0.1805	-0.4921
1	0	0.8700	0.9000	3	4.0000	1.4500	-0.9233	-0.9922
1	0.5000	0.8700	0.9000	3	4.0000	1.4500	-0.0133	-0.3568
1	0	0.8900	0.9000	3	4.0000	1.4500	-1.0025	-1.2162
1	0.5000	0.8900	0.9000	3	4.0000	1.4500	-0.0925	-0.4356
1	0	0.9100	0.9000	3	4.0000	1.4500	-1.1269	-1.0615
1	0.5000	0.9100	0.9000	3	4.0000	1.4500	-0.2169	-0.2348
1	0	0.9300	0.9000	3	4.0000	1.4500	-1.3509	-0.6880
1	0.5000	0.9300	0.9000	3	4.0000	1.4500	-0.4409	0.0368
1	0	0.8700	0.9000	3	4.0000	1.5350	-0.9233	-0.6036
1	0.5000	0.8700	0.9000	3	4.0000	1.5350	-0.0133	0.2217
1	0	0.8900	0.9000	3	4.0000	1.5350	-1.0025	-1.0534
1	0.5000	0.8900	0.9000	3	4.0000	1.5350	-0.0925	-0.0786
1	0	0.9100	0.9000	3	4.0000	1.5350	-1.1269	-1.2761

Table E.6 contd.

1	0.5000	0.9100	0.9000	3	4.0000	1.5350	-0.2169	-0.2367
1	0	0.9300	0.9000	3	4.0000	1.5350	-1.3509	-1.2439
1	0.5000	0.9300	0.9000	3	4.0000	1.5350	-0.4409	-0.2783
1	0	0.8700	0.9000	6	2.6000	1.4500	-0.9406	-0.5131
1	0.5000	0.8700	0.9000	6	2.6000	1.4500	0.0219	0.1618
1	1.0000	0.8700	0.9000	6	2.6000	1.4500	0.9844	0.9862
1	0	0.8900	0.9000	6	2.6000	1.4500	-0.9921	-0.8538
1	0.5000	0.8900	0.9000	6	2.6000	1.4500	-0.0296	-0.1621
1	1.0000	0.8900	0.9000	6	2.6000	1.4500	0.9329	0.7187
1	0	0.9100	0.9000	6	2.6000	1.4500	-1.0730	-1.2168
1	0.5000	0.9100	0.9000	6	2.6000	1.4500	-0.1105	-0.4415
1	1.0000	0.9100	0.9000	6	2.6000	1.4500	0.8520	0.5114
1	0	0.9300	0.9000	6	2.6000	1.4500	-1.2186	-1.4670
1	0.5000	0.9300	0.9000	6	2.6000	1.4500	-0.2561	-0.5666
1	1.0000	0.9300	0.9000	6	2.6000	1.4500	0.7064	0.4088
1	0	0.8700	0.9000	6	2.6000	1.5350	-0.9406	-0.6581
1	0.5000	0.8700	0.9000	6	2.6000	1.5350	0.0219	-0.0916
1	1.0000	0.8700	0.9000	6	2.6000	1.5350	0.9844	0.6085
1	0	0.8900	0.9000	6	2.6000	1.5350	-0.9921	-0.9545
1	0.5000	0.8900	0.9000	6	2.6000	1.5350	-0.0296	-0.2719
1	1.0000	0.8900	0.9000	6	2.6000	1.5350	0.9329	0.6098
1	0	0.9100	0.9000	6	2.6000	1.5350	-1.0730	-1.2277
1	0.5000	0.9100	0.9000	6	2.6000	1.5350	-0.1105	-0.4334
1	1.0000	0.9100	0.9000	6	2.6000	1.5350	0.8520	0.5866
1	0	0.9300	0.9000	6	2.6000	1.5350	-1.2186	-1.4487
1	0.5000	0.9300	0.9000	6	2.6000	1.5350	-0.2561	-0.5755
1	1.0000	0.9300	0.9000	6	2.6000	1.5350	0.7064	0.4601
1	0	0.8700	0.9000	6	4.0000	1.4500	-1.0514	-0.8161
1	0.5000	0.8700	0.9000	6	4.0000	1.4500	-0.0889	-0.2656
1	1.0000	0.8700	0.9000	6	4.0000	1.4500	0.8736	0.4339
1	0	0.8900	0.9000	6	4.0000	1.4500	-1.1306	-1.1823
1	0.5000	0.8900	0.9000	6	4.0000	1.4500	-0.1681	-0.4647
1	1.0000	0.8900	0.9000	6	4.0000	1.4500	0.7944	0.3398
1	0	0.9100	0.9000	6	4.0000	1.4500	-1.2550	-1.1850
1	0.5000	0.9100	0.9000	6	4.0000	1.4500	-0.2925	-0.3509
1	1.0000	0.9100	0.9000	6	4.0000	1.4500	0.6700	0.4270
1	0	0.9300	0.9000	6	4.0000	1.4500	-1.4790	-0.9065
1	0.5000	0.9300	0.9000	6	4.0000	1.4500	-0.5165	-0.1035
1	1.0000	0.9300	0.9000	6	4.0000	1.4500	0.4460	0.5296
1	0	0.8700	0.9000	6	4.0000	1.5350	-1.0514	-0.6245
1	0.5000	0.8700	0.9000	6	4.0000	1.5350	-0.0889	0.1200
1	1.0000	0.8700	0.9000	6	4.0000	1.5350	0.8736	1.0725
1	0	0.8900	0.9000	6	4.0000	1.5350	-1.1306	-1.0677
1	0.5000	0.8900	0.9000	6	4.0000	1.5350	-0.1681	-0.2022
1	1.0000	0.8900	0.9000	6	4.0000	1.5350	0.7944	0.8470
1	0	0.9100	0.9000	6	4.0000	1.5350	-1.2550	-1.3454
1	0.5000	0.9100	0.9000	6	4.0000	1.5350	-0.2925	-0.3853
1	1.0000	0.9100	0.9000	6	4.0000	1.5350	0.6700	0.6424
1	0	0.9300	0.9000	6	4.0000	1.5350	-1.4790	-1.3818
1	0.5000	0.9300	0.9000	6	4.0000	1.5350	-0.5165	-0.4309
1	1.0000	0.9300	0.9000	6	4.0000	1.5350	0.4460	0.4677
1	0	0.8700	0.9000	9	2.6000	1.4500	-1.0687	-0.4187
1	0.5000	0.8700	0.9000	9	2.6000	1.4500	-0.0537	0.2383
1	1.0000	0.8700	0.9000	9	2.6000	1.4500	0.9613	1.0420
1	0	0.8900	0.9000	9	2.6000	1.4500	-1.1202	-0.6935
1	0.5000	0.8900	0.9000	9	2.6000	1.4500	-0.1052	-0.0213
1	1.0000	0.8900	0.9000	9	2.6000	1.4500	0.9098	0.8292
1	0	0.9100	0.9000	9	2.6000	1.4500	-1.2011	-1.0443
1	0.5000	0.9100	0.9000	9	2.6000	1.4500	-0.1861	-0.2972
1	1.0000	0.9100	0.9000	9	2.6000	1.4500	0.8289	0.6370
1	0	0.9300	0.9000	9	2.6000	1.4500	-1.3467	-1.3713
1	0.5000	0.9300	0.9000	9	2.6000	1.4500	-0.3317	-0.4926
1	1.0000	0.9300	0.9000	9	2.6000	1.4500	0.6833	0.5079
1	0	0.8700	0.9000	9	2.6000	1.5350	-1.0687	-0.6750
1	0.5000	0.8700	0.9000	9	2.6000	1.5350	-0.0537	-0.0987
1	1.0000	0.8700	0.9000	9	2.6000	1.5350	0.9613	0.6063
1	0	0.8900	0.9000	9	2.6000	1.5350	-1.1202	-0.9550
1	0.5000	0.8900	0.9000	9	2.6000	1.5350	-0.1052	-0.3021
1	1.0000	0.8900	0.9000	9	2.6000	1.5350	0.9098	0.5523
1	0	0.9100	0.9000	9	2.6000	1.5350	-1.2011	-1.1885
1	0.5000	0.9100	0.9000	9	2.6000	1.5350	-0.1861	-0.4495
1	1.0000	0.9100	0.9000	9	2.6000	1.5350	0.8289	0.5310
1	0	0.9300	0.9000	9	2.6000	1.5350	-1.3467	-1.3704
1	0.5000	0.9300	0.9000	9	2.6000	1.5350	-0.3317	-0.5439
1	1.0000	0.9300	0.9000	9	2.6000	1.5350	0.6833	0.4779

Table E.6 contd.

1	0	0.8700	0.9000	9	4.0000	1.4500	-1.1795	-0.5522
1	0.5000	0.8700	0.9000	9	4.0000	1.4500	-0.1645	-0.0681
1	1.0000	0.8700	0.9000	9	4.0000	1.4500	0.8505	0.5464
1	0	0.8900	0.9000	9	4.0000	1.4500	-1.2587	-1.0347
1	0.5000	0.8900	0.9000	9	4.0000	1.4500	-0.2437	-0.3795
1	1.0000	0.8900	0.9000	9	4.0000	1.4500	0.7713	0.3852
1	0	0.9100	0.9000	9	4.0000	1.4500	-1.3831	-1.2112
1	0.5000	0.9100	0.9000	9	4.0000	1.4500	-0.3681	-0.3857
1	1.0000	0.9100	0.9000	9	4.0000	1.4500	0.6469	0.4320
1	0	0.9300	0.9000	9	4.0000	1.4500	-1.6071	-1.0689
1	0.5000	0.9300	0.9000	9	4.0000	1.4500	-0.5921	-0.1983
1	1.0000	0.9300	0.9000	9	4.0000	1.4500	0.4229	0.5293
1	0	0.8700	0.9000	9	4.0000	1.5350	-1.1795	-0.6714
1	0.5000	0.8700	0.9000	9	4.0000	1.5350	-0.1645	-0.0297
1	1.0000	0.8700	0.9000	9	4.0000	1.5350	0.8505	0.8466
1	0	0.8900	0.9000	9	4.0000	1.5350	-1.2587	-1.0458
1	0.5000	0.8900	0.9000	9	4.0000	1.5350	-0.2437	-0.3053
1	1.0000	0.8900	0.9000	9	4.0000	1.5350	0.7713	0.6638
1	0	0.9100	0.9000	9	4.0000	1.5350	-1.3831	-1.3398
1	0.5000	0.9100	0.9000	9	4.0000	1.5350	-0.3681	-0.4793
1	1.0000	0.9100	0.9000	9	4.0000	1.5350	0.6469	0.5085
1	0	0.9300	0.9000	9	4.0000	1.5350	-1.6071	-1.4457
1	0.5000	0.9300	0.9000	9	4.0000	1.5350	-0.5921	-0.5288
1	1.0000	0.9300	0.9000	9	4.0000	1.5350	0.4229	0.3776
1	0	0.8700	0.9000	12	2.6000	1.4500	-1.1968	-0.2523
1	0.5000	0.8700	0.9000	12	2.6000	1.4500	-0.1293	0.3742
1	1.0000	0.8700	0.9000	12	2.6000	1.4500	0.9382	1.1270
1	0	0.8900	0.9000	12	2.6000	1.4500	-1.2483	-0.4611
1	0.5000	0.8900	0.9000	12	2.6000	1.4500	-0.1808	0.1879
1	1.0000	0.8900	0.9000	12	2.6000	1.4500	0.8867	0.9895
1	0	0.9100	0.9000	12	2.6000	1.4500	-1.3292	-0.7764
1	0.5000	0.9100	0.9000	12	2.6000	1.4500	-0.2617	-0.0597
1	1.0000	0.9100	0.9000	12	2.6000	1.4500	0.8058	0.8330
1	0	0.9300	0.9000	12	2.6000	1.4500	-1.4748	-1.1353
1	0.5000	0.9300	0.9000	12	2.6000	1.4500	-0.4073	-0.2948
1	1.0000	0.9300	0.9000	12	2.6000	1.4500	0.6602	0.6934
1	0	0.8700	0.9000	12	2.6000	1.5350	-1.1968	-0.7138
1	0.5000	0.8700	0.9000	12	2.6000	1.5350	-0.1293	-0.1708
1	1.0000	0.8700	0.9000	12	2.6000	1.5350	0.9382	0.4964
1	0	0.8900	0.9000	12	2.6000	1.5350	-1.2483	-0.9347
1	0.5000	0.8900	0.9000	12	2.6000	1.5350	-0.1808	-0.3495
1	1.0000	0.8900	0.9000	12	2.6000	1.5350	0.8867	0.4204
1	0	0.9100	0.9000	12	2.6000	1.5350	-1.3292	-1.0979
1	0.5000	0.9100	0.9000	12	2.6000	1.5350	-0.2617	-0.4425
1	1.0000	0.9100	0.9000	12	2.6000	1.5350	0.8058	0.4370
1	0	0.9300	0.9000	12	2.6000	1.5350	-1.4748	-1.2132
1	0.5000	0.9300	0.9000	12	2.6000	1.5350	-0.4073	-0.4608
1	1.0000	0.9300	0.9000	12	2.6000	1.5350	0.6602	0.4909
1	0	0.8700	0.9000	12	4.0000	1.4500	-1.3076	-0.3133
1	0.5000	0.8700	0.9000	12	4.0000	1.4500	-0.2401	0.1303
1	1.0000	0.8700	0.9000	12	4.0000	1.4500	0.8274	0.6819
1	0	0.8900	0.9000	12	4.0000	1.4500	-1.3868	-0.8611
1	0.5000	0.8900	0.9000	12	4.0000	1.4500	-0.3193	-0.2649
1	1.0000	0.8900	0.9000	12	4.0000	1.4500	0.7482	0.4537
1	0	0.9100	0.9000	12	4.0000	1.4500	-1.5112	-1.1848
1	0.5000	0.9100	0.9000	12	4.0000	1.4500	-0.4437	-0.3920
1	1.0000	0.9100	0.9000	12	4.0000	1.4500	0.6238	0.4434
1	0	0.9300	0.9000	12	4.0000	1.4500	-1.7352	-1.1780
1	0.5000	0.9300	0.9000	12	4.0000	1.4500	-0.6677	-0.2716
1	1.0000	0.9300	0.9000	12	4.0000	1.4500	0.3998	0.5365
1	0	0.8700	0.9000	12	4.0000	1.5350	-1.3076	-0.7653
1	0.5000	0.8700	0.9000	12	4.0000	1.5350	-0.2401	-0.2157
1	1.0000	0.8700	0.9000	12	4.0000	1.5350	0.8274	0.5717
1	0	0.8900	0.9000	12	4.0000	1.5350	-1.3868	-1.0549
1	0.5000	0.8900	0.9000	12	4.0000	1.5350	-0.3193	-0.4208
1	1.0000	0.8900	0.9000	12	4.0000	1.5350	0.7482	0.4607
1	0	0.9100	0.9000	12	4.0000	1.5350	-1.5112	-1.3363
1	0.5000	0.9100	0.9000	12	4.0000	1.5350	-0.4437	-0.5732
1	1.0000	0.9100	0.9000	12	4.0000	1.5350	0.6238	0.3670
1	0	0.9300	0.9000	12	4.0000	1.5350	-1.7352	-1.4920
1	0.5000	0.9300	0.9000	12	4.0000	1.5350	-0.6677	-0.6264
1	1.0000	0.9300	0.9000	12	4.0000	1.5350	0.3998	0.2822
1	0	0.8700	0.9250	3	2.6000	1.4500	-0.6550	-0.5071
1	0.5000	0.8700	0.9250	3	2.6000	1.4500	0.2550	0.3369
1	0	0.8900	0.9250	3	2.6000	1.4500	-0.7065	-0.7641

Table E.6 contd.

1	0.5000	0.8900	0.9250	3	2.6000	1.4500	0.2035	0.1797
1	0	0.9100	0.9250	3	2.6000	1.4500	-0.7874	-0.9963
1	0.5000	0.9100	0.9250	3	2.6000	1.4500	0.1226	0.0251
1	0	0.9300	0.9250	3	2.6000	1.4500	-0.9330	-1.1261
1	0.5000	0.9300	0.9250	3	2.6000	1.4500	-0.0230	-0.1010
1	0	0.8700	0.9250	3	2.6000	1.5350	-0.6550	-0.5114
1	0.5000	0.8700	0.9250	3	2.6000	1.5350	0.2550	0.0726
1	0	0.8900	0.9250	3	2.6000	1.5350	-0.7065	-0.6371
1	0.5000	0.8900	0.9250	3	2.6000	1.5350	0.2035	0.1376
1	0	0.9100	0.9250	3	2.6000	1.5350	-0.7874	-0.7570
1	0.5000	0.9100	0.9250	3	2.6000	1.5350	0.1226	0.1765
1	0	0.9300	0.9250	3	2.6000	1.5350	-0.9330	-0.8942
1	0.5000	0.9300	0.9250	3	2.6000	1.5350	-0.0230	0.0668
1	0	0.8700	0.9250	3	4.0000	1.4500	-0.7658	-0.7316
1	0.5000	0.8700	0.9250	3	4.0000	1.4500	0.1442	0.1223
1	0	0.8900	0.9250	3	4.0000	1.4500	-0.8450	-0.9377
1	0.5000	0.8900	0.9250	3	4.0000	1.4500	0.0650	-0.0122
1	0	0.9100	0.9250	3	4.0000	1.4500	-0.9694	-0.8517
1	0.5000	0.9100	0.9250	3	4.0000	1.4500	-0.0594	0.0194
1	0	0.9300	0.9250	3	4.0000	1.4500	-1.1934	-0.6333
1	0.5000	0.9300	0.9250	3	4.0000	1.4500	-0.2834	0.0785
1	0	0.8700	0.9250	3	4.0000	1.5350	-0.7658	-0.4294
1	0.5000	0.8700	0.9250	3	4.0000	1.5350	0.1442	0.4646
1	0	0.8900	0.9250	3	4.0000	1.5350	-0.8450	-0.7494
1	0.5000	0.8900	0.9250	3	4.0000	1.5350	0.0650	0.2524
1	0	0.9100	0.9250	3	4.0000	1.5350	-0.9694	-1.0085
1	0.5000	0.9100	0.9250	3	4.0000	1.5350	-0.0594	-0.0292
1	0	0.9300	0.9250	3	4.0000	1.5350	-1.1934	-1.1798
1	0.5000	0.9300	0.9250	3	4.0000	1.5350	-0.2834	-0.3295
1	0	0.8700	0.9250	6	2.6000	1.4500	-0.7831	-0.6760
1	0.5000	0.8700	0.9250	6	2.6000	1.4500	0.1794	0.1490
1	1.0000	0.8700	0.9250	6	2.6000	1.4500	1.1419	1.1791
1	0	0.8900	0.9250	6	2.6000	1.4500	-0.8346	-0.8989
1	0.5000	0.8900	0.9250	6	2.6000	1.4500	0.1279	-0.0060
1	1.0000	0.8900	0.9250	6	2.6000	1.4500	1.0904	1.1113
1	0	0.9100	0.9250	6	2.6000	1.4500	-0.9155	-1.1072
1	0.5000	0.9100	0.9250	6	2.6000	1.4500	0.0470	-0.1373
1	1.0000	0.9100	0.9250	6	2.6000	1.4500	1.0095	0.9890
1	0	0.9300	0.9250	6	2.6000	1.4500	-1.0611	-1.2503
1	0.5000	0.9300	0.9250	6	2.6000	1.4500	-0.0986	-0.2413
1	1.0000	0.9300	0.9250	6	2.6000	1.4500	0.8639	0.7879
1	0	0.8700	0.9250	6	2.6000	1.5350	-0.7831	-0.5354
1	0.5000	0.8700	0.9250	6	2.6000	1.5350	0.1794	0.0996
1	1.0000	0.8700	0.9250	6	2.6000	1.5350	1.1419	0.8719
1	0	0.8900	0.9250	6	2.6000	1.5350	-0.8346	-0.7792
1	0.5000	0.8900	0.9250	6	2.6000	1.5350	0.1279	0.0291
1	1.0000	0.8900	0.9250	6	2.6000	1.5350	1.0904	1.0045
1	0	0.9100	0.9250	6	2.6000	1.5350	-0.9155	-0.9712
1	0.5000	0.9100	0.9250	6	2.6000	1.5350	0.0470	-0.0142
1	1.0000	0.9100	0.9250	6	2.6000	1.5350	1.0095	1.0726
1	0	0.9300	0.9250	6	2.6000	1.5350	-1.0611	-1.0984
1	0.5000	0.9300	0.9250	6	2.6000	1.5350	-0.0986	-0.1038
1	1.0000	0.9300	0.9250	6	2.6000	1.5350	0.8639	0.9176
1	0	0.8700	0.9250	6	4.0000	1.4500	-0.8939	-0.7546
1	0.5000	0.8700	0.9250	6	4.0000	1.4500	0.0686	0.0381
1	1.0000	0.8700	0.9250	6	4.0000	1.4500	1.0311	0.9508
1	0	0.8900	0.9250	6	4.0000	1.4500	-0.9731	-1.0738
1	0.5000	0.8900	0.9250	6	4.0000	1.4500	-0.0106	-0.1750
1	1.0000	0.8900	0.9250	6	4.0000	1.4500	0.9519	0.7322
1	0	0.9100	0.9250	6	4.0000	1.4500	-1.0975	-1.1071
1	0.5000	0.9100	0.9250	6	4.0000	1.4500	-0.1350	-0.1942
1	1.0000	0.9100	0.9250	6	4.0000	1.4500	0.8275	0.5935
1	0	0.9300	0.9250	6	4.0000	1.4500	-1.3215	-0.9440
1	0.5000	0.9300	0.9250	6	4.0000	1.4500	-0.3590	-0.1448
1	1.0000	0.9300	0.9250	6	4.0000	1.4500	0.6035	0.4721
1	0	0.8700	0.9250	6	4.0000	1.5350	-0.8939	-0.5775
1	0.5000	0.8700	0.9250	6	4.0000	1.5350	0.0686	0.2905
1	1.0000	0.8700	0.9250	6	4.0000	1.5350	1.0311	1.3330
1	0	0.8900	0.9250	6	4.0000	1.5350	-0.9731	-0.9199
1	0.5000	0.8900	0.9250	6	4.0000	1.5350	-0.0106	0.0499
1	1.0000	0.8900	0.9250	6	4.0000	1.5350	0.9519	1.1222
1	0	0.9100	0.9250	6	4.0000	1.5350	-1.0975	-1.1998
1	0.5000	0.9100	0.9250	6	4.0000	1.5350	-0.1350	-0.2377
1	1.0000	0.9100	0.9250	6	4.0000	1.5350	0.8275	0.7327
1	0	0.9300	0.9250	6	4.0000	1.5350	-1.3215	-1.3848

Table E.6 contd.

1	0.5000	0.9300	0.9250	6	4.0000	1.5350	-0.3590	-0.5359
1	1.0000	0.9300	0.9250	6	4.0000	1.5350	0.6035	0.2800
1	0	0.8700	0.9250	9	2.6000	1.4500	-0.9112	-0.7219
1	0.5000	0.8700	0.9250	9	2.6000	1.4500	0.1038	0.0496
1	1.0000	0.8700	0.9250	9	2.6000	1.4500	1.1188	1.0248
1	0	0.8900	0.9250	9	2.6000	1.4500	-0.9627	-0.9020
1	0.5000	0.8900	0.9250	9	2.6000	1.4500	0.0523	-0.0791
1	1.0000	0.8900	0.9250	9	2.6000	1.4500	1.0673	0.9728
1	0	0.9100	0.9250	9	2.6000	1.4500	-1.0436	-1.0733
1	0.5000	0.9100	0.9250	9	2.6000	1.4500	-0.0286	-0.1715
1	1.0000	0.9100	0.9250	9	2.6000	1.4500	0.9864	0.9166
1	0	0.9300	0.9250	9	2.6000	1.4500	-1.1892	-1.2090
1	0.5000	0.9300	0.9250	9	2.6000	1.4500	-0.1742	-0.2406
1	1.0000	0.9300	0.9250	9	2.6000	1.4500	0.8408	0.7914
1	0	0.8700	0.9250	9	2.6000	1.5350	-0.9112	-0.5718
1	0.5000	0.8700	0.9250	9	2.6000	1.5350	0.1038	0.0807
1	1.0000	0.8700	0.9250	9	2.6000	1.5350	1.1188	0.8455
1	0	0.8900	0.9250	9	2.6000	1.5350	-0.9627	-0.8735
1	0.5000	0.8900	0.9250	9	2.6000	1.5350	0.0523	-0.0894
1	1.0000	0.8900	0.9250	9	2.6000	1.5350	1.0673	0.8684
1	0	0.9100	0.9250	9	2.6000	1.5350	-1.0436	-1.0728
1	0.5000	0.9100	0.9250	9	2.6000	1.5350	-0.0286	-0.1685
1	1.0000	0.9100	0.9250	9	2.6000	1.5350	0.9864	0.9115
1	0	0.9300	0.9250	9	2.6000	1.5350	-1.1892	-1.1458
1	0.5000	0.9300	0.9250	9	2.6000	1.5350	-0.1742	-0.1875
1	1.0000	0.9300	0.9250	9	2.6000	1.5350	0.8408	0.8502
1	0	0.8700	0.9250	9	4.0000	1.4500	-1.0220	-0.6734
1	0.5000	0.8700	0.9250	9	4.0000	1.4500	-0.0070	0.0659
1	1.0000	0.8700	0.9250	9	4.0000	1.4500	1.0080	0.9439
1	0	0.8900	0.9250	9	4.0000	1.4500	-1.1012	-1.0723
1	0.5000	0.8900	0.9250	9	4.0000	1.4500	-0.0862	-0.2063
1	1.0000	0.8900	0.9250	9	4.0000	1.4500	0.9288	0.7173
1	0	0.9100	0.9250	9	4.0000	1.4500	-1.2256	-1.2266
1	0.5000	0.9100	0.9250	9	4.0000	1.4500	-0.2106	-0.2857
1	1.0000	0.9100	0.9250	9	4.0000	1.4500	0.8044	0.5747
1	0	0.9300	0.9250	9	4.0000	1.4500	-1.4496	-1.1347
1	0.5000	0.9300	0.9250	9	4.0000	1.4500	-0.4346	-0.2509
1	1.0000	0.9300	0.9250	9	4.0000	1.4500	0.5804	0.4615
1	0	0.8700	0.9250	9	4.0000	1.5350	-1.0220	-0.7688
1	0.5000	0.8700	0.9250	9	4.0000	1.5350	-0.0070	0.0307
1	1.0000	0.8700	0.9250	9	4.0000	1.5350	1.0080	1.0569
1	0	0.8900	0.9250	9	4.0000	1.5350	-1.1012	-1.0507
1	0.5000	0.8900	0.9250	9	4.0000	1.5350	-0.0862	-0.1586
1	1.0000	0.8900	0.9250	9	4.0000	1.5350	0.9288	0.9078
1	0	0.9100	0.9250	9	4.0000	1.5350	-1.2256	-1.2788
1	0.5000	0.9100	0.9250	9	4.0000	1.5350	-0.2106	-0.3648
1	1.0000	0.9100	0.9250	9	4.0000	1.5350	0.8044	0.6078
1	0	0.9300	0.9250	9	4.0000	1.5350	-1.4496	-1.4303
1	0.5000	0.9300	0.9250	9	4.0000	1.5350	-0.4346	-0.5914
1	1.0000	0.9300	0.9250	9	4.0000	1.5350	0.5804	0.2238
1	0	0.8700	0.9250	12	2.6000	1.4500	-1.0393	-0.6586
1	0.5000	0.8700	0.9250	12	2.6000	1.4500	0.0282	0.0174
1	1.0000	0.8700	0.9250	12	2.6000	1.4500	1.0957	0.8734
1	0	0.8900	0.9250	12	2.6000	1.4500	-1.0908	-0.7773
1	0.5000	0.8900	0.9250	12	2.6000	1.4500	-0.0233	-0.0531
1	1.0000	0.8900	0.9250	12	2.6000	1.4500	1.0442	0.8804
1	0	0.9100	0.9250	12	2.6000	1.4500	-1.1717	-0.8929
1	0.5000	0.9100	0.9250	12	2.6000	1.4500	-0.1042	-0.0858
1	1.0000	0.9100	0.9250	12	2.6000	1.4500	0.9633	0.9168
1	0	0.9300	0.9250	12	2.6000	1.4500	-1.3173	-1.0001
1	0.5000	0.9300	0.9250	12	2.6000	1.4500	-0.2498	-0.1040
1	1.0000	0.9300	0.9250	12	2.6000	1.4500	0.8177	0.8944
1	0	0.8700	0.9250	12	2.6000	1.5350	-1.0393	-0.6544
1	0.5000	0.8700	0.9250	12	2.6000	1.5350	0.0282	-0.0538
1	1.0000	0.8700	0.9250	12	2.6000	1.5350	1.0957	0.6509
1	0	0.8900	0.9250	12	2.6000	1.5350	-1.0908	-0.9223
1	0.5000	0.8900	0.9250	12	2.6000	1.5350	-0.0233	-0.2422
1	1.0000	0.8900	0.9250	12	2.6000	1.5350	1.0442	0.6131
1	0	0.9100	0.9250	12	2.6000	1.5350	-1.1717	-1.0562
1	0.5000	0.9100	0.9250	12	2.6000	1.5350	-0.1042	-0.2804
1	1.0000	0.9100	0.9250	12	2.6000	1.5350	0.9633	0.6909
1	0	0.9300	0.9250	12	2.6000	1.5350	-1.3173	-1.0434
1	0.5000	0.9300	0.9250	12	2.6000	1.5350	-0.2498	-0.1857
1	1.0000	0.9300	0.9250	12	2.6000	1.5350	0.8177	0.7867
1	0	0.8700	0.9250	12	4.0000	1.4500	-1.1501	-0.5461

Table E.6 contd.

1	0.5000	0.8700	0.9250	12	4.0000	1.4500	-0.0826	0.1457
1	1.0000	0.8700	0.9250	12	4.0000	1.4500	0.9849	0.9817
1	0	0.8900	0.9250	12	4.0000	1.4500	-1.2293	-0.9491
1	0.5000	0.8900	0.9250	12	4.0000	1.4500	-0.1618	-0.1291
1	1.0000	0.8900	0.9250	12	4.0000	1.4500	0.9057	0.7886
1	0	0.9100	0.9250	12	4.0000	1.4500	-1.3537	-1.1768
1	0.5000	0.9100	0.9250	12	4.0000	1.4500	-0.2862	-0.2367
1	1.0000	0.9100	0.9250	12	4.0000	1.4500	0.7813	0.6745
1	0	0.9300	0.9250	12	4.0000	1.4500	-1.5777	-1.1468
1	0.5000	0.9300	0.9250	12	4.0000	1.4500	-0.5102	-0.1983
1	1.0000	0.9300	0.9250	12	4.0000	1.4500	0.5573	0.6012
1	0	0.8700	0.9250	12	4.0000	1.5350	-1.1501	-0.9760
1	0.5000	0.8700	0.9250	12	4.0000	1.5350	-0.0826	-0.2689
1	1.0000	0.8700	0.9250	12	4.0000	1.5350	0.9849	0.6932
1	0	0.8900	0.9250	12	4.0000	1.5350	-1.2293	-1.1434
1	0.5000	0.8900	0.9250	12	4.0000	1.5350	-0.1618	-0.3480
1	1.0000	0.8900	0.9250	12	4.0000	1.5350	0.9057	0.6671
1	0	0.9100	0.9250	12	4.0000	1.5350	-1.3537	-1.2651
1	0.5000	0.9100	0.9250	12	4.0000	1.5350	-0.2862	-0.4136
1	1.0000	0.9100	0.9250	12	4.0000	1.5350	0.7813	0.5390
1	0	0.9300	0.9250	12	4.0000	1.5350	-1.5777	-1.3369
1	0.5000	0.9300	0.9250	12	4.0000	1.5350	-0.5102	-0.5138
1	1.0000	0.9300	0.9250	12	4.0000	1.5350	0.5573	0.2965
1	0	0.8700	0.9500	3	2.6000	1.4500	-0.4975	-0.5050
1	0.5000	0.8700	0.9500	3	2.6000	1.4500	0.4125	0.4548
1	0	0.8900	0.9500	3	2.6000	1.4500	-0.5490	-0.6203
1	0.5000	0.8900	0.9500	3	2.6000	1.4500	0.3610	0.4747
1	0	0.9100	0.9500	3	2.6000	1.4500	-0.6299	-0.7131
1	0.5000	0.9100	0.9500	3	2.6000	1.4500	0.2801	0.4045
1	0	0.9300	0.9500	3	2.6000	1.4500	-0.7755	-0.8045
1	0.5000	0.9300	0.9500	3	2.6000	1.4500	0.1345	0.2008
1	0	0.8700	0.9500	3	2.6000	1.5350	-0.4975	-0.3555
1	0.5000	0.8700	0.9500	3	2.6000	1.5350	0.4125	0.3281
1	0	0.8900	0.9500	3	2.6000	1.5350	-0.5490	-0.3345
1	0.5000	0.8900	0.9500	3	2.6000	1.5350	0.3610	0.5425
1	0	0.9100	0.9500	3	2.6000	1.5350	-0.6299	-0.2822
1	0.5000	0.9100	0.9500	3	2.6000	1.5350	0.2801	0.7039
1	0	0.9300	0.9500	3	2.6000	1.5350	-0.7755	-0.3055
1	0.5000	0.9300	0.9500	3	2.6000	1.5350	0.1345	0.6214
1	0	0.8700	0.9500	3	4.0000	1.4500	-0.6083	-0.5880
1	0.5000	0.8700	0.9500	3	4.0000	1:4500	0.3017	0.4592
1	0	0.8900	0.9500	3	4.0000	1.4500	-0.6875	-0.8034
1	0.5000	0.8900	0.9500	3	4.0000	1.4500	0.2225	0.2158
1	0	0.9100	0.9500	3	4.0000	1.4500	-0.8119	-0.8527
1	0.5000	0.9100	0.9500	3	4.0000	1.4500	0.0981	0.0272
1	0	0.9300	0.9500	3	4.0000	1.4500	-1.0359	-0.8178
1	0.5000	0.9300	0.9500	3	4.0000	1.4500	-0.1259	-0.1175
1	0	0.8700	0.9500	3	4.0000	1.5350	-0.6083	-0.3080
1	0.5000	0.8700	0.9500	3	4.0000	1.5350	0.3017	0.6273
1	0	0.8900	0.9500	3	4.0000	1.5350	-0.6875	-0.5001
1	0.5000	0.8900	0.9500	3	4.0000	1.5350	0.2225	0.4592
1	0	0.9100	0.9500	3	4.0000	1.5350	-0.8119	-0.8014
1	0.5000	0.9100	0.9500	3	4.0000	1.5350	0.0981	0.0531
1	0	0.9300	0.9500	3	4.0000	1.5350	-1.0359	-1.1685
1	0.5000	0.9300	0.9500	3	4.0000	1.5350	-0.1259	-0.4646
1	0	0.8700	0.9500	6	2.6000	1.4500	-0.6256	-0.7833
1	0.5000	0.8700	0.9500	6	2.6000	1.4500	0.3369	0.1323
1	1.0000	0.8700	0.9500	6	2.6000	1.4500	1.2994	1.2549
1	0	0.8900	0.9500	6	2.6000	1.4500	-0.6771	-0.8777
1	0.5000	0.8900	0.9500	6	2.6000	1.4500	0.2854	0.1421
1	1.0000	0.8900	0.9500	6	2.6000	1.4500	1.2479	1.3484
1	0	0.9100	0.9500	6	2.6000	1.4500	-0.7580	-0.8907
1	0.5000	0.9100	0.9500	6	2.6000	1.4500	0.2045	0.1639
1	1.0000	0.9100	0.9500	6	2.6000	1.4500	1.1670	1.2933
1	0	0.9300	0.9500	6	2.6000	1.4500	-0.9036	-0.8741
1	0.5000	0.9300	0.9500	6	2.6000	1.4500	0.0589	0.0997
1	1.0000	0.9300	0.9500	6	2.6000	1.4500	1.0214	1.0303
1	0	0.8700	0.9500	6	2.6000	1.5350	-0.6256	-0.4094
1	0.5000	0.8700	0.9500	6	2.6000	1.5350	0.3369	0.2846
1	1.0000	0.8700	0.9500	6	2.6000	1.5350	1.2994	1.1099
1	0	0.8900	0.9500	6	2.6000	1.5350	-0.6771	-0.5151
1	0.5000	0.8900	0.9500	6	2.6000	1.5350	0.2854	0.3738
1	1.0000	0.8900	0.9500	6	2.6000	1.5350	1.2479	1.3639
1	0	0.9100	0.9500	6	2.6000	1.5350	-0.7580	-0.5332
1	0.5000	0.9100	0.9500	6	2.6000	1.5350	0.2045	0.4905

Table E.6 contd.

1	1.0000	0.9100	0.9500	6	2.6000	1.5350	1.1670	1.5075
1	0	0.9300	0.9500	6	2.6000	1.5350	-0.9036	-0.5109
1	0.5000	0.9300	0.9500	6	2.6000	1.5350	0.0589	0.4796
1	1.0000	0.9300	0.9500	6	2.6000	1.5350	1.0214	1.3556
1	0	0.8700	0.9500	6	4.0000	1.4500	-0.7364	-0.8161
1	0.5000	0.8700	0.9500	6	4.0000	1.4500	0.2261	0.2009
1	1.0000	0.8700	0.9500	6	4.0000	1.4500	1.1886	1.2935
1	0	0.8900	0.9500	6	4.0000	1.4500	-0.8156	-1.0445
1	0.5000	0.8900	0.9500	6	4.0000	1.4500	0.1469	-0.0255
1	1.0000	0.8900	0.9500	6	4.0000	1.4500	1.1094	0.9453
1	0	0.9100	0.9500	6	4.0000	1.4500	-0.9400	-1.1000
1	0.5000	0.9100	0.9500	6	4.0000	1.4500	0.0225	-0.1793
1	1.0000	0.9100	0.9500	6	4.0000	1.4500	0.9850	0.6092
1	0	0.9300	0.9500	6	4.0000	1.4500	-1.1640	-1.0395
1	0.5000	0.9300	0.9500	6	4.0000	1.4500	-0.2015	-0.2785
1	1.0000	0.9300	0.9500	6	4.0000	1.4500	0.7610	0.3412
1	0	0.8700	0.9500	6	4.0000	1.5350	-0.7364	-0.5605
1	0.5000	0.8700	0.9500	6	4.0000	1.5350	0.2261	0.4111
1	1.0000	0.8700	0.9500	6	4.0000	1.5350	1.1886	1.4751
1	0	0.8900	0.9500	6	4.0000	1.5350	-0.8156	-0.7571
1	0.5000	0.8900	0.9500	6	4.0000	1.5350	0.1469	0.2580
1	1.0000	0.8900	0.9500	6	4.0000	1.5350	1.1094	1.2626
1	0	0.9100	0.9500	6	4.0000	1.5350	-0.9400	-0.9967
1	0.5000	0.9100	0.9500	6	4.0000	1.5350	0.0225	-0.0968
1	1.0000	0.9100	0.9500	6	4.0000	1.5350	0.9850	0.7511
1	0	0.9300	0.9500	6	4.0000	1.5350	-1.1640	-1.2873
1	0.5000	0.9300	0.9500	6	4.0000	1.5350	-0.2015	-0.5738
1	1.0000	0.9300	0.9500	6	4.0000	1.5350	0.7610	0.1222
1	0	0.8700	0.9500	9	2.6000	1.4500	-0.7537	-0.9307
1	0.5000	0.8700	0.9500	9	2.6000	1.4500	0.2613	-0.1225
1	1.0000	0.8700	0.9500	9	2.6000	1.4500	1.2763	0.9085
1	0	0.8900	0.9500	9	2.6000	1.4500	-0.8052	-0.9755
1	0.5000	0.8900	0.9500	9	2.6000	1.4500	0.2098	-0.0849
1	1.0000	0.8900	0.9500	9	2.6000	1.4500	1.2248	1.0308
1	0	0.9100	0.9500	9	2.6000	1.4500	-0.8861	-0.9134
1	0.5000	0.9100	0.9500	9	2.6000	1.4500	0.1289	0.0379
1	1.0000	0.9100	0.9500	9	2.6000	1.4500	1.1439	1.1155
1	0	0.9300	0.9500	9	2.6000	1.4500	-1.0317	-0.8090
1	0.5000	0.9300	0.9500	9	2.6000	1.4500	-0.0167	0.1136
1	1.0000	0.9300	0.9500	9	2.6000	1.4500	0.9983	1.0205
1	0	0.8700	0.9500	9	2.6000	1.5350	-0.7537	-0.4388
1	0.5000	0.8700	0.9500	9	2.6000	1.5350	0.2613	0.2298
1	1.0000	0.8700	0.9500	9	2.6000	1.5350	1.2763	1.0071
1	0	0.8900	0.9500	9	2.6000	1.5350	-0.8052	-0.6279
1	0.5000	0.8900	0.9500	9	2.6000	1.5350	0.2098	0.2142
1	1.0000	0.8900	0.9500	9	2.6000	1.5350	1.2248	1.1831
1	0	0.9100	0.9500	9	2.6000	1.5350	-0.8861	-0.6823
1	0.5000	0.9100	0.9500	9	2.6000	1.5350	0.1289	0.2949
1	1.0000	0.9100	0.9500	9	2.6000	1.5350	1.1439	1.3372
1	0	0.9300	0.9500	9	2.6000	1.5350	-1.0317	-0.6044
1	0.5000	0.9300	0.9500	9	2.6000	1.5350	-0.0167	0.3713
1	1.0000	0.9300	0.9500	9	2.6000	1.5350	0.9983	1.2944
1	0	0.8700	0.9500	9	4.0000	1.4500	-0.8645	-0.8399
1	0.5000	0.8700	0.9500	9	4.0000	1.4500	0.1505	0.1256
1	1.0000	0.8700	0.9500	9	4.0000	1.4500	1.1655	1.2249
1	0	0.8900	0.9500	9	4.0000	1.4500	-0.9437	-1.0431
1	0.5000	0.8900	0.9500	9	4.0000	1.4500	0.0713	-0.0358
1	1.0000	0.8900	0.9500	9	4.0000	1.4500	1.0863	0.9765
1	0	0.9100	0.9500	9	4.0000	1.4500	-1.0681	-1.1073
1	0.5000	0.9100	0.9500	9	4.0000	1.4500	-0.0531	-0.1439
1	1.0000	0.9100	0.9500	9	4.0000	1.4500	0.9619	0.7083
1	0	0.9300	0.9500	9	4.0000	1.4500	-1.2921	-1.0487
1	0.5000	0.9300	0.9500	9	4.0000	1.4500	-0.2771	-0.2119
1	1.0000	0.9300	0.9500	9	4.0000	1.4500	0.7379	0.4765
1	0	0.8700	0.9500	9	4.0000	1.5350	-0.8645	-0.7746
1	0.5000	0.8700	0.9500	9	4.0000	1.5350	0.1505	0.1719
1	1.0000	0.8700	0.9500	9	4.0000	1.5350	1.1655	1.2775
1	0	0.8900	0.9500	9	4.0000	1.5350	-0.9437	-0.8897
1	0.5000	0.8900	0.9500	9	4.0000	1.5350	0.0713	0.1147
1	1.0000	0.8900	0.9500	9	4.0000	1.5350	1.0863	1.1808
1	0	0.9100	0.9500	9	4.0000	1.5350	-1.0681	-0.9923
1	0.5000	0.9100	0.9500	9	4.0000	1.5350	-0.0531	-0.0793
1	1.0000	0.9100	0.9500	9	4.0000	1.5350	0.9619	0.8061
1	0	0.9300	0.9500	9	4.0000	1.5350	-1.2921	-1.1538
1	0.5000	0.9300	0.9500	9	4.0000	1.5350	-0.2771	-0.4241

Table E.6 contd.

1	1.0000	0.9300	0.9500	9	4.0000	1.5350	0.7379	0.2668
1	0	0.8700	0.9500	12	2.6000	1.4500	-0.8818	-0.9254
1	0.5000	0.8700	0.9500	12	2.6000	1.4500	0.1857	-0.2601
1	1.0000	0.8700	0.9500	12	2.6000	1.4500	1.2532	0.6212
1	0	0.8900	0.9500	12	2.6000	1.4500	-0.9333	-0.9202
1	0.5000	0.8900	0.9500	12	2.6000	1.4500	0.1342	-0.1753
1	1.0000	0.8900	0.9500	12	2.6000	1.4500	1.2017	0.8028
1	0	0.9100	0.9500	12	2.6000	1.4500	-1.0142	-0.8131
1	0.5000	0.9100	0.9500	12	2.6000	1.4500	0.0533	0.0231
1	1.0000	0.9100	0.9500	12	2.6000	1.4500	1.1208	1.0239
1	0	0.9300	0.9500	12	2.6000	1.4500	-1.1598	-0.6588
1	0.5000	0.9300	0.9500	12	2.6000	1.4500	-0.0923	0.2064
1	1.0000	0.9300	0.9500	12	2.6000	1.4500	0.9752	1.0963
1	0	0.8700	0.9500	12	2.6000	1.5350	-0.8818	-0.4883
1	0.5000	0.8700	0.9500	12	2.6000	1.5350	0.1857	0.1112
1	1.0000	0.8700	0.9500	12	2.6000	1.5350	1.2532	0.8140
1	0	0.8900	0.9500	12	2.6000	1.5350	-0.9333	-0.6912
1	0.5000	0.8900	0.9500	12	2.6000	1.5350	0.1342	0.0424
1	1.0000	0.8900	0.9500	12	2.6000	1.5350	1.2017	0.9323
1	0	0.9100	0.9500	12	2.6000	1.5350	-1.0142	-0.7322
1	0.5000	0.9100	0.9500	12	2.6000	1.5350	0.0533	0.1260
1	1.0000	0.9100	0.9500	12	2.6000	1.5350	1.1208	1.1163
1	0	0.9300	0.9500	12	2.6000	1.5350	-1.1598	-0.5959
1	0.5000	0.9300	0.9500	12	2.6000	1.5350	-0.0923	0.3061
1	1.0000	0.9300	0.9500	12	2.6000	1.5350	0.9752	1.2195
1	0	0.8700	0.9500	12	4.0000	1.4500	-0.9926	-0.6719
1	0.5000	0.8700	0.9500	12	4.0000	1.4500	0.0749	0.2241
1	1.0000	0.8700	0.9500	12	4.0000	1.4500	1.1424	1.2811
1	0	0.8900	0.9500	12	4.0000	1.4500	-1.0718	-0.8147
1	0.5000	0.8900	0.9500	12	4.0000	1.4500	-0.0043	0.1612
1	1.0000	0.8900	0.9500	12	4.0000	1.4500	1.0632	1.1833
1	0	0.9100	0.9500	12	4.0000	1.4500	-1.1962	-0.8798
1	0.5000	0.9100	0.9500	12	4.0000	1.4500	-0.1287	0.1092
1	1.0000	0.9100	0.9500	12	4.0000	1.4500	0.9388	1.0097
1	0	0.9300	0.9500	12	4.0000	1.4500	-1.4202	-0.8424
1	0.5000	0.9300	0.9500	12	4.0000	1.4500	-0.3527	0.0629
1	1.0000	0.9300	0.9500	12	4.0000	1.4500	0.7148	0.8126
1	0	0.8700	0.9500	12	4.0000	1.5350	-0.9926	-0.9082
1	0.5000	0.8700	0.9500	12	4.0000	1.5350	0.0749	-0.0461
1	1.0000	0.8700	0.9500	12	4.0000	1.5350	1.1424	1.0234
1	0	0.8900	0.9500	12	4.0000	1.5350	-1.0718	-0.8968
1	0.5000	0.8900	0.9500	12	4.0000	1.5350	-0.0043	0.0407
1	1.0000	0.8900	0.9500	12	4.0000	1.5350	1.0632	1.0955
1	0	0.9100	0.9500	12	4.0000	1.5350	-1.1962	-0.8369
1	0.5000	0.9100	0.9500	12	4.0000	1.5350	-0.1287	0.0608
1	1.0000	0.9100	0.9500	12	4.0000	1.5350	0.9388	0.9460
1	0	0.9300	0.9500	12	4.0000	1.5350	-1.4202	-0.8415
1	0.5000	0.9300	0.9500	12	4.0000	1.5350	-0.3527	-0.0952
1	1.0000	0.9300	0.9500	12	4.0000	1.5350	0.7148	0.5786
2	0.5000	0.8700	0.9000	3	2.6000	1.4500	-0.7151	-0.5169
2	1.0000	0.8700	0.9000	3	2.6000	1.4500	0.1949	0.1925
2	1.5000	0.8700	0.9000	3	2.6000	1.4500	1.1049	0.9888
2	0.5000	0.8900	0.9000	3	2.6000	1.4500	-0.8180	-1.0107
2	1.0000	0.8900	0.9000	3	2.6000	1.4500	0.0920	-0.1662
2	1.5000	0.8900	0.9000	3	2.6000	1.4500	1.0020	0.8028
2	0.5000	0.9100	0.9000	3	2.6000	1.4500	-0.9798	-1.2895
2	1.0000	0.9100	0.9000	3	2.6000	1.4500	-0.0698	-0.2981
2	1.5000	0.9100	0.9000	3	2.6000	1.4500	0.8402	0.7583
2	0.5000	0.9300	0.9000	3	2.6000	1.4500	-1.2710	-1.2254
2	1.0000	0.9300	0.9000	3	2.6000	1.4500	-0.3610	-0.1770
2	1.5000	0.9300	0.9000	3	2.6000	1.4500	0.5490	0.8043
2	0.5000	0.8700	0.9000	3	2.6000	1.5350	-0.7151	-0.6622
2	1.0000	0.8700	0.9000	3	2.6000	1.5350	0.1949	-0.1161
2	1.5000	0.8700	0.9000	3	2.6000	1.5350	1.1049	0.6189
2	0.5000	0.8900	0.9000	3	2.6000	1.5350	-0.8180	-0.9765
2	1.0000	0.8900	0.9000	3	2.6000	1.5350	0.0920	-0.2622
2	1.5000	0.8900	0.9000	3	2.6000	1.5350	1.0020	0.6446
2	0.5000	0.9100	0.9000	3	2.6000	1.5350	-0.9798	-1.3165
2	1.0000	0.9100	0.9000	3	2.6000	1.5350	-0.0698	-0.4124
2	1.5000	0.9100	0.9000	3	2.6000	1.5350	0.8402	0.6180
2	0.5000	0.9300	0.9000	3	2.6000	1.5350	-1.2710	-1.5712
2	1.0000	0.9300	0.9000	3	2.6000	1.5350	-0.3610	-0.5371
2	1.5000	0.9300	0.9000	3	2.6000	1.5350	0.5490	0.5145
2	0.5000	0.8700	0.9000	3	4.0000	1.4500	-0.9365	-1.1746
2	1.0000	0.8700	0.9000	3	4.0000	1.4500	-0.0265	-0.3985

Table E.6 contd.

2	1.5000	0.8700	0.9000	3	4.0000	1.4500	0.8835	0.4999
2	0.5000	0.8900	0.9000	3	4.0000	1.4500	-1.0949	-1.1312
2	1.0000	0.8900	0.9000	3	4.0000	1.4500	-0.1849	-0.2537
2	1.5000	0.8900	0.9000	3	4.0000	1.4500	0.7251	0.6228
2	0.5000	0.9100	0.9000	3	4.0000	1.4500	-1.3438	-0.7829
2	1.0000	0.9100	0.9000	3	4.0000	1.4500	-0.4338	0.0407
2	1.5000	0.9100	0.9000	3	4.0000	1.4500	0.4762	0.7441
2	0.5000	0.8700	0.9000	3	4.0000	1.5350	-0.9365	-0.9535
2	1.0000	0.8700	0.9000	3	4.0000	1.5350	-0.0265	-0.0371
2	1.5000	0.8700	0.9000	3	4.0000	1.5350	0.8835	0.9642
2	0.5000	0.8900	0.9000	3	4.0000	1.5350	-1.0949	-1.1968
2	1.0000	0.8900	0.9000	3	4.0000	1.5350	-0.1849	-0.0867
2	1.5000	0.8900	0.9000	3	4.0000	1.5350	0.7251	0.9928
2	0.5000	0.9100	0.9000	3	4.0000	1.5350	-1.3438	-1.1579
2	1.0000	0.9100	0.9000	3	4.0000	1.5350	-0.4338	-0.0077
2	1.5000	0.9100	0.9000	3	4.0000	1.5350	0.4762	1.0019
2	0.5000	0.8700	0.9000	6	2.6000	1.4500	-0.9188	-0.5263
2	1.0000	0.8700	0.9000	6	2.6000	1.4500	0.0437	0.2134
2	1.5000	0.8700	0.9000	6	2.6000	1.4500	1.0062	1.0515
2	2.0000	0.8700	0.9000	6	2.6000	1.4500	1.9687	1.9604
2	0.5000	0.8900	0.9000	6	2.6000	1.4500	-1.0217	-1.0176
2	1.0000	0.8900	0.9000	6	2.6000	1.4500	-0.0592	-0.2083
2	1.5000	0.8900	0.9000	6	2.6000	1.4500	0.9033	0.7547
2	2.0000	0.8900	0.9000	6	2.6000	1.4500	1.8658	1.7789
2	0.5000	0.9100	0.9000	6	2.6000	1.4500	-1.1835	-1.3654
2	1.0000	0.9100	0.9000	6	2.6000	1.4500	-0.2210	-0.4320
2	1.5000	0.9100	0.9000	6	2.6000	1.4500	0.7415	0.6212
2	2.0000	0.9100	0.9000	6	2.6000	1.4500	1.7040	1.6378
2	0.5000	0.9300	0.9000	6	2.6000	1.4500	-1.4747	-1.4098
2	1.0000	0.9300	0.9000	6	2.6000	1.4500	-0.5122	-0.3732
2	1.5000	0.9300	0.9000	6	2.6000	1.4500	0.4503	0.6482
2	2.0000	0.9300	0.9000	6	2.6000	1.4500	1.4128	1.5252
2	0.5000	0.8700	0.9000	6	2.6000	1.5350	-0.9188	-0.5276
2	1.0000	0.8700	0.9000	6	2.6000	1.5350	0.0437	0.0688
2	1.5000	0.8700	0.9000	6	2.6000	1.5350	1.0062	0.8235
2	2.0000	0.8700	0.9000	6	2.6000	1.5350	1.9687	1.7502
2	0.5000	0.8900	0.9000	6	2.6000	1.5350	-1.0217	-0.9151
2	1.0000	0.8900	0.9000	6	2.6000	1.5350	-0.0592	-0.1655
2	1.5000	0.8900	0.9000	6	2.6000	1.5350	0.9033	0.7754
2	2.0000	0.8900	0.9000	6	2.6000	1.5350	1.8658	1.8385
2	0.5000	0.9100	0.9000	6	2.6000	1.5350	-1.1835	-1.3372
2	1.0000	0.9100	0.9000	6	2.6000	1.5350	-0.2210	-0.4218
2	1.5000	0.9100	0.9000	6	2.6000	1.5350	0.7415	0.6683
2	2.0000	0.9100	0.9000	6	2.6000	1.5350	1.7040	1.7673
2	0.5000	0.9300	0.9000	6	2.6000	1.5350	-1.4747	-1.6851
2	1.0000	0.9300	0.9000	6	2.6000	1.5350	-0.5122	-0.6516
2	1.5000	0.9300	0.9000	6	2.6000	1.5350	0.4503	0.4740
2	2.0000	0.9300	0.9000	6	2.6000	1.5350	1.4128	1.5149
2	0.5000	0.8700	0.9000	6	4.0000	1.4500	-1.1402	-1.2229
2	1.0000	0.8700	0.9000	6	4.0000	1.4500	-0.1777	-0.5384
2	1.5000	0.8700	0.9000	6	4.0000	1.4500	0.7848	0.3122
2	2.0000	0.8700	0.9000	6	4.0000	1.4500	1.7473	1.2256
2	0.5000	0.8900	0.9000	6	4.0000	1.4500	-1.2986	-1.3123
2	1.0000	0.8900	0.9000	6	4.0000	1.4500	-0.3361	-0.4861
2	1.5000	0.8900	0.9000	6	4.0000	1.4500	0.6264	0.3976
2	2.0000	0.8900	0.9000	6	4.0000	1.4500	1.5889	1.2159
2	0.5000	0.9100	0.9000	6	4.0000	1.4500	-1.5475	-1.0502
2	1.0000	0.9100	0.9000	6	4.0000	1.4500	-0.5850	-0.2001
2	1.5000	0.9100	0.9000	6	4.0000	1.4500	0.3775	0.5607
2	2.0000	0.9100	0.9000	6	4.0000	1.4500	1.3400	1.1862
2	0.5000	0.8700	0.9000	6	4.0000	1.5350	-1.1402	-0.9877
2	1.0000	0.8700	0.9000	6	4.0000	1.5350	-0.1777	-0.0980
2	1.5000	0.8700	0.9000	6	4.0000	1.5350	0.7848	0.9305
2	2.0000	0.8700	0.9000	6	4.0000	1.5350	1.7473	1.9414
2	0.5000	0.8900	0.9000	6	4.0000	1.5350	-1.2986	-1.3875
2	1.0000	0.8900	0.9000	6	4.0000	1.5350	-0.3361	-0.3111
2	1.5000	0.8900	0.9000	6	4.0000	1.5350	0.6264	0.8355
2	2.0000	0.8900	0.9000	6	4.0000	1.5350	1.5889	1.8461
2	0.5000	0.9100	0.9000	6	4.0000	1.5350	-1.5475	-1.4786
2	1.0000	0.9100	0.9000	6	4.0000	1.5350	-0.5850	-0.3283
2	1.5000	0.9100	0.9000	6	4.0000	1.5350	0.3775	0.7725
2	2.0000	0.9100	0.9000	6	4.0000	1.5350	1.3400	1.6832
2	1.0000	0.8700	0.9000	9	2.6000	1.4500	-0.1075	0.3181
2	1.5000	0.8700	0.9000	9	2.6000	1.4500	0.9075	1.1577
2	2.0000	0.8700	0.9000	9	2.6000	1.4500	1.9225	2.0723

Table E.6 contd.

2	1.0000	0.8900	0.9000	9	2.6000	1.4500	-0.2104	-0.1083
2	1.5000	0.8900	0.9000	9	2.6000	1.4500	0.8046	0.8084
2	2.0000	0.8900	0.9000	9	2.6000	1.4500	1.8196	1.8159
2	1.0000	0.9100	0.9000	9	2.6000	1.4500	-0.3722	-0.4030
2	1.5000	0.9100	0.9000	9	2.6000	1.4500	0.6428	0.6058
2	2.0000	0.9100	0.9000	9	2.6000	1.4500	1.6578	1.6219
2	1.0000	0.9300	0.9000	9	2.6000	1.4500	-0.6634	-0.4429
2	1.5000	0.9300	0.9000	9	2.6000	1.4500	0.3516	0.5875
2	2.0000	0.9300	0.9000	9	2.6000	1.4500	1.3666	1.4922
2	1.0000	0.8700	0.9000	9	2.6000	1.5350	-0.1075	0.2474
2	1.5000	0.8700	0.9000	9	2.6000	1.5350	0.9075	1.0040
2	2.0000	0.8700	0.9000	9	2.6000	1.5350	1.9225	1.9041
2	1.0000	0.8900	0.9000	9	2.6000	1.5350	-0.2104	-0.0638
2	1.5000	0.8900	0.9000	9	2.6000	1.5350	0.8046	0.8760
2	2.0000	0.8900	0.9000	9	2.6000	1.5350	1.8196	1.9526
2	1.0000	0.9100	0.9000	9	2.6000	1.5350	-0.3722	-0.3911
2	1.5000	0.9100	0.9000	9	2.6000	1.5350	0.6428	0.6991
2	2.0000	0.9100	0.9000	9	2.6000	1.5350	1.6578	1.8546
2	1.0000	0.9300	0.9000	9	2.6000	1.5350	-0.6634	-0.6770
2	1.5000	0.9300	0.9000	9	2.6000	1.5350	0.3516	0.4533
2	2.0000	0.9300	0.9000	9	2.6000	1.5350	1.3666	1.5537
2	1.0000	0.8700	0.9000	9	4.0000	1.4500	-0.3289	-0.4843
2	1.5000	0.8700	0.9000	9	4.0000	1.4500	0.6861	0.2902
2	2.0000	0.8700	0.9000	9	4.0000	1.4500	1.7011	1.1821
2	1.0000	0.8900	0.9000	9	4.0000	1.4500	-0.4873	-0.5502
2	1.5000	0.8900	0.9000	9	4.0000	1.4500	0.5277	0.3249
2	2.0000	0.8900	0.9000	9	4.0000	1.4500	1.5427	1.1799
2	1.0000	0.9100	0.9000	9	4.0000	1.4500	-0.7362	-0.3285
2	1.5000	0.9100	0.9000	9	4.0000	1.4500	0.2788	0.4994
2	2.0000	0.9100	0.9000	9	4.0000	1.4500	1.2938	1.1967
2	1.0000	0.8700	0.9000	9	4.0000	1.5350	-0.3289	-0.1763
2	1.5000	0.8700	0.9000	9	4.0000	1.5350	0.6861	0.8433
2	2.0000	0.8700	0.9000	9	4.0000	1.5350	1.7011	1.9199
2	1.0000	0.8900	0.9000	9	4.0000	1.5350	-0.4873	-0.4787
2	1.5000	0.8900	0.9000	9	4.0000	1.5350	0.5277	0.6696
2	2.0000	0.8900	0.9000	9	4.0000	1.5350	1.5427	1.7731
2	1.0000	0.9100	0.9000	9	4.0000	1.5350	-0.7362	-0.5533
2	1.5000	0.9100	0.9000	9	4.0000	1.5350	0.2788	0.5821
2	2.0000	0.9100	0.9000	9	4.0000	1.5350	1.2938	1.5832
2	1.5000	0.8700	0.9000	12	2.6000	1.4500	0.8088	1.2563
2	2.0000	0.8700	0.9000	12	2.6000	1.4500	1.8763	2.1175
2	1.5000	0.8900	0.9000	12	2.6000	1.4500	0.7059	0.9169
2	2.0000	0.8900	0.9000	12	2.6000	1.4500	1.7734	1.8523
2	1.5000	0.9100	0.9000	12	2.6000	1.4500	0.5441	0.6719
2	2.0000	0.9100	0.9000	12	2.6000	1.4500	1.6116	1.6437
2	1.5000	0.9300	0.9000	12	2.6000	1.4500	0.2529	0.5919
2	2.0000	0.9300	0.9000	12	2.6000	1.4500	1.3204	1.5011
2	1.5000	0.8700	0.9000	12	2.6000	1.5350	0.8088	1.0257
2	2.0000	0.8700	0.9000	12	2.6000	1.5350	1.8763	1.8734
2	1.5000	0.8900	0.9000	12	2.6000	1.5350	0.7059	0.8333
2	2.0000	0.8900	0.9000	12	2.6000	1.5350	1.7734	1.8691
2	1.5000	0.9100	0.9000	12	2.6000	1.5350	0.5441	0.6327
2	2.0000	0.9100	0.9000	12	2.6000	1.5350	1.6116	1.7685
2	1.5000	0.9300	0.9000	12	2.6000	1.5350	0.2529	0.4110
2	2.0000	0.9300	0.9000	12	2.6000	1.5350	1.3204	1.4973
2	1.5000	0.8700	0.9000	12	4.0000	1.4500	0.5874	0.3921
2	2.0000	0.8700	0.9000	12	4.0000	1.4500	1.6549	1.2252
2	1.5000	0.8900	0.9000	12	4.0000	1.4500	0.4290	0.3598
2	2.0000	0.8900	0.9000	12	4.0000	1.4500	1.4965	1.2286
2	1.5000	0.9100	0.9000	12	4.0000	1.4500	0.1801	0.5202
2	2.0000	0.9100	0.9000	12	4.0000	1.4500	1.2476	1.2877
2	1.5000	0.8700	0.9000	12	4.0000	1.5350	0.5874	0.6855
2	2.0000	0.8700	0.9000	12	4.0000	1.5350	1.6549	1.7936
2	1.5000	0.8900	0.9000	12	4.0000	1.5350	0.4290	0.4878
2	2.0000	0.8900	0.9000	12	4.0000	1.5350	1.4965	1.6317
2	1.5000	0.9100	0.9000	12	4.0000	1.5350	0.1801	0.4141
2	2.0000	0.9100	0.9000	12	4.0000	1.5350	1.2476	1.4627
2	0.5000	0.8700	0.9250	3	2.6000	1.4500	-0.4001	-0.2993
2	1.0000	0.8700	0.9250	3	2.6000	1.4500	0.5099	0.5923
2	1.5000	0.8700	0.9250	3	2.6000	1.4500	1.4199	1.5755
2	0.5000	0.8900	0.9250	3	2.6000	1.4500	-0.5030	-0.6506
2	1.0000	0.8900	0.9250	3	2.6000	1.4500	0.4070	0.4072
2	1.5000	0.8900	0.9250	3	2.6000	1.4500	1.3170	1.5183
2	0.5000	0.9100	0.9250	3	2.6000	1.4500	-0.6648	-0.8773
2	1.0000	0.9100	0.9250	3	2.6000	1.4500	0.2452	0.2935

Table E.6 contd.

2	1.5000	0.9100	0.9250	3	2.6000	1.4500	1.1552	1.4152
2	0.5000	0.9300	0.9250	3	2.6000	1.4500	-0.9560	-0.8789
2	1.0000	0.9300	0.9250	3	2.6000	1.4500	-0.0460	0.2674
2	1.5000	0.9300	0.9250	3	2.6000	1.4500	0.8640	1.2717
2	0.5000	0.8700	0.9250	3	2.6000	1.5350	-0.4001	-0.3389
2	1.0000	0.8700	0.9250	3	2.6000	1.5350	0.5099	0.2795
2	1.5000	0.8700	0.9250	3	2.6000	1.5350	1.4199	1.0848
2	0.5000	0.8900	0.9250	3	2.6000	1.5350	-0.5030	-0.4692
2	1.0000	0.8900	0.9250	3	2.6000	1.5350	0.4070	0.3235
2	1.5000	0.8900	0.9250	3	2.6000	1.5350	1.3170	1.2457
2	0.5000	0.9100	0.9250	3	2.6000	1.5350	-0.6648	-0.6459
2	1.0000	0.9100	0.9250	3	2.6000	1.5350	0.2452	0.2808
2	1.5000	0.9100	0.9250	3	2.6000	1.5350	1.1552	1.2115
2	0.5000	0.9300	0.9250	3	2.6000	1.5350	-0.9560	-0.8807
2	1.0000	0.9300	0.9250	3	2.6000	1.5350	-0.0460	0.0747
2	1.5000	0.9300	0.9250	3	2.6000	1.5350	0.8640	0.9376
2	0.5000	0.8700	0.9250	3	4.0000	1.4500	-0.6215	-0.7915
2	1.0000	0.8700	0.9250	3	4.0000	1.4500	0.2885	0.1613
2	1.5000	0.8700	0.9250	3	4.0000	1.4500	1.1985	1.1179
2	0.5000	0.8900	0.9250	3	4.0000	1.4500	-0.7799	-0.7592
2	1.0000	0.8900	0.9250	3	4.0000	1.4500	0.1301	0.2250
2	1.5000	0.8900	0.9250	3	4.0000	1.4500	1.0401	1.1017
2	0.5000	0.9100	0.9250	3	4.0000	1.4500	-1.0288	-0.5097
2	1.0000	0.9100	0.9250	3	4.0000	1.4500	-0.1188	0.3410
2	1.5000	0.9100	0.9250	3	4.0000	1.4500	0.7912	1.0351
2	0.5000	0.8700	0.9250	3	4.0000	1.5350	-0.6215	-0.7630
2	1.0000	0.8700	0.9250	3	4.0000	1.5350	0.2885	0.1876
2	1.5000	0.8700	0.9250	3	4.0000	1.5350	1.1985	1.1899
2	0.5000	0.8900	0.9250	3	4.0000	1.5350	-0.7799	-0.9539
2	1.0000	0.8900	0.9250	3	4.0000	1.5350	0.1301	0.1225
2	1.5000	0.8900	0.9250	3	4.0000	1.5350	1.0401	1.1193
2	0.5000	0.9100	0.9250	3	4.0000	1.5350	-1.0288	-1.0131
2	1.0000	0.9100	0.9250	3	4.0000	1.5350	-0.1188	0.0550
2	1.5000	0.9100	0.9250	3	4.0000	1.5350	0.7912	0.9824
2	0.5000	0.8700	0.9250	6	2.6000	1.4500	-0.6038	-0.4108
2	1.0000	0.8700	0.9250	6	2.6000	1.4500	0.3587	0.4974
2	1.5000	0.8700	0.9250	6	2.6000	1.4500	1.3212	1.5169
2	2.0000	0.8700	0.9250	6	2.6000	1.4500	2.2837	2.5495
2	0.5000	0.8900	0.9250	6	2.6000	1.4500	-0.7067	-0.7986
2	1.0000	0.8900	0.9250	6	2.6000	1.4500	0.2558	0.2260
2	1.5000	0.8900	0.9250	6	2.6000	1.4500	1.2183	1.3654
2	2.0000	0.8900	0.9250	6	2.6000	1.4500	2.1808	2.4360
2	0.5000	0.9100	0.9250	6	2.6000	1.4500	-0.8685	-1.0840
2	1.0000	0.9100	0.9250	6	2.6000	1.4500	0.0940	0.0376
2	1.5000	0.9100	0.9250	6	2.6000	1.4500	1.0565	1.1900
2	2.0000	0.9100	0.9250	6	2.6000	1.4500	2.0190	2.1792
2	0.5000	0.9300	0.9250	6	2.6000	1.4500	-1.1597	-1.1468
2	1.0000	0.9300	0.9250	6	2.6000	1.4500	-0.1972	-0.0181
2	1.5000	0.9300	0.9250	6	2.6000	1.4500	0.7653	1.0160
2	2.0000	0.9300	0.9250	6	2.6000	1.4500	1.7278	1.8515
2	0.5000	0.8700	0.9250	6	2.6000	1.5350	-0.6038	-0.2274
2	1.0000	0.8700	0.9250	6	2.6000	1.5350	0.3587	0.4106
2	1.5000	0.8700	0.9250	6	2.6000	1.5350	1.3212	1.2025
2	2.0000	0.8700	0.9250	6	2.6000	1.5350	2.2837	2.1349
2	0.5000	0.8900	0.9250	6	2.6000	1.5350	-0.7067	-0.4711
2	1.0000	0.8900	0.9250	6	2.6000	1.5350	0.2558	0.3468
2	1.5000	0.8900	0.9250	6	2.6000	1.5350	1.2183	1.2990
2	2.0000	0.8900	0.9250	6	2.6000	1.5350	2.1808	2.2808
2	0.5000	0.9100	0.9250	6	2.6000	1.5350	-0.8685	-0.7488
2	1.0000	0.9100	0.9250	6	2.6000	1.5350	0.0940	0.2216
2	1.5000	0.9100	0.9250	6	2.6000	1.5350	1.0565	1.2344
2	2.0000	0.9100	0.9250	6	2.6000	1.5350	2.0190	2.1450
2	0.5000	0.9300	0.9250	6	2.6000	1.5350	-1.1597	-1.0479
2	1.0000	0.9300	0.9250	6	2.6000	1.5350	-0.1972	-0.0398
2	1.5000	0.9300	0.9250	6	2.6000	1.5350	0.7653	0.9111
2	2.0000	0.9300	0.9250	6	2.6000	1.5350	1.7278	1.7145
2	0.5000	0.8700	0.9250	6	4.0000	1.4500	-0.8252	-1.0013
2	1.0000	0.8700	0.9250	6	4.0000	1.4500	0.1373	-0.0856
2	1.5000	0.8700	0.9250	6	4.0000	1.4500	1.0998	0.9126
2	2.0000	0.8700	0.9250	6	4.0000	1.4500	2.0623	1.8453
2	0.5000	0.8900	0.9250	6	4.0000	1.4500	-0.9836	-1.0911
2	1.0000	0.8900	0.9250	6	4.0000	1.4500	-0.0211	-0.1031
2	1.5000	0.8900	0.9250	6	4.0000	1.4500	0.9414	0.8423
2	2.0000	0.8900	0.9250	6	4.0000	1.4500	1.9039	1.6491
2	0.5000	0.9100	0.9250	6	4.0000	1.4500	-1.2325	-0.8911

Table E.6 contd.

2	1.0000	0.9100	0.9250	6	4.0000	1.4500	-0.2700	0.0210
2	1.5000	0.9100	0.9250	6	4.0000	1.4500	0.6925	0.7912
2	2.0000	0.9100	0.9250	6	4.0000	1.4500	1.6550	1.4238
2	0.5000	0.8700	0.9250	6	4.0000	1.5350	-0.8252	-0.8379
2	1.0000	0.8700	0.9250	6	4.0000	1.5350	0.1373	0.1339
2	1.5000	0.8700	0.9250	6	4.0000	1.5350	1.0998	1.2170
2	2.0000	0.8700	0.9250	6	4.0000	1.5350	2.0623	2.2285
2	0.5000	0.8900	0.9250	6	4.0000	1.5350	-0.9836	-1.1963
2	1.0000	0.8900	0.9250	6	4.0000	1.5350	-0.0211	-0.0803
2	1.5000	0.8900	0.9250	6	4.0000	1.5350	0.9414	1.0396
2	2.0000	0.8900	0.9250	6	4.0000	1.5350	1.9039	1.9930
2	0.5000	0.9100	0.9250	6	4.0000	1.5350	-1.2325	-1.3907
2	1.0000	0.9100	0.9250	6	4.0000	1.5350	-0.2700	-0.2691
2	1.5000	0.9100	0.9250	6	4.0000	1.5350	0.6925	0.7812
2	2.0000	0.9100	0.9250	6	4.0000	1.5350	1.6550	1.6655
2	1.0000	0.8700	0.9250	9	2.6000	1.4500	0.2075	0.3584
2	1.5000	0.8700	0.9250	9	2.6000	1.4500	1.2225	1.3639
2	2.0000	0.8700	0.9250	9	2.6000	1.4500	2.2375	2.4283
2	1.0000	0.8900	0.9250	9	2.6000	1.4500	0.1046	0.0535
2	1.5000	0.8900	0.9250	9	2.6000	1.4500	1.1196	1.1517
2	2.0000	0.8900	0.9250	9	2.6000	1.4500	2.1346	2.2628
2	1.0000	0.9100	0.9250	9	2.6000	1.4500	-0.0572	-0.1585
2	1.5000	0.9100	0.9250	9	2.6000	1.4500	0.9578	0.9565
2	2.0000	0.9100	0.9250	9	2.6000	1.4500	1.9728	1.9788
2	1.0000	0.9300	0.9250	9	2.6000	1.4500	-0.3484	-0.2238
2	1.5000	0.9300	0.9250	9	2.6000	1.4500	0.6666	0.7956
2	2.0000	0.9300	0.9250	9	2.6000	1.4500	1.6816	1.6397
2	1.0000	0.8700	0.9250	9	2.6000	1.5350	0.2075	0.4807
2	1.5000	0.8700	0.9250	9	2.6000	1.5350	1.2225	1.2403
2	2.0000	0.8700	0.9250	9	2.6000	1.5350	2.2375	2.1331
2	1.0000	0.8900	0.9250	9	2.6000	1.5350	0.1046	0.2945
2	1.5000	0.8900	0.9250	9	2.6000	1.5350	1.1196	1.2421
2	2.0000	0.8900	0.9250	9	2.6000	1.5350	2.1346	2.2536
2	1.0000	0.9100	0.9250	9	2.6000	1.5350	-0.0572	0.0915
2	1.5000	0.9100	0.9250	9	2.6000	1.5350	0.9578	1.1388
2	2.0000	0.9100	0.9250	9	2.6000	1.5350	1.9728	2.1309
2	1.0000	0.9300	0.9250	9	2.6000	1.5350	-0.3484	-0.1740
2	1.5000	0.9300	0.9250	9	2.6000	1.5350	0.6666	0.8159
2	2.0000	0.9300	0.9250	9	2.6000	1.5350	1.6816	1.6859
2	1.0000	0.8700	0.9250	9	4.0000	1.4500	-0.0139	-0.1657
2	1.5000	0.8700	0.9250	9	4.0000	1.4500	1.0011	0.8274
2	2.0000	0.8700	0.9250	9	4.0000	1.4500	2.0161	1.8192
2	1.0000	0.8900	0.9250	9	4.0000	1.4500	-0.1723	-0.2681
2	1.5000	0.8900	0.9250	9	4.0000	1.4500	0.8427	0.7197
2	2.0000	0.8900	0.9250	9	4.0000	1.4500	1.8577	1.5965
2	1.0000	0.9100	0.9250	9	4.0000	1.4500	-0.4212	-0.1608
2	1.5000	0.9100	0.9250	9	4.0000	1.4500	0.5938	0.6889
2	2.0000	0.9100	0.9250	9	4.0000	1.4500	1.6088	1.3845
2	1.0000	0.8700	0.9250	9	4.0000	1.5350	-0.0139	0.0455
2	1.5000	0.8700	0.9250	9	4.0000	1.5350	1.0011	1.1784
2	2.0000	0.8700	0.9250	9	4.0000	1.5350	2.0161	2.3018
2	1.0000	0.8900	0.9250	9	4.0000	1.5350	-0.1723	-0.2518
2	1.5000	0.8900	0.9250	9	4.0000	1.5350	0.8427	0.9401
2	2.0000	0.8900	0.9250	9	4.0000	1.5350	1.8577	2.0220
2	1.0000	0.9100	0.9250	9	4.0000	1.5350	-0.4212	-0.4881
2	1.5000	0.9100	0.9250	9	4.0000	1.5350	0.5938	0.6284
2	2.0000	0.9100	0.9250	9	4.0000	1.5350	1.6088	1.6150
2	1.5000	0.8700	0.9250	12	2.6000	1.4500	1.1238	1.1388
2	2.0000	0.8700	0.9250	12	2.6000	1.4500	2.1913	2.1819
2	1.5000	0.8900	0.9250	12	2.6000	1.4500	1.0209	0.9365
2	2.0000	0.8900	0.9250	12	2.6000	1.4500	2.0884	2.0273
2	1.5000	0.9100	0.9250	12	2.6000	1.4500	0.8591	0.7853
2	2.0000	0.9100	0.9250	12	2.6000	1.4500	1.9266	1.8017
2	1.5000	0.9300	0.9250	12	2.6000	1.4500	0.5679	0.6743
2	2.0000	0.9300	0.9250	12	2.6000	1.4500	1.6354	1.5197
2	1.5000	0.8700	0.9250	12	2.6000	1.5350	1.1238	1.1164
2	2.0000	0.8700	0.9250	12	2.6000	1.5350	2.1913	1.9540
2	1.5000	0.8900	0.9250	12	2.6000	1.5350	1.0209	1.0407
2	2.0000	0.8900	0.9250	12	2.6000	1.5350	2.0884	2.0460
2	1.5000	0.9100	0.9250	12	2.6000	1.5350	0.8591	0.9381
2	2.0000	0.9100	0.9250	12	2.6000	1.5350	1.9266	1.9704
2	1.5000	0.9300	0.9250	12	2.6000	1.5350	0.5679	0.7020
2	2.0000	0.9300	0.9250	12	2.6000	1.5350	1.6354	1.6032
2	1.5000	0.8700	0.9250	12	4.0000	1.4500	0.9024	0.8571
2	2.0000	0.8700	0.9250	12	4.0000	1.4500	1.9699	1.8352

Table E.6 contd.

2	1.5000	0.8900	0.9250	12	4.0000	1.4500	0.7440	0.7425
2	2.0000	0.8900	0.9250	12	4.0000	1.4500	1.8115	1.6401
2	1.5000	0.9100	0.9250	12	4.0000	1.4500	0.4951	0.7460
2	2.0000	0.9100	0.9250	12	4.0000	1.4500	1.5626	1.4794
2	1.5000	0.8700	0.9250	12	4.0000	1.5350	0.9024	1.0178
2	2.0000	0.8700	0.9250	12	4.0000	1.5350	1.9699	2.2053
2	1.5000	0.8900	0.9250	12	4.0000	1.5350	0.7440	0.7976
2	2.0000	0.8900	0.9250	12	4.0000	1.5350	1.8115	1.9421
2	1.5000	0.9100	0.9250	12	4.0000	1.5350	0.4951	0.5375
2	2.0000	0.9100	0.9250	12	4.0000	1.5350	1.5626	1.5488
2	0.5000	0.8700	0.9500	3	2.6000	1.4500	-0.0851	-0.0929
2	1.0000	0.8700	0.9500	3	2.6000	1.4500	0.8249	0.8974
2	1.5000	0.8700	0.9500	3	2.6000	1.4500	1.7349	1.9658
2	0.5000	0.8900	0.9500	3	2.6000	1.4500	-0.1880	-0.3047
2	1.0000	0.8900	0.9500	3	2.6000	1.4500	0.7220	0.8360
2	1.5000	0.8900	0.9500	3	2.6000	1.4500	1.6320	1.9444
2	0.5000	0.9100	0.9500	3	2.6000	1.4500	-0.3498	-0.4947
2	1.0000	0.9100	0.9500	3	2.6000	1.4500	0.5602	0.6839
2	1.5000	0.9100	0.9500	3	2.6000	1.4500	1.4702	1.7260
2	0.5000	0.9300	0.9500	3	2.6000	1.4500	-0.6410	-0.6121
2	1.0000	0.9300	0.9500	3	2.6000	1.4500	0.2690	0.4726
2	1.5000	0.9300	0.9500	3	2.6000	1.4500	1.1790	1.3958
2	0.5000	0.8700	0.9500	3	2.6000	1.5350	-0.0851	-0.0007
2	1.0000	0.8700	0.9500	3	2.6000	1.5350	0.8249	0.7009
2	1.5000	0.8700	0.9500	3	2.6000	1.5350	1.7349	1.5384
2	0.5000	0.8900	0.9500	3	2.6000	1.5350	-0.1880	0.0437
2	1.0000	0.8900	0.9500	3	2.6000	1.5350	0.7220	0.8842
2	1.5000	0.8900	0.9500	3	2.6000	1.5350	1.6320	1.7432
2	0.5000	0.9100	0.9500	3	2.6000	1.5350	-0.3498	0.0349
2	1.0000	0.9100	0.9500	3	2.6000	1.5350	0.5602	0.9021
2	1.5000	0.9100	0.9500	3	2.6000	1.5350	1.4702	1.6511
2	0.5000	0.9300	0.9500	3	2.6000	1.5350	-0.6410	-0.1530
2	1.0000	0.9300	0.9500	3	2.6000	1.5350	0.2690	0.6229
2	1.5000	0.9300	0.9500	3	2.6000	1.5350	1.1790	1.2306
2	0.5000	0.8700	0.9500	3	4.0000	1.4500	-0.3065	-0.5166
2	1.0000	0.8700	0.9500	3	4.0000	1.4500	0.6035	0.5768
2	1.5000	0.8700	0.9500	3	4.0000	1.4500	1.5135	1.5517
2	0.5000	0.8900	0.9500	3	4.0000	1.4500	-0.4649	-0.5548
2	1.0000	0.8900	0.9500	3	4.0000	1.4500	0.4451	0.4987
2	1.5000	0.8900	0.9500	3	4.0000	1.4500	1.3551	1.3718
2	0.5000	0.9100	0.9500	3	4.0000	1.4500	-0.7138	-0.4718
2	1.0000	0.9100	0.9500	3	4.0000	1.4500	0.1962	0.4128
2	1.5000	0.9100	0.9500	3	4.0000	1.4500	1.1062	1.1364
2	0.5000	0.8700	0.9500	3	4.0000	1.5350	-0.3065	-0.5347
2	1.0000	0.8700	0.9500	3	4.0000	1.5350	0.6035	0.3995
2	1.5000	0.8700	0.9500	3	4.0000	1.5350	1.5135	1.3249
2	0.5000	0.8900	0.9500	3	4.0000	1.5350	-0.4649	-0.7226
2	1.0000	0.8900	0.9500	3	4.0000	1.5350	0.4451	0.2275
2	1.5000	0.8900	0.9500	3	4.0000	1.5350	1.3551	1.0771
2	0.5000	0.9100	0.9500	3	4.0000	1.5350	-0.7138	-0.9387
2	1.0000	0.9100	0.9500	3	4.0000	1.5350	0.1962	-0.0409
2	1.5000	0.9100	0.9500	3	4.0000	1.5350	1.1062	0.7550
2	0.5000	0.8700	0.9500	6	2.6000	1.4500	-0.2888	-0.3431
2	1.0000	0.8700	0.9500	6	2.6000	1.4500	0.6737	0.6205
2	1.5000	0.8700	0.9500	6	2.6000	1.4500	1.6362	1.7046
2	2.0000	0.8700	0.9500	6	2.6000	1.4500	2.5987	2.7644
2	0.5000	0.8900	0.9500	6	2.6000	1.4500	-0.3917	-0.5948
2	1.0000	0.8900	0.9500	6	2.6000	1.4500	0.5708	0.5040
2	1.5000	0.8900	0.9500	6	2.6000	1.4500	1.5333	1.6584
2	2.0000	0.8900	0.9500	6	2.6000	1.4500	2.4958	2.6619
2	0.5000	0.9100	0.9500	6	2.6000	1.4500	-0.5535	-0.7832
2	1.0000	0.9100	0.9500	6	2.6000	1.4500	0.4090	0.3499
2	1.5000	0.9100	0.9500	6	2.6000	1.4500	1.3715	1.4333
2	2.0000	0.9100	0.9500	6	2.6000	1.4500	2.3340	2.3067
2	0.5000	0.9300	0.9500	6	2.6000	1.4500	-0.8447	-0.8664
2	1.0000	0.9300	0.9500	6	2.6000	1.4500	0.1178	0.1726
2	1.5000	0.9300	0.9500	6	2.6000	1.4500	1.0803	1.0956
2	2.0000	0.9300	0.9500	6	2.6000	1.4500	2.0428	1.8379
2	0.5000	0.8700	0.9500	6	2.6000	1.5350	-0.2888	-0.0206
2	1.0000	0.8700	0.9500	6	2.6000	1.5350	0.6737	0.6460
2	1.5000	0.8700	0.9500	6	2.6000	1.5350	1.6362	1.4488
2	2.0000	0.8700	0.9500	6	2.6000	1.5350	2.5987	2.3285
2	0.5000	0.8900	0.9500	6	2.6000	1.5350	-0.3917	-0.0715
2	1.0000	0.8900	0.9500	6	2.6000	1.5350	0.5708	0.7624
2	1.5000	0.8900	0.9500	6	2.6000	1.5350	1.5333	1.6497

Table E.6 contd.

2	2.0000	0.8900	0.9500	6	2.6000	1.5350	2.4958	2.4707
2	0.5000	0.9100	0.9500	6	2.6000	1.5350	-0.5535	-0.1258
2	1.0000	0.9100	0.9500	6	2.6000	1.5350	0.4090	0.7890
2	1.5000	0.9100	0.9500	6	2.6000	1.5350	1.3715	1.6186
2	2.0000	0.9100	0.9500	6	2.6000	1.5350	2.3340	2.2803
2	0.5000	0.9300	0.9500	6	2.6000	1.5350	-0.8447	-0.2901
2	1.0000	0.9300	0.9500	6	2.6000	1.5350	0.1178	0.5580
2	1.5000	0.9300	0.9500	6	2.6000	1.5350	1.0803	1.2414
2	2.0000	0.9300	0.9500	6	2.6000	1.5350	2.0428	1.7595
2	0.5000	0.8700	0.9500	6	4.0000	1.4500	-0.5102	-0.8578
2	1.0000	0.8700	0.9500	6	4.0000	1.4500	0.4523	0.2651
2	1.5000	0.8700	0.9500	6	4.0000	1.4500	1.4148	1.3562
2	2.0000	0.8700	0.9500	6	4.0000	1.4500	2.3773	2.2618
2	0.5000	0.8900	0.9500	6	4.0000	1.4500	-0.6686	-0.9559
2	1.0000	0.8900	0.9500	6	4.0000	1.4500	0.2939	0.1440
2	1.5000	0.8900	0.9500	6	4.0000	1.4500	1.2564	1.1178
2	2.0000	0.8900	0.9500	6	4.0000	1.4500	2.2189	1.9033
2	0.5000	0.9100	0.9500	6	4.0000	1.4500	-0.9175	-0.8440
2	1.0000	0.9100	0.9500	6	4.0000	1.4500	0.0450	0.1007
2	1.5000	0.9100	0.9500	6	4.0000	1.4500	1.0075	0.8890
2	2.0000	0.9100	0.9500	6	4.0000	1.4500	1.9700	1.5412
2	0.5000	0.8700	0.9500	6	4.0000	1.5350	-0.5102	-0.6364
2	1.0000	0.8700	0.9500	6	4.0000	1.5350	0.4523	0.3680
2	1.5000	0.8700	0.9500	6	4.0000	1.5350	1.4148	1.4082
2	2.0000	0.8700	0.9500	6	4.0000	1.5350	2.3773	2.3343
2	0.5000	0.8900	0.9500	6	4.0000	1.5350	-0.6686	-0.9169
2	1.0000	0.8900	0.9500	6	4.0000	1.5350	0.2939	0.1309
2	1.5000	0.8900	0.9500	6	4.0000	1.5350	1.2564	1.1106
2	2.0000	0.8900	0.9500	6	4.0000	1.5350	2.2189	1.9383
2	0.5000	0.9100	0.9500	6	4.0000	1.5350	-0.9175	-1.2009
2	1.0000	0.9100	0.9500	6	4.0000	1.5350	0.0450	-0.2206
2	1.5000	0.9100	0.9500	6	4.0000	1.5350	1.0075	0.6790
2	2.0000	0.9100	0.9500	6	4.0000	1.5350	1.9700	1.4660
2	1.0000	0.8700	0.9500	9	2.6000	1.4500	0.5225	0.2775
2	1.5000	0.8700	0.9500	9	2.6000	1.4500	1.5375	1.3383
2	2.0000	0.8700	0.9500	9	2.6000	1.4500	2.5525	2.4554
2	1.0000	0.8900	0.9500	9	2.6000	1.4500	0.4196	0.1516
2	1.5000	0.8900	0.9500	9	2.6000	1.4500	1.4346	1.2929
2	2.0000	0.8900	0.9500	9	2.6000	1.4500	2.4496	2.3842
2	1.0000	0.9100	0.9500	9	2.6000	1.4500	0.2578	0.0567
2	1.5000	0.9100	0.9500	9	2.6000	1.4500	1.2728	1.1309
2	2.0000	0.9100	0.9500	9	2.6000	1.4500	2.2878	2.0660
2	1.0000	0.9300	0.9500	9	2.6000	1.4500	-0.0334	-0.0371
2	1.5000	0.9300	0.9500	9	2.6000	1.4500	0.9816	0.8614
2	2.0000	0.9300	0.9500	9	2.6000	1.4500	1.9966	1.6074
2	1.0000	0.8700	0.9500	9	2.6000	1.5350	0.5225	0.5666
2	1.5000	0.8700	0.9500	9	2.6000	1.5350	1.5375	1.3206
2	2.0000	0.8700	0.9500	9	2.6000	1.5350	2.5525	2.1815
2	1.0000	0.8900	0.9500	9	2.6000	1.5350	0.4196	0.5999
2	1.5000	0.8900	0.9500	9	2.6000	1.5350	1.4346	1.4959
2	2.0000	0.8900	0.9500	9	2.6000	1.5350	2.4496	2.3725
2	1.0000	0.9100	0.9500	9	2.6000	1.5350	0.2578	0.6121
2	1.5000	0.9100	0.9500	9	2.6000	1.5350	1.2728	1.5144
2	2.0000	0.9100	0.9500	9	2.6000	1.5350	2.2878	2.2719
2	1.0000	0.9300	0.9500	9	2.6000	1.5350	-0.0334	0.4434
2	1.5000	0.9300	0.9500	9	2.6000	1.5350	0.9816	1.2081
2	2.0000	0.9300	0.9500	9	2.6000	1.5350	1.9966	1.8043
2	1.0000	0.8700	0.9500	9	4.0000	1.4500	0.3011	0.1170
2	1.5000	0.8700	0.9500	9	4.0000	1.4500	1.3161	1.2515
2	2.0000	0.8700	0.9500	9	4.0000	1.4500	2.3311	2.2480
2	1.0000	0.8900	0.9500	9	4.0000	1.4500	0.1427	-0.0008
2	1.5000	0.8900	0.9500	9	4.0000	1.4500	1.1577	1.0197
2	2.0000	0.8900	0.9500	9	4.0000	1.4500	2.1727	1.8589
2	1.0000	0.9100	0.9500	9	4.0000	1.4500	-0.1062	0.0090
2	1.5000	0.9100	0.9500	9	4.0000	1.4500	0.9088	0.8386
2	2.0000	0.9100	0.9500	9	4.0000	1.4500	1.9238	1.5050
2	1.0000	0.8700	0.9500	9	4.0000	1.5350	0.3011	0.3486
2	1.5000	0.8700	0.9500	9	4.0000	1.5350	1.3161	1.4610
2	2.0000	0.8700	0.9500	9	4.0000	1.5350	2.3311	2.4829
2	1.0000	0.8900	0.9500	9	4.0000	1.5350	0.1427	0.1089
2	1.5000	0.8900	0.9500	9	4.0000	1.5350	1.1577	1.1787
2	2.0000	0.8900	0.9500	9	4.0000	1.5350	2.1727	2.0968
2	1.0000	0.9100	0.9500	9	4.0000	1.5350	-0.1062	-0.2425
2	1.5000	0.9100	0.9500	9	4.0000	1.5350	0.9088	0.7117
2	2.0000	0.9100	0.9500	9	4.0000	1.5350	1.9238	1.5526

Table E.6 contd.

2	1.5000	0.8700	0.9500	12	2.6000	1.4500	1.4388	0.9665
2	2.0000	0.8700	0.9500	12	2.6000	1.4500	2.5063	2.0951
2	1.5000	0.8900	0.9500	12	2.6000	1.4500	1.3359	0.9556
2	2.0000	0.8900	0.9500	12	2.6000	1.4500	2.4034	2.0864
2	1.5000	0.9100	0.9500	12	2.6000	1.4500	1.1741	0.9051
2	2.0000	0.9100	0.9500	12	2.6000	1.4500	2.2416	1.8769
2	1.5000	0.9300	0.9500	12	2.6000	1.4500	0.8829	0.7474
2	2.0000	0.9300	0.9500	12	2.6000	1.4500	1.9504	1.4988
2	1.5000	0.8700	0.9500	12	2.6000	1.5350	1.4388	1.1676
2	2.0000	0.8700	0.9500	12	2.6000	1.5350	2.5063	1.9891
2	1.5000	0.8900	0.9500	12	2.6000	1.5350	1.3359	1.3064
2	2.0000	0.8900	0.9500	12	2.6000	1.5350	2.4034	2.2211
2	1.5000	0.9100	0.9500	12	2.6000	1.5350	1.1741	1.3615
2	2.0000	0.9100	0.9500	12	2.6000	1.5350	2.2416	2.2140
2	1.5000	0.9300	0.9500	12	2.6000	1.5350	0.8829	1.1550
2	2.0000	0.9300	0.9500	12	2.6000	1.5350	1.9504	1.8372
2	1.5000	0.8700	0.9500	12	4.0000	1.4500	1.2174	1.2654
2	2.0000	0.8700	0.9500	12	4.0000	1.4500	2.2849	2.2702
2	1.5000	0.8900	0.9500	12	4.0000	1.4500	1.0590	1.0996
2	2.0000	0.8900	0.9500	12	4.0000	1.4500	2.1265	1.9368
2	1.5000	0.9100	0.9500	12	4.0000	1.4500	0.8101	0.9784
2	2.0000	0.9100	0.9500	12	4.0000	1.4500	1.8776	1.6326
2	1.5000	0.8700	0.9500	12	4.0000	1.5350	1.2174	1.4121
2	2.0000	0.8700	0.9500	12	4.0000	1.5350	2.2849	2.4835
2	1.5000	0.8900	0.9500	12	4.0000	1.5350	1.0590	1.2131
2	2.0000	0.8900	0.9500	12	4.0000	1.5350	2.1265	2.1744
2	1.5000	0.9100	0.9500	12	4.0000	1.5350	0.8101	0.8101
2	2.0000	0.9100	0.9500	12	4.0000	1.5350	1.8776	1.6420
3	2.0000	0.8700	0.9000	3	2.6000	1.4500	1.2024	0.8247
3	2.0000	0.8900	0.9000	3	2.6000	1.4500	1.0480	0.9225
3	2.0000	0.9100	0.9000	3	2.6000	1.4500	0.8053	1.1527
3	2.0000	0.8700	0.9000	3	2.6000	1.5350	1.2024	0.7073
3	2.0000	0.8900	0.9000	3	2.6000	1.5350	1.0480	0.6929
3	2.0000	0.9100	0.9000	3	2.6000	1.5350	0.8053	0.7266
3	2.0000	0.8700	0.9000	3	4.0000	1.4500	0.8702	0.7907
3	2.0000	0.8900	0.9000	3	4.0000	1.4500	0.6326	1.0529
3	2.0000	0.8700	0.9000	3	4.0000	1.5350	0.8702	0.7273
3	2.0000	0.8900	0.9000	3	4.0000	1.5350	0.6326	1.0136
3	2.0000	0.8700	0.9000	6	2.6000	1.4500	1.0281	0.9493
3	2.0000	0.8900	0.9000	6	2.6000	1.4500	0.8737	0.8374
3	2.0000	0.9100	0.9000	6	2.6000	1.4500	0.6310	0.9262
3	2.0000	0.8700	0.9000	6	2.6000	1.5350	1.0281	0.9921
3	2.0000	0.8900	0.9000	6	2.6000	1.5350	0.8737	0.8910
3	2.0000	0.9100	0.9000	6	2.6000	1.5350	0.6310	0.8042
3	2.0000	0.8700	0.9000	6	4.0000	1.4500	0.6959	0.5090
3	2.0000	0.8900	0.9000	6	4.0000	1.4500	0.4583	0.7436
3	2.0000	0.8700	0.9000	6	4.0000	1.5350	0.6959	0.7543
3	2.0000	0.8900	0.9000	6	4.0000	1.5350	0.4583	0.9099
3	2.0000	0.8700	0.9000	9	2.6000	1.4500	0.8538	1.0548
3	2.0000	0.8900	0.9000	9	2.6000	1.4500	0.6994	0.7450
3	2.0000	0.9100	0.9000	9	2.6000	1.4500	0.4567	0.6868
3	2.0000	0.8700	0.9000	9	2.6000	1.5350	0.8538	1.2480
3	2.0000	0.8900	0.9000	9	2.6000	1.5350	0.6994	1.0508
3	2.0000	0.9100	0.9000	9	2.6000	1.5350	0.4567	0.8287
3	2.0000	0.8700	0.9000	9	4.0000	1.4500	0.5216	0.3534
3	2.0000	0.8900	0.9000	9	4.0000	1.4500	0.2840	0.5608
3	2.0000	0.8700	0.9000	9	4.0000	1.5350	0.5216	0.8125
3	2.0000	0.8900	0.9000	9	4.0000	1.5350	0.2840	0.8249
3	2.0000	0.8700	0.9250	3	2.6000	1.4500	1.6749	1.5494
3	2.0000	0.8900	0.9250	3	2.6000	1.4500	1.5205	1.6060
3	2.0000	0.9100	0.9250	3	2.6000	1.4500	1.2778	1.6892
3	2.0000	0.8700	0.9250	3	2.6000	1.5350	1.6749	1.3222
3	2.0000	0.8900	0.9250	3	2.6000	1.5350	1.5205	1.3513
3	2.0000	0.9100	0.9250	3	2.6000	1.5350	1.2778	1.2391
3	2.0000	0.8700	0.9250	3	4.0000	1.4500	1.3427	1.3209
3	2.0000	0.8900	0.9250	3	4.0000	1.4500	1.1051	1.4954
3	2.0000	0.8700	0.9250	3	4.0000	1.5350	1.3427	0.9929
3	2.0000	0.8900	0.9250	3	4.0000	1.5350	1.1051	1.1083
3	2.0000	0.8700	0.9250	6	2.6000	1.4500	1.5006	1.5558
3	2.0000	0.8900	0.9250	6	2.6000	1.4500	1.3462	1.4504
3	2.0000	0.9100	0.9250	6	2.6000	1.4500	1.1035	1.3922
3	2.0000	0.8700	0.9250	6	2.6000	1.5350	1.5006	1.4268
3	2.0000	0.8900	0.9250	6	2.6000	1.5350	1.3462	1.4230
3	2.0000	0.9100	0.9250	6	2.6000	1.5350	1.1035	1.2590
3	2.0000	0.8700	0.9250	6	4.0000	1.4500	1.1684	1.1118

Table E.6 contd.

3	2.0000	0.8900	0.9250	6	4.0000	1.4500	0.9308	1.2229
3	2.0000	0.8700	0.9250	6	4.0000	1.5350	1.1684	1.1100
3	2.0000	0.8900	0.9250	6	4.0000	1.5350	0.9308	1.0992
3	2.0000	0.8700	0.9250	9	2.6000	1.4500	1.3263	1.4371
3	2.0000	0.8900	0.9250	9	2.6000	1.4500	1.1719	1.1734
3	2.0000	0.9100	0.9250	9	2.6000	1.4500	0.9292	0.9919
3	2.0000	0.8700	0.9250	9	2.6000	1.5350	1.3263	1.4484
3	2.0000	0.8900	0.9250	9	2.6000	1.5350	1.1719	1.3856
3	2.0000	0.9100	0.9250	9	2.6000	1.5350	0.9292	1.1704
3	2.0000	0.8700	0.9250	9	4.0000	1.4500	0.9941	0.9406
3	2.0000	0.8900	0.9250	9	4.0000	1.4500	0.7565	0.9945
3	2.0000	0.8700	0.9250	9	4.0000	1.5350	0.9941	1.2350
3	2.0000	0.8900	0.9250	9	4.0000	1.5350	0.7565	1.0885
3	2.0000	0.8700	0.9500	3	2.6000	1.4500	2.1474	2.0260
3	2.0000	0.8900	0.9500	3	2.6000	1.4500	1.9930	1.9487
3	2.0000	0.9100	0.9500	3	2.6000	1.4500	1.7503	1.8099
3	2.0000	0.8700	0.9500	3	2.6000	1.5350	2.1474	1.7772
3	2.0000	0.8900	0.9500	3	2.6000	1.5350	1.9930	1.7746
3	2.0000	0.9100	0.9500	3	2.6000	1.5350	1.7503	1.4927
3	2.0000	0.8700	0.9500	3	4.0000	1.4500	1.8152	1.6699
3	2.0000	0.8900	0.9500	3	4.0000	1.4500	1.5776	1.7160
3	2.0000	0.8700	0.9500	3	4.0000	1.5350	1.8152	1.1977
3	2.0000	0.8900	0.9500	3	4.0000	1.5350	1.5776	1.0644
3	2.0000	0.8700	0.9500	6	2.6000	1.4500	1.9731	1.8432
3	2.0000	0.8900	0.9500	6	2.6000	1.4500	1.8187	1.7062
3	2.0000	0.9100	0.9500	6	2.6000	1.4500	1.5760	1.4854
3	2.0000	0.8700	0.9500	6	2.6000	1.5350	1.9731	1.6509
3	2.0000	0.8900	0.9500	6	2.6000	1.5350	1.8187	1.6990
3	2.0000	0.9100	0.9500	6	2.6000	1.5350	1.5760	1.4700
3	2.0000	0.8700	0.9500	6	4.0000	1.4500	1.6409	1.5029
3	2.0000	0.8900	0.9500	6	4.0000	1.4500	1.4033	1.4640
3	2.0000	0.8700	0.9500	6	4.0000	1.5350	1.6409	1.3404
3	2.0000	0.8900	0.9500	6	4.0000	1.5350	1.4033	1.1079
3	2.0000	0.8700	0.9500	9	2.6000	1.4500	1.7988	1.5550
3	2.0000	0.8900	0.9500	9	2.6000	1.4500	1.6444	1.3822
3	2.0000	0.9100	0.9500	9	2.6000	1.4500	1.4017	1.1207
3	2.0000	0.8700	0.9500	9	2.6000	1.5350	1.7988	1.4850
3	2.0000	0.8900	0.9500	9	2.6000	1.5350	1.6444	1.5741
3	2.0000	0.9100	0.9500	9	2.6000	1.5350	1.4017	1.4191
3	2.0000	0.8700	0.9500	9	4.0000	1.4500	1.4666	1.3152
3	2.0000	0.8900	0.9500	9	4.0000	1.4500	1.2290	1.2193
3	2.0000	0.8700	0.9500	9	4.0000	1.5350	1.4666	1.4767
3	2.0000	0.8900	0.9500	9	4.0000	1.5350	1.2290	1.1750

A.1.7 Learning

ANN learn patterns of activation hence learning can be equated to determining the proper values of connection strength that allows all the nodes to achieve the correct state of activation for given pattern of inputs. Once the pattern of activation is established, the resulting outputs let the network classify an input pattern. The adaptive nature of ANN allows the weights to be learned by experience, thus producing self-organizing system. [*Himmelbalu, et al(1988)*]

A robust learning heuristic for multi layered feed forward ANN called generalized delta rule (GDR) or back propagation learning rule used successfully in training the NN for wide applications. In the feed forward nets inputs feed through hidden layers to an output layer. Each neuron forms a weighted sum of inputs from previous layers to which it is connected, adds a threshold value and produces a non-linear function of this sum as its output value. The output values serves as the input to the next layer to which the neuron is connected, and the process is repeated until output values are obtained for the neurons in the output layer. Thus each neuron performs

$$y_j^p = f(\sum_i w_{ij}.x_j^p - \theta_j) \qquad \text{(A.3)}$$

where w_{ij} is the weight from neuron i to neuron j, w_{ij} can be positive or negative real number and θ_j is the threshold of the jth neuron, p means the pth pattern. The *f*(x) is non-linear function of activation that is often chosen to be of a sigmoidal form.

If d_j^p are the desired outputs and y_i^p are the outputs obtained from the output layer for the pth pattern. Neural nets are trained by minimizing the error function

$$E = \sum_{p=1}\sum_{i=1}(d_i^p - y_i^p)^2 \qquad \text{(A.4)}$$

where i indexes the number of neurons in the output layer, and p means the pth input pattern of the training set is presented on the input layer.

Minimizing the sum of square of errors is not always the best way of training a NN but for some applications it suffices. Back propagation by GDR, a kind of gradient descent method is one popular method. The gradient descent is described by the following equations. The commonly used steepest descents procedure in minimizing E is to change w_{ij} and θ_j by Δw_{ij} and $\Delta\theta_j$ where

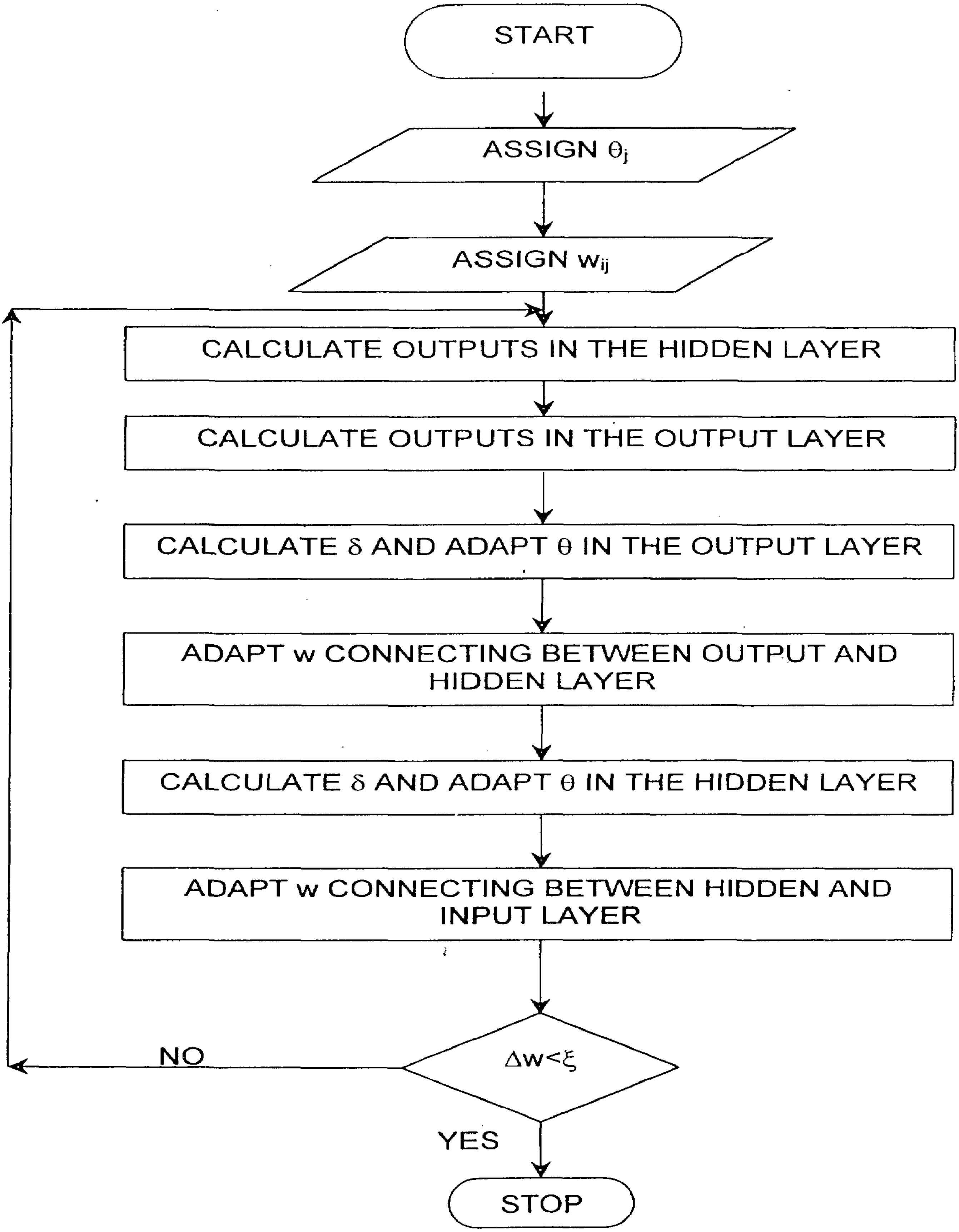

START
ASSIGN θ_j
ASSIGN w_{ij}
CALCULATE OUTPUTS IN THE HIDDEN LAYER
CALCULATE OUTPUTS IN THE OUTPUT LAYER
CALCULATE δ AND ADAPT θ IN THE OUTPUT LAYER
ADAPT w CONNECTING BETWEEN OUTPUT AND HIDDEN LAYER
CALCULATE δ AND ADAPT θ IN THE HIDDEN LAYER
ADAPT w CONNECTING BETWEEN HIDDEN AND INPUT LAYER
NO
$\Delta w<\xi$
YES
STOP

Figure. A.7 Back Propagation Algorithm

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}}\eta \qquad (A.5)$$

$$\Delta \theta_j = -\frac{\partial E}{\partial \theta_j}\eta \qquad (A.6)$$

where η is the learning rate

After simplification Δw_{ij} and $\Delta\theta_j$ can be expressed as

$$\Delta w_{ij}^p = \eta \sum_p \delta_j^p y_j^p + \alpha \Delta w_{ij}^{p-1} \qquad (A.7)$$

$$\Delta \theta_j = \eta \sum_p \delta_j^p \qquad (A.8)$$

where

$$\delta_j^p = (d_j^p - y_j^p) y_j^p (1 - y_j^p) \qquad (A.9)$$

if the jth neuron is in the output layer and

$$\delta_j^p = y_j^p (1 - y_j^p) \sum_k \delta_k^p w_{jk} \qquad (A.10)$$

if jth neuron is the hidden layer and k are overall neurons in the layer above neuron j.

A.1.7.1 Training Algorithm

To begin learning, the network is initialised with small random weights on each branch. A training example is selected randomly, and the input vector x^p propagated through the network to get predicted output y^p. η is called the learning rate which is equivalent to step size and α which acts as momentum term to keep the direction of descent from changing too rapidly from step to step. On the first step the steepest descent direction is used(α=0)(fig.A.7):

```
Step (1) : Assign all neuron offsets (thresholds) to small random values : $\theta_j$
Step (2) : Assign all weights to small random values : $w_{ij}$
Step (3) : Repeat
           for p=1 to TP (TP is the total number of training patterns)
           for j=1 to $n_2$ ($n_2$ is the number of neurons in the hidden layer)
           calculate neuron outputs in hidden layer $y_j$
           end for
```

```
        for k=1 to $n_3$ ($n_3$ is the number of neurons in the output layer)
        calculate neuron outputs in the output layer $y_k$
          calculate $\delta_k^p$
          $\theta_k^{(p)} = \theta_k^{(p-1)} + \Delta\theta_k^{(p)}$
        end for
        for j=1 to $n_2$ and k=1 to $n_3$
        $w_{jk}^{(p)} = w_{jk}^{(p-1)} + \Delta w_{jk}^{(p)}$
        end for
        for j=1 to $n_2$
        calculate $\delta_j^p$
        $\theta_j^{(p)} = \theta_j^{(p-1)} + \Delta\theta_j^{(p)}$
        end for
for i= 1 to $n_1$ and j= 1 to $n_2$
($n_1$ is number of neurons in the input layer )
$w_{ij}^{(p)} = w_{ij}^{(p-1)} + \Delta w_{ij}^{(p)}$
end for
end for
until $\Delta w < \xi$ ($\xi$ is the convergence criterion)
```

A.1.7.2 Properties of back propagation algorithm

1. The use of a grad ent algorithm to train its weights makes it slow to converge. [*Zurada*
2. It may be trapped at local minima. [*Himmelbalu, et al(1988)*]
3. It can be sensitive to user selectable parameter. [*Himmelbalu, et al(1988)*]
4. Being a feed-forward algorithm, the advantage of this method is that each adjustment step is computed quickly without presentation of all the patterns and without finding an overall direction of the descent for the training cycle. [*Himmelbalu, et al(1988)*, [*Zurada (1996)*]]