



# Application of Neural Network Models for Mathematical Programming Problems: A State of Art Review

Kailash Lachhwani<sup>1</sup>

Received: 18 May 2018 / Accepted: 10 December 2018 / Published online: 2 January 2019  
© CIMNE, Barcelona, Spain 2019

## Abstract

Artificial neural networks or neural networks (NN) are new computational models based on the working of biological neurons of human body. A NN model consists of an interactive system through which external or internal information flows. Nowadays, NN models are being used to deal with complex real problems. On the other hand, mathematical programming problems (MPPs) are a particular class of optimization problems with mathematical structure of objective function(s) and set of constraints. Use of NN models in solving MPPs is a complex area of research and researchers have tried to contribute to apply NN models on different mathematical programming problems. This paper describes classification of MPPs, different neural network models and the detailed literature review on application of NN models for solving different MPPs along with comprehensive analysis on references. Some new research issues and scopes are also discussed on the use of different NN models on MPPs. This paper aims to present the state of art literature review on the use of NNs for solving MPPs with constructive analysis to elaborate future research scope and new directions in this area for future researchers.

## 1 Introduction

Mathematical programming problems (MPPs) consisting of programming problems with general mathematical structure of objective function(s) and set of constraints in different formats so as to find optimal solution (or satisfactory solution) of the problems. In literature, the elementary and simple mathematical programming problem known as linear programming problem (LPP) is defined in mathematical format as:

$$\begin{aligned} & \text{Max./Min. } Z = C^T X \\ & \text{Subject to, } AX(\leq, =, \geq)b \\ & \text{and } X \geq 0 \end{aligned} \quad (1)$$

where  $X$  is the set of variables,  $C$  is the cost matrix,  $A$  being coefficient matrix and  $b$  is the requirement matrix of compatible orders in terminology. Linear programming problems (LPPs) are further extended to different types of programming problems with modifications in structure of objective

function and constraints. In general, MPPs are classified as linear programming problem (LPP), non-linear programming problems (NLPPs), multiobjective programming problem (MOPP), multi-level programming problems (MLPPs) etc. which are further classified and separated according to the mathematical structure of the problem given as:

(A) Linear programming problems with single objective function are classified as given in Lachhwani and Dwivedi [35]:

- “Linear fractional programming problem (LFPP).
- Linear plus linear fractional programming problem (L + LFPP).
- Non-linear programming problem (NLPP).
- Quadratic programming problem (QPP).
- Quadratic fractional programming problem (QFPP)” etc.

(B) Programming problems with more than one conflicting objective function are known as multi-objective programming problems and some of multi-objective programming problems as given in Lachhwani and Dwivedi [35] are:

✉ Kailash Lachhwani  
kailashlachhwani@yahoo.com

<sup>1</sup> Department of Applied Science, National Institute of Technical Teachers Training and Research, Chandigarh 160 019, India

- “Multiobjective linear programming problem (MOLPP).
- Multiobjective linear fractional programming problem (MOLFP).
- Multiobjective linear plus fractional programming problem (MOL + FPP).
- Multiobjective non-linear programming problem (MONLPP).
- Multiobjective quadratic programming problem (MOQPP).
- Multiobjective quadratic fractional programming problem (MOQFPP)” etc.

(C) Programming problems with hierarchical structure with more than one levels are known as multi-level programming problems (MLPPs) which are further extended to more complex hierarchical problems as given in Lachhwani and Dwivedi [35]:

- “Multi-level nonlinear programming problem (ML-NLPP).
- Multi-level linear programming problem (ML-LPP).
- Multi-level linear fractional programming problem (ML-LFPP).
- Multi-level linear plus linear fractional programming problem (ML-L + LFPP).
- Multi-level quadratic programming problem (ML-QPP).
- Multi-level quadratic fractional programming problem (ML-QFPP)” etc.

(D) Hierarchical programming problems with two level structure are considered as Bi-level programming problems (BLPPs) which also have extension problems as given in Lachhwani and Dwivedi [35]:

- “Bi-level nonlinear programming problem (ML-NLPP).
- Bi-level linear programming problem (ML-LPP).
- Bi-level linear fractional programming problem (ML-LFPP).

- Bi-level linear plus linear fractional programming problem (ML-L + LFPP).
- Bi-level quadratic programming problem (ML-QPP).
- Bi-level quadratic fractional programming problems(ML-QFPP)” etc.

This list is not limited and there are many new and extended MPPs available in literature. This classification is well described in figure format by Fig. 1. The detailed literature reviews and taxonomy on MOPPs, BLPPs and MLPPs are respectively given in Bhati et al. [5], Lachhwani and Dwivedi [35].

## 2 Artificial Neural Network (Neural Network)

The fundamentals of NN are based on working of the biological neurons of the brain. Artificial neural networks (ANN) are massively connected networks of computational “neurons” and represent parallel distributed learning structures. A typical ANN composed of a set of parallel and distributed computation unit called nodes or neurons. These are usually ordered into layers and interconnected with weighted signal channels called connections or synaptic weights. A typical representation of three layer NN model is given in Fig. 2. A key feature of neural networks is their ability to approximate arbitrary nonlinear functions.

The basic element of computational process of a neuron is that it takes weighted sum of their inputs from other nodes and apply to them a nonlinear function (or called activation function) before delivering the output to the next neuron.

Basic steps to be followed to design a suitable Neural network model: the following basic steps are required to be followed to design a suitable neural network model as:

- Step 1 Identifying the input and output variables of the problem to be modelled.
- Step 2 Normalize the variable in the range of 0–1 or  $-1$  to  $+1$ .

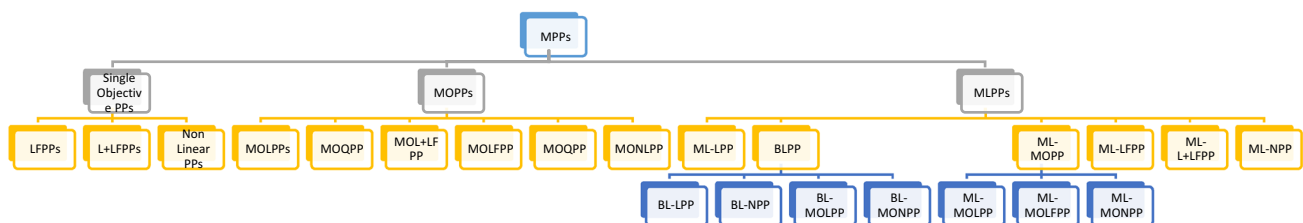
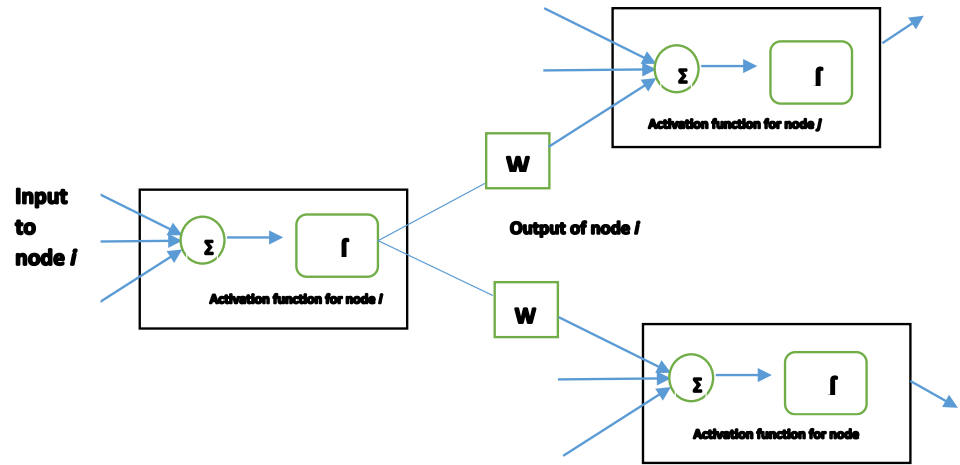


Fig. 1 Classification of MPPs

Fig. 2 Generalized NN model



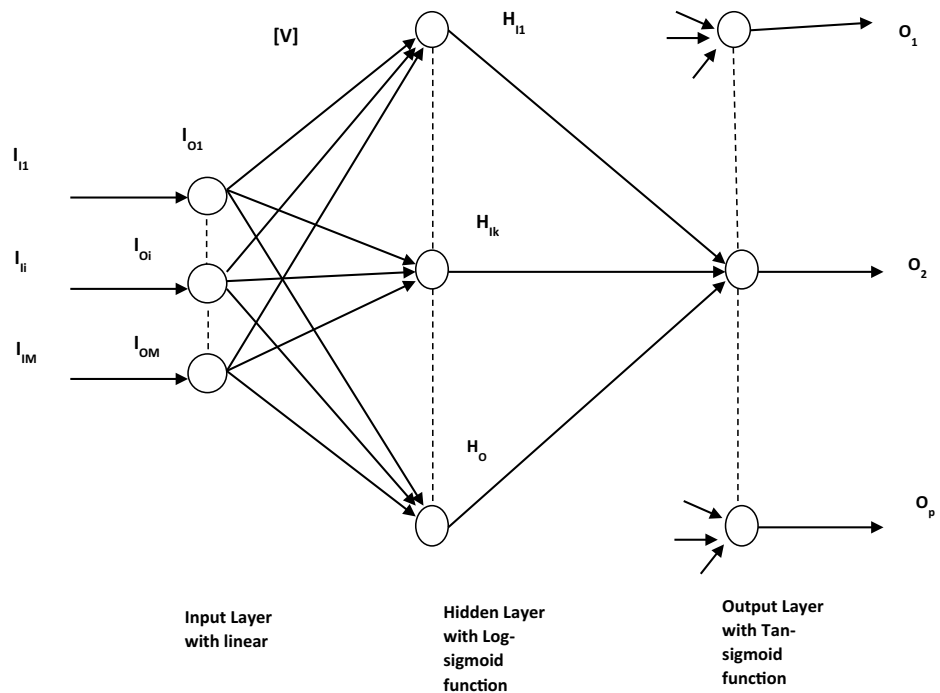
- Step 3 Initialize the number of hidden layers and number of neurons in each layer. Select the appropriate activation function for each neuron.
- Step 4 Generate the values of weights, bias values and coefficients of the activation function.
- Step 5 Update the above parameters iteratively using suitable algorithm of training.
- Step 6 Continue the iterations until the termination criteria is reached.
- Step 7 Testing of the optimized neural network.

Various types of neural networks models are in use, such as “feed forward neural networks (FFNNs), recurrent neural networks (RNNs), radial basis function neural

network (RBFNN), self organizing map neural network (SOM), Combined/hybrid neural network” and others. These types of networks are implemented based on the mathematical operations and a set of parameters required to determine the output. Some of these types of NNs are discussed in brief as:

1. Feedforward neural network (FFNNs): This neural network is one of the simplest NN form and widely used NN in which information is passed into one direction, that is, starting from the input layer towards the output layer through the hidden layers. So, it does not form any cycle or loop. Figure 3 shows a multi-layered feed-forward neural network.

Fig. 3 Multi-layered feed-forward neural network



2. Radial basis function neural network (RBFNNs): it is a special type of neural network which consists of a layer of some input data nodes, generally one hidden layer of a few neurons with radial basis transfer function and an output layer composed of some neurons with linear transfer function usually. This is the reason that RBFNNs are also called two-layered neural networks. Figure 4 shows a radial basis function neural network.
3. Self-organizing map (SOM) neural network: it is a sequential clustering algorithm., which produces the self-organizing feature maps similar to those present in our brain. It is a modified version of neural network working based on unsupervised and competitive learning. Figure 5 shows the systematic view of a self-organizing map neural network.
4. Recurrent neural network (RNN): a recurrent neural network (RNN) has both feed-forward as well as feed-back connections and as a result of which, information can be processed from the input layer to output layer and vice versa. Thus, forms a cycle or loop. It requires a less number of neurons and consequently, becomes less expensive computationally compared to a feed-forward network. Figure 6 shows sample of RNN.
5. Combined/Hybrid neural network: these NNs are combination of two or more NNs models in tuning of connecting weights or bias values or other parameters of NNs. Popularly known hybrid NNs are genetic–neural systems (GNSs), neuro-fuzzy systems (NFSs), fuzzy-neural networks etc. Likewise, genetic algorithms (GAs) have been combined with neural networks (NNs) to develop the combined—genetic-neuro systems (GNSs). When

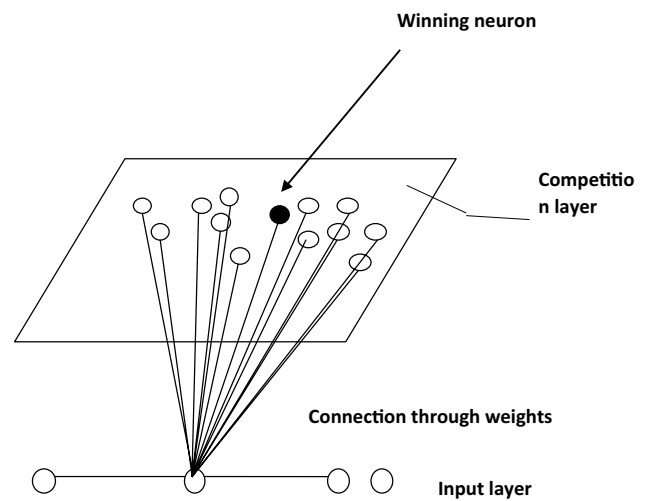


Fig. 5 Self-organizing neural network

neural networks (NNs) have been combined with fuzzy logic (FL) techniques in different ways, then it formulated neuro-fuzzy systems (NFSs). In another approach, when the neurons of an NN have been designed using the concept of fuzzy set theory and the developed network is known as fuzzy-neural networks (FNNs).

Besides this, there are other NN models like stochastic neural network (SNN), modular neural network (MNN), CPNN etc. NN has some specific characteristics that (1) the information can be processed in both the directions and (2) interactive process of NNs. Using these characteristics,

Fig. 4 Radial basis function neural network

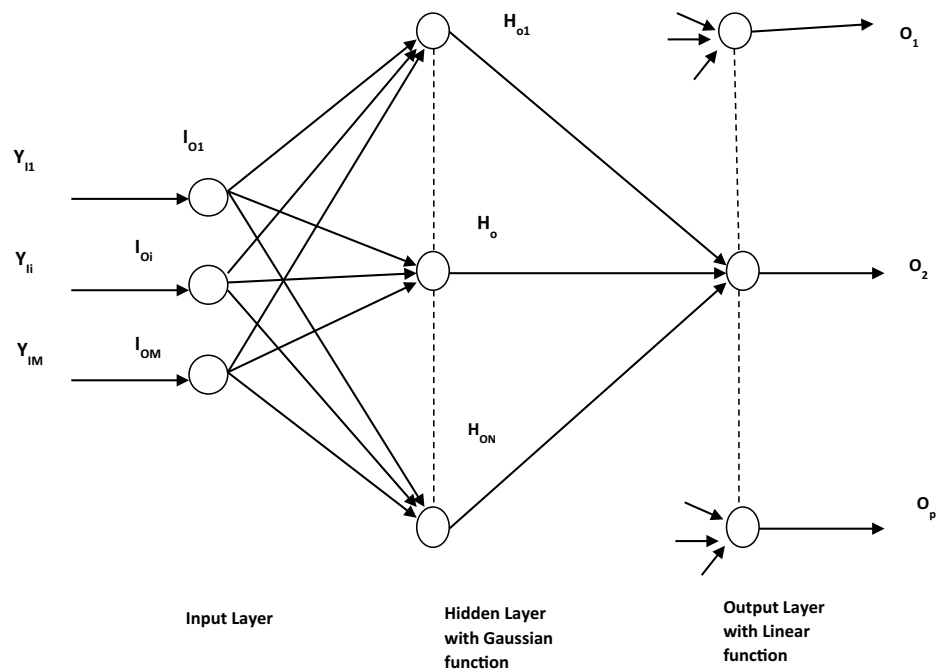
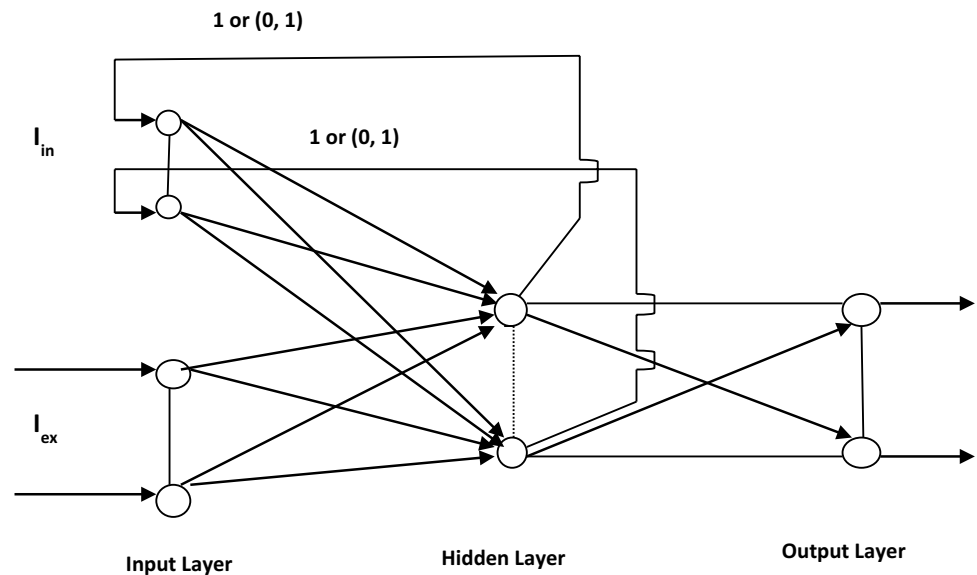


Fig. 6 Recurrent neural network



several authors have contributed solution techniques for MPPs. In the next section, we have discussion on the literature review on use of NN models for MPPs along with constructive analysis on these references.

### 3 State of Art: Review on Use of NN Models for Solving MPP and Analysis

Artificial neural network models play important role to solve complex computational problems and in recent years, researchers have identified this tool as important tool for solving complex MPPs. But still, there is lack of proper literature review analysis on the use of NN models for MPPs. Due to this, researchers do not get proper research directions. This motivates us to present the state of art literature review and analysis on this area for the benefits of researchers. In literature, this is the first attempt to present such state of art review on applications of neural networks for mathematical programming problems. This review has carried out with research articles/papers from reputed journals of SCOPUS, IEEE, Scienccdirect, Google etc. databases for presentation of this work.

Now, we present the literature review on use of NNs for MPPs in year-wise sequence along with conclusive analysis as: starting from 1986, Xia [79] proposed a new and unique NN without parameters for the exact solutions of linear programming problems (LPPs). In next year 1987, Greenberg [25] suggested addition of an interactive system for the analysis of LP models and its solutions. Rodriquez-Vazquez et al. [58] proposed an online NN circuit solving of LPPs.

Greenberg [26] suggested some preliminary results on the application of neural networks to the design of a system for solving mathematical programming system which

are mainly: model completion, discourse and rulebase generation. Wang and Malakooti [72] presented a FFNN for solving discrete multiple criteria decision making (MCDM) problems under uncertainty. In decade of 1990s, Zhang and Constantinides [87] proposed NN based on the Lagrange multipliers theory for the non-linear programming problems which proved to provide solutions satisfying the conditions of optimality. Maa and Shanblatt [47] analysed NNs for LPPs and QPPs. Burke and Ignizio [6] discussed the use of NNs in solving OR analysis problems including LPPs. Wang and Chankong [71] explored potential role of recurrent NNs for solving basic LPPs. Wang [68] proposed “a recurrent neural network (RNN) for solving quadratic programming problems with equality constraints”. Wang [69] presented “a recurrent NN with a time-varying threshold vector for LPPs”. Malakooti and Zhou [48] presented an adaptive FFNN to solve multiple criteria decision making (MCDM) problems. Wang [70] proposed “a recurrent neural network (RNN) for solving convex programming problems”. Gee and Prager [20] elaborated limitations of NNs in context of solving travelling salesman problems (TSPs) and also suggested solution for the failure of a network even when 10-city problem which is significantly high. Xia and Wang [78] presented a new NN for solving LPPs with bounded variables. Zak et al. [86] studied three classes of NN models for solving LPPs and investigated the dependent characteristics namely model complexity, complexity of individual neurons and accuracy of solutions. Wu et al. [74] proposed two classes of high performance NNs for LPPs and QPPs. Xia [79] presented a new globally convergent NN for solving linear and quadratic programming problems. Cichocki et al. [10] proposed and analysed a new class of neural network models with energy function for solving LPPs. Xia [76] presented a new NN for solving general LPPs and its

dual problems simultaneously. Aourid and Kaminska [2] presented “a Boolean neural network (BNN) for the 0–1 LPP under inequality constraints by using relation between concave programming problems and integer programming problems”.

Sun et al. [62] proposed a new interactive FFNN procedure for solving MOPPs. Gong et al. [24] proposed ANN approach based on Lagrangian multiplier method (Lagrangian ANN) for the solution of convex programming problems with linear constraints. Xia [75] presented a globally convergent neural network for solving extended linear programming problems. Kennedy and Chua [34] invented an A/D converter signal decision circuit based NN to solve general NLPPs. Gen et al. [21] introduced a new NN technique to solve fuzzy MOPPs. Walsh et al. [67] proposed an augmented Hopfield network to solve mixed integer programming problems. Liao and Hou-Duo [40] proposed an artificial neural network for linear complementarity problem. The proposed new NN based is based on conversion of linear complementarity problem into the unconstrained problem which can easily be implemented on a circuit. Chong et al. [9] analysed a class of NNs that solve LPPs. In 2000, Nguyen [57] presented a new recurrent NN for solving LPPs. Li and Da [40] discussed the neural representation of linear programming (LP) and fuzzy linear programming (FP). Sun et al. [63] proposed a new interactive multiple objective programming procedure with the combination of the interactive weighted and FFANN procedure. Tao et al. [65] proposed a continuous NN for NLPPs with high performance. Tao et al. [64] proposed a simplified high performance NN for solving quadratic programming problems. Meida-Casermeiro et al. [53] proposed a simple discrete multivalued Hopfield NN for the solution of travelling salesman problems (TSPs). Leung et al. [38] constructed “a new gradient-based NN with the duality theory, optimization theory etc. to provide solution of linear and quadratic programming problems”. Dillona and O’Malley [11] proposed “the Hopfield neural network to solve mixed integer non linear programming problems”. Chen et al. [8] proposed “a NN model for solving convex NLPPs”. Zhang and Wang [89] proposed “a recurrent neural network (RNN) for solving the strictly convex QPPs”. Leung et al. [37] proposed “high performance feedback NN for solving convex NLPPs based on successive approximation”. Shih et al. [62] studied the dynamic behaviour of artificial neural networks (ANNs) for the solution of multiobjective programming problems (MOPPs) and multilevel programming problems (MLPPs). Forti et al. [17] introduced a generalized circuit based NN for solving in real time NLPPs. Gao [19] presented a NN for solving the convex NLPP by the projection method. This NN is based on conversion of a convex NLPP into a variational inequality problem. Effati and Baymain [12] presented a new recurrent NN for solving convex NLPPs. Cao and Liu [7]

introduced fuzzy decentralized decision-making problem, fuzzy multilevel programming and chance-constrained multilevel programming problem in context of NN. Malek and Yari [50] represented two new methods for the solution of LPPs with condition to minimize energy function of the corresponding neural network.

Zhang [88] presented “a primal–dual NN for the online solution based on linear variational inequalities (LVI)”. Xia and Wang [77] proposed “a recurrent neural network for solving convex NLPPs with linear constraints”. The proposed NN has simple structure and lower complexities in implementation in comparison of other NNs for solving such problems. Ghasabi-Oskoei and Mahdavi-Amiri [23] presented a high-performance and efficiently simplified NN for solving general LPPs and QPPs. Effati and Nazemi [15] considered “two recurrent neural network model for solving linear and quadratic programming problems”. Yang and Cao [83] proposed the delayed projection neural network for solving convex QPPs. Liu and Wang [42] proposed a new recurrent NN for QPPs (this NN is also known as simplified dual NN) and also discussed its design and analysis on this NN. Lan et al. [36] proposed “a combined neural network and tabu search hybrid algorithm for solving the bilevel programming problem (BLPP)”. Effati et al. [13] proved that “solution of convex programming problems is equivalent with solution of projection formulation problem and then they introduced NN models for projection formulation problem and analysis its convergence”. Ghasabi-Oskoei et al. [23] proposed “a novel NN for linear and quadratic programming problem with recurrent ANN approach”.

Malek and Alipour [49] proposed “a recurrent neural network for solving LPPs and QPPs”. The main advantage of this NN is that there is no need of parameters setting. Lv Yibing et al. [46] discussed a NN model for solving a NP-hard nonlinear bilevel programming problem (BL-NLPP). This NN is also proved to be stable, convergent and capable of generating approximal optimal solution to this complex problem.

Xiaolin and Wang [77] presented “a novel recurrent neural network (RNN) for convex QPPs with linear constraints”. Liu and Wang [41] proposed “a one-layer recurrent neural network with a discontinuous activation function for LPP”. Wen et al. [73] proposed “the hopfield neural network (HNN) for solving MPPs”. The major advantage of proposed HNN is its structure can be easily applied on an electronic circuit. Xiaolin [29] considered solving “the extended linear quadratic programming (ELQP) problems with general polyhedral sets by using recurrent neural networks”. Hu and Zhang [32] proposed “a new recurrent neural network for solving convex QPPs”. Yang and Du [82] proposed “a novel neural network algorithm to solve QPP with linear constraints based on Fibonacci method”. Tiesong et al. [29] proposed “a novel neural network for BLPP which is also



proved to be stable, convergent and capable of generating optimal solution to the problem". Lv Y et al. [45] presented "a neural network for solving a convex quadratic bilevel programming problem (BLQPP)". Gao and Liao [18] presented "a new neural network for solving LPP and QPP".

Arvind Babu and Palaniappan [4] proposed "ANN based hybrid algorithm which trains the constrains and parameters before applying the formal LPP method for solution of LPPs". Nasira et al. [55] presented "back propagation for training and simulation with the new inputs to implement NN for integer linear programming problems (ILPPs)". Sahu and Avanish Kumar [59] proposed "circuit based simple NN for solution of LPPs". Lv et al. [44] presented "a neural network approach for solving mathematical programs with equilibrium constraints (MPEC)". Yang and Gao [85] presented "a new NN for solving convex NLPPs with linear constrains". Alipour [54] proposed "a recurrent NN for convex NLPPs subject to linear equality and inequality constraints". Yaakob and Watada [80] demonstrated "a double-layered hybrid NN method to solve mixed integer quadratic bilevel programming problems".

Yaakob and Watada [81] proposed "a hybrid neural network approach to solve mixed integer quadratic bilevel programming problems". In this NN, the combination of a genetic algorithm (GA) and a meta-controlled Boltzmann machine (BM) is formulated as a hybrid neural network approach to solve BLPPs. Effati and Ranjbar [16] presented "a new NN for solving QPPs". Vahabi and Ghasabi-Oskoei [66] proposed "a high-performance feedback NN model for solving convex NLPPs with hybrid constraints in real time by means of the projection method". Selvaraj and Pandian [60] proposed "a neural network approach for solving fuzzy linear programming problems in which fuzzy concepts are not used". Nazemi [1] proposed "a capable NN for solving strictly convex quadratic programming (SCQP) problems with general linear constraints". Yang et al. [84] proposed "a new NN for solving QPPs with equality and inequality constraints". He et al. [27] proposed "a NN to solve convex quadratic bilevel programming problems (CQBPPs), based on successive approximation". He et al. [28] proposed "a recurrent neural network (NN) modelled for solving BLPPs based on the method of penalty functions".

Effati et al. [14] discussed the application of projection neural network for solving bilinear programming problems (BLPs). Huang et al. [33] proposed "a neural network model for solving convex quadratic programming (CQP) problems using Karush–Kuhn–Tucker (KKT) points of the CQP problem". Jin and Li [44] proposed "Zeroing neural network (ZNN) for the solution of QPPs subject to equality and inequality constraints". Arjmandzadeh et al. [3] presented "a neural network model for solving random interval linear programming problems which proved to be stable and globally convergent to exact solution to the problem". Ranjbar et al.

[57] presented an "ANN to solve the quadratic zero–one programming problems under linear constraints". Mansoori et al. [52] proposed "a representation of a recurrent neural network to solve QPPs with fuzzy parameters (FQP)".

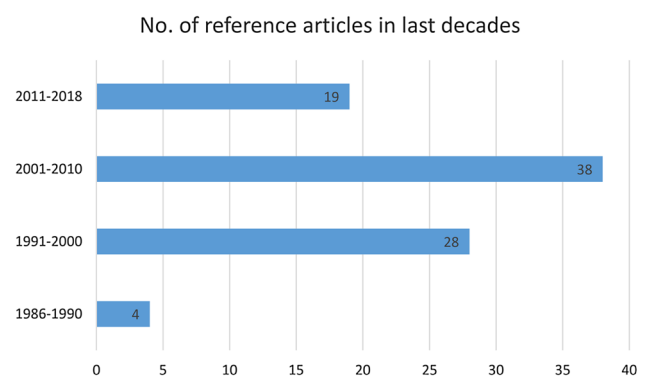
In a view of analysis, it can be easily found that since 1986, researchers have started using NNs models in solving different mathematical programming problems and this is continuously increasing with increasing number of article publications in last four decades. From graph (Graph 1) it is shown that during 2001–2010, highest number of research articles has been published in this research area.

From Table 1, it is shown that researchers have contributed more in the areas of LPPs and QPP while less in the areas of MOPPs, BLPPs, Travelling salesman problems, MLPPs etc. in terms of research publications. Further there are many complex programming problems like ML-MOPPs, ML-MOLFPPs, ML-MOQPPs, ML-MOQFPPs and other extension problems on which use of NN models has not been initiated till this date. The application of NN models may suggest some satisfactory solutions to these complex problems with less computational efforts.

It is evident from Table 2 that RNNs, FFNNs, HFNNs have been applied to solve MPPs in sufficient numbers. However, some other robust neural networks techniques available in literature like stochastic neural network (SNNs), Modular neural networks (MNNs), radial basis function neural network (RBFNNs), self-organizing map (SOM), counter-propagation neural networks (CPNN) etc. have not been applied to solve simple and complex mathematical programming problems.

## 4 Research Issues

It can be observed from literature review, Tables 1, 2 and Graph 1 that NN models play an important role in solving simple to complex mathematical programming problems—MOPPs, BLPPs etc. But still there are some important



**Graph 1** No. of research publications on NN models for MPPs

**Table 1** Description of neural network techniques on different mathematical programming problems

S. no.	Programming problem	Neural network techniques by researchers	No. of references
1.	Linear programming problems (LPPs)	Aourid and Kaminska [2], Arvind and Palaniappan [4], Chong et al. [9], Cichocki et al. [10], Effati and Nazemi [15], Gao and Liao [18], Ghasabi-Oskoei et al. [22], Ghasabi-Oskoei et al. [23], Greenberg [25], Li and Da [39], Maa and Shanblatt [47], Marta et al. [52], Nguyen [56], Rodriguez-Vazquez et al. [58], Sahu and Kumar [59], Wang and Chankong [71], Wu et al. [74], Xia [75], Xia [76], Xia and Wang [77], Zak et al. [86], Wang [69], Leung et al. [38], Liao and Hou-Duo [40], Liu and Wang [41], Malek and Alipur [49], Malek and Yari [50], Nasira et al. [55] and Zhang [88]	29
2.	Non linear programming problems (NLPPs) (excluding quadratic programming problems)	Chen et al. [8], Dillona and O'Malley [11], Effati and Baymain [12], Forti et al. [17], Gao [19], Kennedy and Chua [34], Malek and Alipur [49], Tao et al. [65], Vahabi and Ghasabi-Oskoei [66], Xia and Wang [77], Yang and Gao [85], Effati et al. [13], Gong et al. [24], Leung et al. [37], Lv et al. [46] and Wang [70]	16
3.	Quadratic programming problems (QPPs)	Nazemi [1], Arjmandzadeh et al. [3], Effati and Nazemi [15], Effati and Ranjbar [16], Gao and Liao [18], Ghasabi-Oskoei and Mahdavi-Ahmadi [22], Ghasabi-Oskoei et al. [23], Hu [30], Hu and Wang [31], Hu and Zhang [32], Huang et al. [33], Leung et al. [38], Liu and Wang [41], Jin and Li [43], Ly et al. [45], Maa and Shanblatt [47], Malek and Alipur [49], Mansoori et al. [51], Ranjbar et al. [57], Tao et al. [64], Wang [68], Wu et al. [74], Xia [79], Yaakob and Watada [80], Yang and Du [82], Yang and Cao [83], Yang et al. [84], Zhang and Constantinides [87], Zhang [88] and Zhang and Wang [89]	30
4.	Bi-level programming problems (BLPPs)	Effati et al. [14], He et al. [27], He [28], Hu et al. [29], Lan et al. [36], Lv et al. [45], Lv et al. [46] and Yaakob and Watada [80, 81]	09
5.	Multiobjective programming problems (MOPPs), fuzzy multiobjective programming problems, Multi criteria decision making	Malakooti and Zhou [48], Shih et al. [61], Sun et al. [62], Sun et al. [63], Wang and Malakooti [72], Gen et al. [21] and Selvaraj and Pandian [60]	07
6.	Multi-level programming problems (MLPPs)	Cao and Liu [7] and Shih et al. [61]	02
7.	Integer programming problems (IPPs)	Dillona and O'Malley MJ [11], Nasira et al. [55], Walsh et al. [67] and Yaakob and Watada [80]	04
8.	Travelling salesman problems (TSPs)	Gee and Prager [20] and Meida-Casermeiro et al. [53]	02
9.	General areas—operations research, mathematical programming system, reviews etc.	Burke and Ignizio [6], Greenberg [26], Lv et al. [44] and Wen et al. [73]	04

research issues and problems open for researchers which are discussed as:

- There are variety of NN models are used to solve LPPs but the question of a proper and exact NN model arises simultaneously. A more realistic and exact NN approach is needed to establish for LPPs.
- Theoretical development of optimality conditions for solving different MPPs with different neural networks is needed.
- Many complex programming problems like MON-LPPs, BL-MOPPs, BL-MOQPPs, ML-MOPPs, ML-MOLFPPs, ML-MOQPPs, ML-MOQFPPs etc. are

available in literature (by Bhati et al. [5], Lachhwani and Dwivedi [35]) on which use of NN models has not been started or have very limited contribution in it. The use of NN models in solving these problems may suggest satisfactory solutions to the problems with less computational steps in comparison of other methods. These problems are open for researchers to study the applicability of NN models and comparative analysis with other methods for these problems.

- Comparative study between NN approach and hybrid NNs in context of solving MPPs is very few in literature and required to be studied comprehensive.



**Table 2** Description of different neural network techniques on different mathematical programming problems

S. no.	Neural network approach	Mathematical programming problem	No. of references
1.	Recurrent neural networks (RNNs)	<p>“Nguyen [56] presented a new recurrent NN for solving LPPs”</p> <p>“Wang and Chankong [71] explored potential role of recurrent NNs for solving basic LPPs”</p> <p>“Wang [69] presented a recurrent NN with a time-varying threshold vector for LPPs”</p> <p>“Effati and Baymain [12] proposed a new recurrent NN for solving convex NLPPs”</p> <p>“Effati and Nazemi [15] proposed two recurrent neural network model for solving LPPs and QPPs”</p> <p>“Ghasabi-Oskoei et al. [23] presented a simple NN for linear and quadratic programming problem with recurrent ANN approach”</p> <p>“Hu X [30] considered solving the Extended linear-quadratic programming (ELQP) problems with general polyhedral sets by using RNNs”</p> <p>“Mohsen Alipour [54] proposed a recurrent NN for solving convex NLPPs subject to linear equality and inequality constraints”</p>	08
2.	Feed-forward neural networks (FFNNs)	<p>“Wang and Malakooti [72] presented a feedforward neural network (FFNN) for solving discrete multiple criteria decision making (MCDM) problems under uncertainty”</p> <p>“Malakooti and Zhou [48] presented FFNN to solve MCDMS”</p> <p>“Sun et al. [63] proposed a new interactive NN procedure for solving MOPPs”</p> <p>“Sun et al. [62] proposed a new interactive procedure for MOPPs based on FFNNs”</p> <p>“Leung et al. [37] proposed high performance feedback NN for solving convex NLPPs based on successive approximation”</p> <p>“Vahabi and Ghasabi-Oskoei [66] proposed a high-performance feedback neural network (FBNN) model for convex NLPPs with hybrid constraints by the projection method”</p> <p>“Nasira et al. [55] presented back propagation for training and simulation with the new inputs to implement NN for integer linear programming problems (ILPPs)”</p>	07
3.	Hopfield neural networks (HNNs)	<p>“Walsh et al. [67] proposed an augmented hopfield network similar to the coupled gradient network and shown to apply to solve mixed integer programming problems”</p> <p>“Meida-Casermeiro et al. [53] suggested a simple discrete multivalued Hopfield NN for the solution of travelling salesman problems (TSPs)”</p> <p>“Dillona and O’Malley [11] proposed to solve mixed integer non linear programming problems this function have used continuous neurons to represent discrete variables based on hopfield neural network approaches”</p> <p>“Shih et al. [61] studied the utilization the dynamic behaviour of ANNs for the solution of MOPPs and MLPPs”</p> <p>“Malek and Yari [50] represented two new methods for the solution of LPPs”</p> <p>“Wen et al. [73] proposed the hopfield neural network (HNN) for solving MPPs”</p> <p>“Marta et al. [52] proposed hopfield neural networks (HNNs) and affine scaling interior point methods in hybrid for linear optimization problems”</p>	07
4.	Hybrid algorithm based neural networks	<p>“Lan et al. [36] suggested a combined neural network and tabu search hybrid algorithm for solving the BLPP”</p> <p>“Marta et al. [52] suggested HNNs and affine scaling interior point methods in a hybrid for solving linear optimization problems”</p> <p>“Tao et al. [65] presented a continuous NN for solving nonlinear programming problems with high performance”</p> <p>“Cao and Liu [7] introduced fuzzy decentralized decision-making problem, fuzzy multilevel programming and chance-constrained multilevel programming”. Furthermore, fuzzy simulation, NN and genetic algorithm (GA) are combined to produce a hybrid intelligent algorithm for finding the satisfactory solutions</p> <p>“Arvind Babu and Palaniappan [4] proposed ANN based hybrid algorithm which trains the constraints and parameters before applying the formal LPP method for solution of LPPs”</p> <p>“Yaakob and Watada [80] demonstrated a double-layered hybrid NN method to solve mixed integer quadratic bilevel programming problems”</p> <p>“Yaakob and Watada [81] proposed a hybrid neural network approach to solve mixed integer quadratic bilevel programming problems”</p> <p>“Marta et al. [52] suggested HNNs and affine scaling interior point methods in a hybrid approach for solving linear optimization problems”</p>	08

**Table 2** (continued)

S. no.	Neural network approach	Mathematical programming problem	No. of references
5.	Zeroing neural networks (ZNNs)	“Jin and Li [43] discovered zeroing neural network (ZNN) for QPPs with equality and inequality constraints”	01

- Use of other NNs like stochastic neural network (SNNs), modular neural networks (MNNs), radial basis function neural network (RBFNNs), self-organizing map (SOM), counter-propagation neural networks (CPNN) etc. have not been applied to solve any MPPs. This research area is available open for researchers.

## 5 Conclusions

In this article, an effort has been made to describes classification of MPPs, different neural network models and the detailed literature review on application of NN models for solving different MPPs along with comprehensive analysis on references. As conclusions, Up to till date, research work with the use of NNs for solving MPPs is not quite satisfactory and development of new NN theories, algorithms, use of NNs in solving complex problems etc. are emerging areas available for future research. Some open problems and research issues on NNs in context of MPPs have also been discussed. This paper aims to present state of art literature review on the use of NNs for solving MPPs with constructive analysis to elaborate future research scope and new directions in this area for future researchers.

## Compliance with Ethical Standards

**Conflict of interest** Author declare that there is no conflict of interest regarding the publication of this manuscript.

## References

- Nazemi A (2014) A neural network model for solving convex quadratic programming problems with some applications. *Eng Appl Artif Intell* 32:54–62
- Aourid M, Kaminska B (1996) Minimization of the 0–1 linear programming problem under linear constraints by using neural networks synthesis and analysis. *IEEE Trans Circuits Syst I Fund Theory Appl* 43:421–425
- Arjmandzadeh Z, Safi M, Nazemi A (2017) Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neural Netw* 89:11–18
- Arvind Babu LR, Palaniappan B (2010) Artificial neural network based hybrid algorithmic structure for solving linear programming problems. *Int J Comput Electr Eng* 2(4):1793–8163
- Bhati D, Singh P, Arya R (2017) A taxonomy and review of the multi-objective fractional programming problems. *Int J Appl Comput Math* 3(3):2695–2717
- Burke LI, Ignizio JP (1992) Neural network and operations research: an overview. *Comput Oper Res* 19:179–189
- Cao J, Liu B (2005) Fuzzy multilevel programming with a hybrid intelligent algorithm. *Comput Math Appl* 49:1539–1548
- Chen KZ, Leung Y, Leung KS, Gao XB (2002) A neural network for solving nonlinear programming problems. *Neural Comput Appl* 11:103–111
- Chong EKP, Hui S, Zak SH (1999) An analysis of a class of neural networks for solving linear programming problems. *IEEE Trans Autom Control* 44:1995–2006
- Cichocki A, Unbehauen R, Weinzierl K (1996) A new neural network for solving linear programming problems. *Eur J Oper Res* 93:244–256
- Dillona JD, O’Malley MJ (2002) A Lagrangian augmented Hopfield network for mixed integer non-linear programming problems. *Neurocomputing* 42:323–330
- Effati S, Baymain M (2005) A new nonlinear neural network for solving convex nonlinear programming problems. *Appl Math Comput* 168:1370–1379
- Effati S, Ghomashi A, Nazemi AR (2007) Application of projection neural network in solving convex programming problem. *Appl Math Comput* 188:1103–1114
- Effati S, Mansoori A, Eshaghnezh M (2015) An efficient projection neural network for solving bilinear programming problems. *Neurocomputing* 168:1188–1197
- Effati S, Nazemi AR (2006) Neural network models and its application for solving linear and quadratic programming problems. *Appl Math Comput* 172:305–331
- Effati S, Ranjbar M (2011) A novel recurrent nonlinear neural network for solving quadratic programming problems. *Appl Math Model* 35(4):1688–1695
- Forti M, Nistri P, Quincampoix M (2004) Generalized neural network for nonsmooth nonlinear programming problems. *IEEE Trans Circuits Syst I* 51(9):1741–1754
- Gao X, Liao LZ (2010) A new one-layer neural network for linear and quadratic programming. *IEEE Trans Neural Netw* 21(6):918–927
- Gao XB (2004) A novel neural network for nonlinear convex programming. *IEEE Trans Neural Netw* 15(3):613–621
- Gee AH, Prager RW (1995) Limitations of neural networks for solving travelling salesman problems. *IEEE Trans Neural Netw* 6:280–282
- Gen M, Ida K, Kobuchi R (1998) Neural network technique for fuzzy multiobjective linear programming. *Comput Ind Eng* 35(3–4):543–546
- Ghasabi-Oskoei H, Mahdavi-Amiri N (2006) An efficient simplified neural network for solving linear and quadratic programming problems. *Appl Math Comput* 175:452–464
- Ghasabi-Oskoei H, Malek A, Ahmadi A (2007) Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems. *Comput Math Appl* 53:1439–1454

24. Gong D, Gen M, Yamazaki G, Xu W (1997) Lagrangian ANN for convex programming with linear constraints. *Comput Ind Eng* 32:429–443
25. Greenberg HJ (1987) ANALYZE: a computer-assisted analysis system for linear programming models. *Oper Res Lett* 6:249–255
26. Greenberg HJ (1989) Neural networks for an intelligent mathematical programming system. In: *Proceedings of CSTS symposium: impacts of recent computer advances on operations research*. Elsevier Science Publishers, Amsterdam, 1989, pp 313–320
27. He X, Li C, Huang T, Li C (2014) Neural network for solving convex quadratic bilevel programming problems. *Neural Netw* 51:17–25
28. He X, Li C, Huang T, Li C, Huang J (2014) A recurrent neural network for solving bilevel linear programming problem. *IEEE Trans Neural Netw Learn Syst* 25(4):824–830
29. Hu T, Guo X, Fu X, Lv Y (2010) A neural network approach for solving linear bilevel programming problem. *Knowl-Based Syst* 23(3):239–242
30. Hu X (2009) Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints. *Neurocomputing* 72:1131–1137
31. Hu X, Wang J (2008) An improved dual neural network for solving a class of quadratic programming problems and its  $k$ -winners-take-all application. *IEEE Trans Neural Netw* 19(12):2022–2031
32. Hu X, Zhang B (2009) A new recurrent neural network for solving convex quadratic programming problems with an application to the  $k$ -winners-take-all problem. *IEEE Trans Neural Netw* 20(4):654–664
33. Huang X, Lou X, Cui B (2016) A novel neural network for solving convex quadratic programming problems subject to equality and inequality constraints. *Neurocomputing* 214:23–31
34. Kennedy MP, Chua LO (1998) Neural network for nonlinear programming. *IEEE Trans Circuits Syst* 35(5):554–562
35. Lachhwani K, Dwivedi A (2017) Bi-level and multi-level programming problems: taxonomy of literature review and research issues. *Arch Comput Methods Eng*. <https://doi.org/10.1007/s11831-017-9216-5> (online published)
36. Lan KM, Wen UP, Shih SH, Lee ES (2007) A hybrid neural network approach to bilevel programming problems. *Appl Math Lett* 20:880–884
37. Leung Y, Chen K, Gao X (2003) A high-performance feedback neural network for solving convex nonlinear programming problems. *IEEE Trans Neural Netw* 14(6):1469–1477
38. Leung Y, Chen KJ, Jiao YC, Gao XB, Leung KS (2001) A new gradient-based neural network for solving linear and quadratic programming problems. *IEEE Trans Neural Netw* 12(5):1074–1083
39. Li HX, Da XL (2000) A neural network representation of linear programming. *Eur J Oper Res* 124:224–234
40. Liao LZ, Hou-Duo QIA (1999) Neural network for the linear complementarity problem. *Math Comput Model* 29:9–18
41. Liu Q, Wang J (2008) A one-layer recurrent neural network with a discontinuous activation function for linear programming. *Neural Comput* 20(5):1366–1383
42. Liu S, Wang J (2006) A simplified dual neural network for quadratic programming with its KWTA application. *IEEE Trans Neural Netw* 17(6):1500–1510
43. Jin L, Li S (2017) Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neurocomputing* 267:107–113
44. Lv Y, Chen Z, Wan Z (2011) A neural network approach for solving mathematical programs with equilibrium constraints. *Expert Syst Appl* 38:231–234
45. Lv Y, Chena Z, Wan Z (2010) A neural network for solving a convex quadratic bilevel programming problem. *J Comput Appl Math* 234:505–511
46. Lv Y, Hu T, Wang G, Wan Z (2007) A neural network approach for solving nonlinear bilevel programming problem. *Comput Math Appl* 55:2823–2829
47. Maa CY, Shanblatt MA (1992) Linear and quadratic programming neural network analysis. *IEEE Trans Neural Netw* 3(4):580–594
48. Malakooti B, Zhou Y (1990) An adaptive feedforward artificial neural network with application to multiple criteria decision making. In: *Conference proceedings, IEEE international conference on systems, man and cybernetics*
49. Malek A, Alipour M (2007) Numerical solution for linear and quadratic programming problems using a recurrent neural network. *Appl Math Comput* 192:27–39
50. Malek A, Yari A (2005) Primal–dual solution for the linear programming problems using neural networks. *Appl Math Comput* 167:198–211
51. Mansoori A, Effati S, Eshaghnezhad M (2018) A neural network to solve quadratic programming problems with fuzzy parameters. *Fuzzy Optim Decis Making* 17(1):75–101
52. Marta I, Fontova V, Aurelio RLO, Lyra C (2012) Hopfield neural networks in large-scale linear optimization problems. *Appl Math Comput* 218:6851–6859
53. Meida-Casermeiro E, Galan-Mari G, Munoz-Perez J (2001) An efficient multivalued hopfield network for the traveling salesman problem. *Neural Process Lett* 14:203–216
54. Alipour M (2011) A novel recurrent neural network model for solving nonlinear programming problems with general constraints. *Aust J Basic Appl Sci* 5(10):814–823
55. Nasira GM, Ashok Kumar S, Balaji TSS (2010) Neural network implementation for integer linear programming problem. *Int J Comput Appl* 1(18):93–97
56. Nguyen KV (2000) A nonlinear neural network for solving linear programming problems. In: *International symposium on mathematical programming, ISMP 2000, Atlanta, GA, USA*
57. Ranjbar M, Effati S, Miri SM (2017) An artificial neural network for solving quadratic zero-one programming problems. *Neurocomputing* 235:192–198
58. Rodriquez-Vazquez A, Rueda A, Huertas JL, Dominguez-Castro R (1988) Sinencio switched-capacitor neural networks for linear programming. *Electron Lett* 24(8):496–498
59. Neeraj S, Kumar A (2010) Solution of the linear programming problems based on neural network approach. *Int J Comput Appl* 9(10):24–27
60. Selvaraj G, Pandian P (2013) A neural network approach for fuzzy linear programming problems. In: *Proceedings of the national conference on recent trends in mathematical computing—NCRPMC' 13*, pp 21–27
61. Shih H, Wen U, Lee ES, Lan KM, Hsiao HC (2004) A neural network approach to multiobjective and multilevel programming problems. *Comput Math Appl* 48:95–108
62. Sun M, Stam A, Steuer RE (1996) Solving multiple objective programming problems using feed-forward artificial neural networks: the interactive FFANN procedure. *Manage Sci* 42(6):835–849
63. Sun M, Stam A, Steuer RE (2000) Interactive multiple objective programming using Tchebycheff programs and artificial neural networks. *Comput Oper Res* 27:601–620
64. Tao Q, Cao J, Sun D (1999) A simple and high performance neural network for quadratic programming problems. *Appl Math Comput* 124(2):251–260
65. Tao Q, Cao JD, Xue MS, Qiao H (2001) A high performance neural network for solving nonlinear programming problems with hybrid constraints. *Phys Lett A* 288(2):88–94
66. Vahabi HR, Ghasabi-Oskoei H (2012) A feedback neural network for solving nonlinear programming problems with hybrid constraints. *Int J Comput Appl* 54(5):41–46

67. Walsh MP, Flynn ME, O'Malley MJ (1999) Augmented hopfield network for mixed integer programming. *IEEE Trans Neural Netw* 10(2):456–458
68. Wang J (1992) Recurrent neural network for solving quadratic programming problems with equality constraints. *Electron Lett* 28(14):1345–1347
69. Wang J (1993) Analysis and design of a recurrent neural network for linear programming. *IEEE Trans Circuits Syst I Fund Theory Appl* 40:613–618
70. Wang J (1994) A deterministic annealing neural network for convex programming. *Neural Netw* 7(4):629–641
71. Wang J, Chankong V (1992) Recurrent neural networks for linear programming: analysis and decision principles. *Comput Oper Res* 19(2):297–311
72. Wang J, Malakooti B (1992) A feed forward neural network for multiple criteria decision making. *Comput Oper Res* 19(2):151–167
73. Wen UP, Lan KM, Shih SH (2009) A review of hopfield neural networks for solving mathematical programming problems. *Eur J Oper Res* 198(3):675–687
74. Wu XY, Xia YS, Li J, Chen WK (1996) A high-performance neural network for solving linear and quadratic programming problems. *IEEE Trans Neural Netw* 7(3):643–651
75. Xia Y (1996) A new neural network for solving linear programming problems and its applications. *IEEE Trans Neural Netw* 7(2):525–529
76. Xia Y (1997) Neural network for solving extended linear programming problems. *IEEE Trans Neural Netw* 8(3):803–806
77. Xia Y, Wang J (2005) A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Trans Neural Netw* 16(2):379–386
78. Xia Y, Wang J (1995) Neural network for solving linear programming problem with bounded variables. *IEEE Trans Neural Netw* 6:515–519
79. Xia YS (1996) A new neural network for solving linear and quadratic programming problems. *IEEE Trans Neural Netw* 7(6):1544–1547
80. Yaakob SB, Watada J (2010) Double-layered hybrid neural network approach for solving mixed integer quadratic bilevel problems. In: Huynh VN, Nakamori Y, Lawry J, Inuiguchi M (eds) *Integrated uncertainty management and applications. Advances in intelligent and soft computing*, vol 68. Springer, Berlin, pp 221–230
81. Yaakob SB, Watada J (2011) Solving bilevel programming problems using a neural network approach and its application to power system environment. *SICE J Control Meas Syst Integr* 4(6):387–393
82. Yang J, Du T (2010) A neural network algorithm for solving quadratic programming based on Fibonacci method. In: Zhang L., Lu BL., Kwok J. (eds) *Advances in neural networks—ISNN 2010. ISNN 2010. Lecture notes in computer science*, vol 6063, pp 118–125. Springer, Berlin
83. Yang Y, Cao J (2006) Solving quadratic programming problems by delayed projection neural network. *IEEE Trans Neural Netw* 17(6):1630–1634
84. Yang Y, Cao J, Xu X, Hu M, Gao Y (2014) A new neural network for solving quadratic programming problems with equality and inequality constraints. *Math Comput Simul* 101:103–112
85. Yang Y, Gao Y (2011) A new neural network for solving nonlinear convex programs with linear constraints. *Neurocomputing* 74:3079–3083
86. Zak SH, Upatising V, Hui S (1995) Solving linear programming problems with neural networks: a comparative study. *IEEE Trans Neural Netw* 6(1):94–104
87. Zhang S, Constantinides AG (1992) Lagrange programming neural networks. *IEEE Trans Circuits Syst* 39(7):441–452
88. Zhang Y (2005) On the LVI-based primal–dual neural network for solving online linear and quadratic programming problems. In: *Proceedings of the American control conference*, pp 1351–1356. <https://doi.org/10.1109/acc.2005.1470152>
89. Zhang Y, Wang J (2002) A dual neural network for convex quadratic programming subject to linear equality and inequality constraints. *Phys Lett A* 298:271–278