Solution of Linear Programming Problems using a Neural Network with Non-Linear Feedback

Syed Atiqur RAHMAN, Mohd. Samar ANSARI, Athar Ali MOINUDDIN

Department of Electronics Engineering, A.M.U., Aligarh, India

atiqamu@gmail.com, mdsamar@gmail.com, aamoin@gmail.com

Abstract. *This paper presents a recurrent neural circuit for solving linear programming problems. The objective is to minimize a linear cost function subject to linear constraints. The proposed circuit employs non-linear feedback, in the form of unipolar comparators, to introduce transcendental terms in the energy function ensuring fast convergence to the solution. The proof of validity of the energy function is also provided. The hardware complexity of the proposed circuit compares favorably with other proposed circuits for the same task. PSPICE simulation results are presented for a chosen optimization problem and are found to agree with the algebraic solution. Hardware test results for a 2-variable problem further serve to strengthen the proposed theory.*

Keywords

Linear programming, dynamical systems, neural networks, feedback systems, non-linear feedback.

1. Introduction

Mathematical programming, in general, is concerned with the determination of a minimum or a maximum of a function of several variables, which are required to satisfy a number of constraints. Such solutions are sought in diverse fields including engineering, operations research, management science, computer science, numerical analysis, and economics [1], [2].

A general mathematical programming problem can be stated as [2]:

$$\text{Minimize } f(\mathbf{x}) \tag{1}$$

subject to

$$g_i(\mathbf{x}) \geq 0 \ (i = 1,2,\ldots,m), \tag{2}$$

$$h_j(\mathbf{x}) = 0 \ (j = 1,2,\ldots,p), \tag{3}$$

$$\mathbf{x} \in S \tag{4}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the vector of unknown decision variables, and f, $g_i (i = 1,2,\ldots,m)$, $h_j (j = 1,2,\ldots,p)$ are the real-valued functions of the n real variables $x_1, x_2, \ldots, x_n$.

In this formulation, the function f is called the *objective function*, and inequalities (2), equations (3) and the set restrictions (4) are referred to as the *constraints*. It may be mentioned that although the mathematical programming problem (MPP) has been stated as a minimization problem in the description above, the same may readily be converted into a maximization problem without any loss of generality, by using the identity given in (5)

$$\max \ f(\mathbf{x}) \ = \ -\min\,[-f(\mathbf{x})]. \tag{5}$$

As a special case, if all the functions appearing in the MPP are linear in the decision variables $\mathbf{x}$, the problem is referred to as a *linear programming problem* (LPP). Such LPPs have been investigated extensively over the past decades, in view of their fundamental roles arising in a wide variety of engineering and scientific applications, such as pattern recognition [3], signal processing [4], human movement analysis [5], robotic control [6], and data regression [7]. Other real life applications include portfolio optimization [8], crew scheduling [9], manufacturing and transportation [10], telecommunications [11], and the Traveling Salesman Problem (TSP) [12].

Traditional methods for solving linear programming problems typically involve an iterative process, but long computational time limits their usage. An alternative approach to solution of this problem is to exploit the artificial neural networks (ANN's) which can be considered as an analog computer relying on a highly simplified model of neurons [13]. ANN's have been applied to several classes of constrained optimization problems and have shown promise for solving such problems more effectively. For example, the Hopfield neural network has proven to be a powerful tool for solving some of the optimization problems. Tank and Hopfield first proposed a neural network for solving mathematical programming problems, where a linear programming problem (LPP) was mapped into a closed-loop network [14]. A brief overview of the various neural network based approaches which have been proposed over the past is presented in the next section.

In this paper, a hardware solution to the linear programming problem is presented. The proposed architecture uses non-linear feedback which leads to a new energy function that involves transcendental terms. This transcendental energy function is fundamentally different from the standard quadratic form associated with Hopfield network and its

variants. To solve a LPP in n variables with m constraints, the circuit requires n opamps, m unipolar comparators and $(m+n)$ resistors thereby causing the hardware complexity of the proposed network to compare favorably with the existing hardware implementations. It may be mentioned that a similar approach of using non-linear synaptic interconnections between neurons has also been employed to solve systems of simultaneous linear equations [15] and quadratic programming [16]. An initial result on the solution of a linear programming problem in two variables using this approach appears in [17].

The remainder of this paper is arranged as follows. A brief review of relevant technical literature on the solution of LPP using neural network based methods is presented in Section 2. Section 3 outlines the mathematical formulation of the basic problem and details of the proposed network. Section 4 contains explanation of the energy function and the proof of its validity. Section 5 contains the circuit implementation of the proposed network for a set of sample problem in four variables. PSPICE simulation results of the proposed circuit are also presented. Results of breadboard implementation of the proposed circuit for a 2-variable problem are contained in Section 6. Issues that are expected to arise in actual monolithic implementations are discussed in Section 7. Concluding remarks are presented in Section 8.

2. Existing Neural Networks for LPP

LPP has received considerable research attention from the neural networks community. The first solution of the linear programming problem was proposed by Tank and Hopfield wherein they used the continuous-time Hopfield network [14]. From the computational aspect, the operation of Hopfield network for an optimization problem, like the LPP, manages a dynamic system characterized by an energy function, which is the combination of the objective function and the constraints of the original problem [18]. Over the years, the penalty function approach has become a popular technique for solving optimization problems. Kennedy & Chua proposed an improved version of Tank & Hopfield's network for LPP in which an inexact penalty function was considered [19]. The requirement of setting a large number of parameters was a major drawback of Kennedy & Chua's LPP network [4]. Rodriguez-Vazquez *et al.* used a different penalty method to transform the given LPP into an unconstrained optimization problem [20]. Although Rodriguez-Vazquez *et al.* later pointed out that their network had no equilibrium point in the classical sense [21], investigations by Lan *et al.* proved that the network can indeed converge to an optimal solution of the given problem from any arbitrary initial condition [22]. Maa & Shanblatt employed a two-phase neural network architecture for solving LPPs [23]. Chong *et al.* analyzed a class of neural network models for the solution of LPPs by dynamic gradient approaches based on exact non-differentiable penalty functions [24]. They also developed an analytical tool aimed at helping the system converge to a solution within a finite time. In an approach different from the penalty function methods, Zhu, Zhang and Constantinides proposed a Lagrange method for solving LPPs through Hopfield networks [25]. Xia and Wang used bounded variables to construct a new neural network approach to solve LPP with no penalty parameters. They suggested that the equilibrium point is the same as the exact solution when the primal and dual problems are solved simultaneously [26]. More recently, Malek & Yari proposed two new methods for solving the LPP and presented optimal solutions with efficient convergence within a finite time [27]. Lastly, Ghasabi-Oskoei, Malek and Ahmadi have presented a recurrent neural network model for solving LPP based on a dynamical system using arbitrary initial conditions. The method does not require analog multipliers thereby reducing the system complexity [28].

3. Proposed Circuit

Let the first-order function to be minimized be

$$F = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \tag{6}$$

subject to the following linear constraints

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{7}$$

where $V_1, V_2, \dots, V_n$ are the variables, and a_{ij}, c_j and b_i $(i = 1,2,\dots,m; j = 1,2,\dots,n)$ are constants. The proposed neural-network based circuit to minimize the function given in (6) in accordance with the constraints of (7) is presented in Fig. 1. As can be seen from Fig. 1, individual inequalities from the set of constraints are passed through non-linear synapses which are realized using unipolar comparators. The outputs of the comparators are fed to neurons having weighted inputs. The neurons are realized by using opamps and the weights are implemented using resistances. R_{pi} and C_{pi} are the input resistance and capacitance of the opamp that is used to emulate the functionality of a neuron. These parasitic components are included to model the dynamic nature of the opamp.

If the unipolar comparator is operated with a single $+V_m$ supply, while the opamp realizing the neuronal functionality being biased with $\pm V_m$, the obtained transfer characteristics of the unipolar voltage comparator are presented in Fig. 2 and can be mathematically modelled by (8). As explained in the next section, such unipolar comparator characteristics are utilized to obtain an energy function which acts to bring the neuronal states to the *feasible* region.

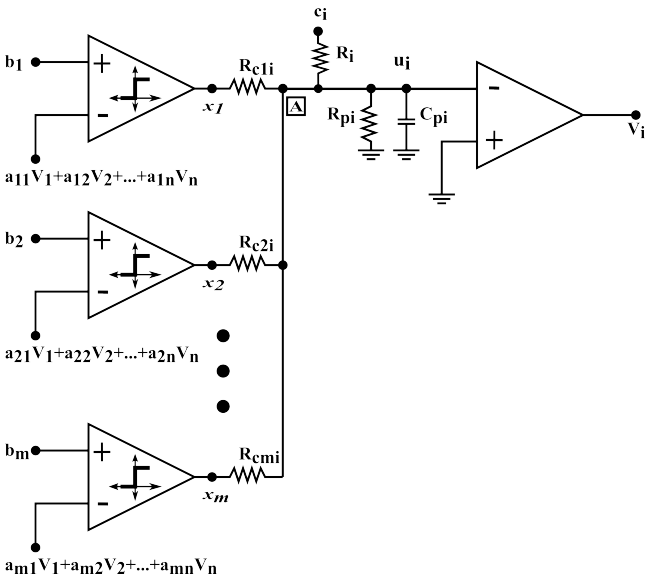

Fig. 1. *i*–th neuron of the proposed feedback neural network circuit to solve a linear programming problem in *n* variables with *m* linear constraints.

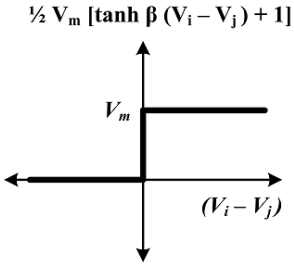

Fig. 2. Transfer characteristics of the unipolar comparator.

$$x = \frac{1}{2} V_m [\tanh \beta \ (V_i - V_j) + 1]. \tag{8}$$

Using (8), the output of the *i*-th unipolar comparator in Fig. 1 can be given by (9) where β is the open-loop gain of the comparator (practically very high), ± V_m are the output voltage levels of the comparator and V_1, V_2, ..., V_n are the neuron outputs.

$$x_i = \frac{1}{2} V_m \ [\tanh \beta \ (a_{i1}V_1 + a_{i2}V_2 + \ldots + a_{in}V_n - b_i) + 1]. \tag{9}$$

Applying node equations for node 'A' in Fig. 1, the equation of motion of the *i*-th neuron can be given as

$$C_{pi}\frac{du_i}{dt} = \sum_{j=1}^{m}\left[\frac{x_j}{R_{cji}}\right] + \frac{c_i}{R_i} - \frac{u_i}{R_{ieqv}} \tag{10}$$

where R_{ieqv} is the parallel equivalent of all resistances connected at node 'A' in Fig. 1 and is given by (11)

$$\frac{1}{R_{ieqv}} = \sum_{j=1}^{m}\left[\frac{1}{R_{cji}}\right] + \frac{1}{R_i} + \frac{1}{R_{pi}} \tag{11}$$

where u_i is the internal state of the *i*-th neuron, R_{c1i}, R_{c2i}, ..., R_{cmi} are the weight resistances connecting the outputs of the unipolar comparators to the input of the *i*-th neuron. As it is shown later in this section, the values of these resistances are governed by the entries in the coefficient matrix of (7). Resistance R_i causes terms corresponding to the linear function to be minimized, to appear in the energy function.

It may be mentioned that the concept of a Lyapunov or energy function being associated with gradient-type neural networks was first employed by Hopfield in the stability analysis of the so-called Hopfield Neural Network (HNN) [14]. The computational energy for dynamical systems like the HNN and the Non-Linear Synapse Neual Network presented in this paper, decreases continuously in time with the network converging to a minimum in state space. The evolution of the system is in the general direction of the negative gradient of the energy function. Typically, the network energy function is made equivalent to a certain objective function that needs to be minimized. The search for an energy minimum performed by the network correspond to the search for the solution of an optimization problem.

As explained in Section 4, the energy function associated with the non-linear feedback neural circuit of Fig. 1, for minimizing a linear objective function subject to linear constraints, is given by

$$E = \sum_{i=1}^{n} c_i V_i + \frac{V_m}{2} \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}V_j + \frac{V_m}{2\beta} \sum_{i=1}^{m} \ln \cosh \beta \left[\sum_{j=1}^{n}(a_{ij}V_j - b_i)\right]. \tag{12}$$

This expression of the energy function can be written in a slightly different (but more illuminating) form as

$$E = \sum_{i=1}^{n} c_i V_i + (P_1 + P_2 + \ldots + P_m) \tag{13}$$

where the first term is the same as the first-order function to be minimized, as given in (6), and P_1, P_2, ..., P_m are the penalty terms. The *i*-th penalty term can be given as

$$P_i = \frac{V_m}{2} \sum_{j=1}^{n} a_{ij}V_j + \frac{V_m}{2\beta} \ln \cosh \beta \left[\sum_{j=1}^{n}(a_{ij}V_j - b_i)\right]. \tag{14}$$

Obtaining a partial differential of the combined penalty term, $P(= P_1 + P_2 + \ldots + P_m)$ with respect to V_i we have

$$\frac{\partial P}{\partial V_i} = \frac{V_m}{2} \sum_{j=1}^{m} a_{ji} + \frac{V_m}{2} \sum_{j=1}^{m} a_{ji} \tanh \beta \left[\sum_{j=1}^{n}(a_{ij}V_j - b_i)\right] \tag{15}$$

which may be simplified to

$$\frac{\partial P}{\partial V_i} = \sum_{j=1}^{m} a_{ji}x_j. \tag{16}$$

Using the above relations to find the derivative of the energy function E with respect to V_i we have

$$\frac{\partial E}{\partial V_i} = \frac{\partial}{\partial V_i}\left[\sum_{i=1}^{n} c_i V_i\right] + \frac{\partial P}{\partial V_i} \tag{17}$$

which in turn yields

$$\frac{\partial E}{\partial V_i} = c_i + \sum_{j=1}^{m} a_{ji} x_j. \tag{18}$$

Also, if E is the energy function, it must satisfy the following condition [29]:

$$\frac{\partial E}{\partial V_i} = KC_{pi}\frac{du_i}{dt} \tag{19}$$

where K is a constant of proportionality and has the dimensions of resistance. Using (9), (10) and (18) in (19) results in (for the i–th neuron)

$$R_{cji} = K/a_{ji}\ ; \quad (j = 1,2,\ldots,m). \tag{20}$$

A similar comparison of the remaining partial fractions for the remaining neurons yields the following:

$$R_{cji} = K/a_{ji}\ ; \quad \begin{matrix}(j = 1,2,\ldots,m),\\ (i = 1,2,\ldots,n),\end{matrix} \tag{21}$$

$$R_i = K\ ; \quad (i = 1,2,\ldots,n). \tag{22}$$

4. Energy Function

This section deals with the explanation of individual terms in the energy function expression given in (12). The last term is transcendental in nature and an indicative plot showing the combined effect of the last two terms is presented in Fig. 3. As can be seen, one 'side' of the energy landscape is flat whilst the other has a slope directed to bring the system state towards the side of the flat slope. During the actual operation of the proposed LPP solving circuit, the comparators remain effective only when the neuronal output states remain outside the feasible region and during this condition, these unipolar comparators work to bring (and restrict) the neuron output voltages to the feasible region. Once that is achieved, first term in (12) takes over and works to minimize the given function.

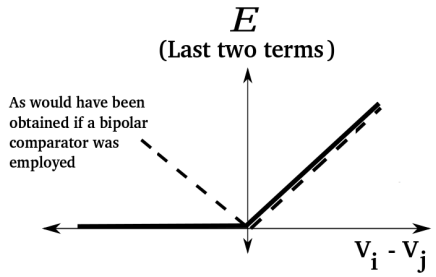

Fig. 3. Combined effect of last two terms in (12).

The validity of the energy function of (12) can be proved as follows. The time derivative of the energy function is given by

$$\frac{dE}{dt} = \sum_{i=1}^{n}\frac{\partial E}{\partial V_i}\frac{dV_i}{dt} = \sum_{i=1}^{n}\frac{\partial E}{\partial V_i}\frac{dV_i}{du_i}\frac{du_i}{dt}. \tag{23}$$

Using (19) in (23) we get

$$\frac{dE}{dt} = \sum_{i=1}^{n} KC_{pi}\left(\frac{du_i}{dt}\right)^2\frac{dV_i}{du_i}. \tag{24}$$

The transfer characteristics of the output opamp used to implement the neurons in Fig. 1 implements the activation function of the neuron and can be written as

$$V_i = f(u_i) \tag{25}$$

where V_i denotes the output of the opamp and u_i corresponds to the internal state at the inverting terminal. The function f is typically a saturating, monotonically decreasing one, as shown in Fig. 4, and therefore [15],

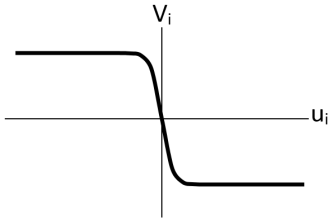

Fig. 4. Transfer characteristics of the opamp used to realize the neurons.

$$\frac{dV_i}{du_i} \leq 0 \tag{26}$$

thereby resulting in

$$\frac{dE}{dt} \leq 0 \tag{27}$$

with the equality being valid for

$$\frac{du_i}{dt} = 0. \tag{28}$$

Equation (27) shows that the energy function can never increase with time which is one of the conditions for a valid energy function. The second criterion i.e. the energy function must have a lower bound is also satisfied for the circuit of Fig. 1 wherein it may be seen that V_1, V_2, …, V_n are all bounded (as they are the outputs of opamps, as given in (25)) amounting to E, as given in (12), having a finite lower bound.

5. Simulation Results

This section deals with the application of the proposed network to the task of minimizing the objective function

$$V_1 + 2V_2 - V_3 + 3V_4 \tag{29}$$

subject to

$$\begin{aligned} V_1 - V_2 + V_3 &\leq 4,\\ V_1 + V_2 + 2V_4 &\leq 6,\\ V_2 - 2V_3 + V_4 &\leq 2,\\ -V_1 + 2V_2 + V_3 &\leq 2,\\ V_1 &\geq 0,\\ V_4 &\geq 0. \end{aligned} \tag{30}$$

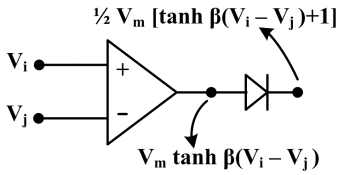

Fig. 5. Obtaining unipolar comparator characteristics using an opamp and a diode.

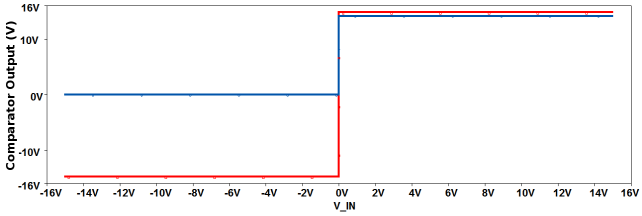

Fig. 6. Transfer characteristics for opamp based unipolar and bipolar comparators as obtained from PSPICE simulations.

The values of resistances acting as the weights on the neurons are obtained from (21), (22). For the purpose of simulation, the value of K was chosen to be 1 kΩ. Using K = 1 kΩ in (21), (22) gives

$$\begin{bmatrix} R_{c11} & R_{c12} & R_{c13} & R_{c14} \\ R_{c21} & R_{c22} & R_{c23} & R_{c24} \\ R_{c31} & R_{c32} & R_{c33} & R_{c34} \\ R_{c41} & R_{c42} & R_{c43} & R_{c44} \\ R_{c51} & R_{c52} & R_{c53} & R_{c54} \\ R_{c61} & R_{c62} & R_{c63} & R_{c64} \end{bmatrix} = \begin{bmatrix} 1K & -1K & 1K & \infty \\ 1K & 1K & \infty & 0.5K \\ \infty & 1K & -0.5K & 1K \\ -1K & 0.5K & 1K & \infty \\ -1K & \infty & \infty & \infty \\ \infty & \infty & \infty & -1K \end{bmatrix}, \tag{31}$$

$$R_1 = R_2 = R_3 = R_4 = 1K. \tag{32}$$

For the purpose of PSPICE simulations, the unipolar voltage comparator was realized using a diode with an opamp based comparator as shown in Fig. 5. The transfer characteristics obtained during the PSPICE simulations for opamp based bipolar and unipolar comparators are presented in Fig. 6 from where it can be observed that the obtained unipolar characterisitics are in agreement with the ideal characteristics of Fig. 2. For the purpose of this simulation, the LMC7101A CMOS opamp model from the Orcad library in PSPICE was utilised. The value of β for this opamp was measured to be 1.1×10^4 using PSPICE simulation. For the negative values arising in (31), inverted outputs would be required from the unipolar voltage comparators. For the present simulation, inverting amplifiers were employed at the outputs of the comparators which need negative weights.

Routine mathematical analysis of (29) yields: $V_1 = 0$, $V_2 = -10$, $V_3 = -6$ and $V_4 = 0$. The resultant plots of the neuron output voltages as obtained after PSPICE simulation are presented in Fig. 7 from where it can be seen that V(1) = 110 μV, V(2) = −10.28 V, V(3) = −6.17 V and V(4) = 110 μV which are very near to the algebraic solution thereby confirming the validity of the approach. For emulating a more realistic 'power-up' scenario, random initial values in the milli-volt range were assigned to node voltages.

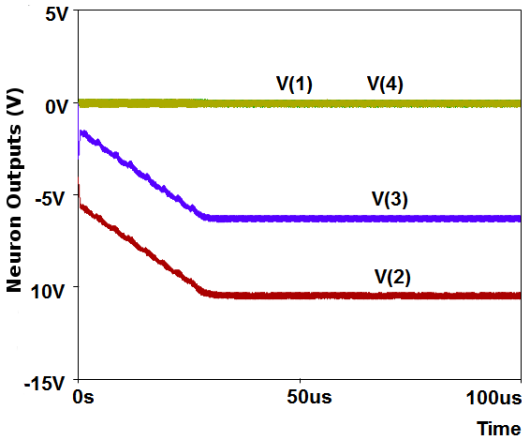

Fig. 7. Simulation results for the proposed circuit applied to minimize (29) subject to (30).

One set of initial node voltage was: V(1) = 2 mV, V(2) = 7 mV, V(3) = -5 mV and V(4) = 10 mV.

6. Hardware Test Results

Breadboad implementation of the proposed circuit was also carried out. Apart from verification of the working of the proposed circuit, the actual circuit realization also served the purpose of testing the convergence of the circuit to the solution starting from different initial conditions. The noise present in any electronic circuit acts as a random initial condition for the convergence of the neural circuit. Standard laboratory components i.e. the μA741 opamp and resistances were used for the purpose. A 2–variable problem for the minimization of the objective function

$$2V_1 + 6V_2 \tag{33}$$

subject to

$$\begin{aligned} V_1 &\geq 1, \\ V_2 &\geq 1 \end{aligned} \tag{34}$$

was chosen for the hardware tests. The values of resistances acting as the weights on the neurons are obtained from (21), (22). The value of K was chosen to be 1 kΩ. Using K = 1 kΩ in (21), (22) gives $R_1 = R_2$ = 1 kΩ, $R_{c11} = R_{c22}$ = 1 kΩ and $R_{c12} = R_{c21} = \infty$. The voltages applied were $b_1 = b_2$ = 1 V, c_1 = 2 V and c_2 = 6 V. The obtained values of the neuronal voltages were V_1 = 1.06 V and V_2 = 1.02 V which are in close agreement with the exact mathematical solution which is V_1 = 1 and V_2 = 1. a snapshot of the obtained results is presented in Fig. 8.

7. Issues in Actual Implementation

This section deals with the monolithic implementation issues of the proposed circuit. The PSPICE simulations assumed that all operational amplifiers (and diodes) are identical, and therefore, it is required to determine how deviations from this assumption affect the performance of the network.

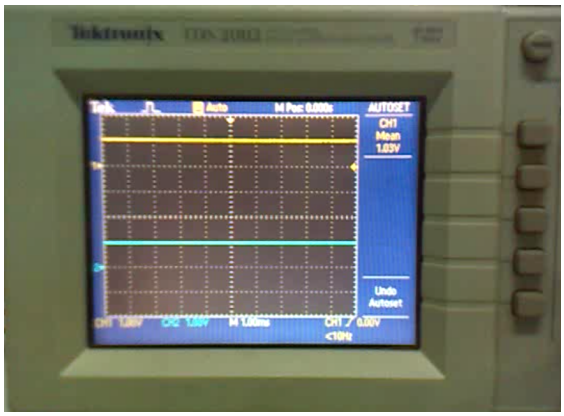

Fig. 8. Hardware test result for the proposed circuit applied to minimize (33) subject to (34).

A 10 % tolerance with Gaussian deviation profile was put on the resistances used in the circuit to solve (29). The analysis was carried out for 200 runs and the Mean Deviation was found out to be -21.23$\times 10^{-06}$ and Mean Sigma (Standard Deviation) was 0.013. Offset analysis was also carried out by incorporating random offset voltages (in the range of 1 mV to 10 mV) to the opamps. The Mean Deviation in this case was measured to be -20.19$\times 10^{-06}$ and the Mean Sigma (Standard Deviation) was 0.007. As can be seen, the effects of mismatches and offsets on the overall precision of the final results are in an acceptable range.

Effects of non-idealities in various components were further investigated in PSPICE by testing the circuit with all resistances having the same percentage deviation from their assigned values. The resulting assessment of the quality of solution is presented in Tab. 1 from where it is evident that the solution point does not change much even for high deviations in the resistance values.

Next, the effect of offset voltages in the opamp-based unipolar comparators was explored. Offset voltages were applied at the inverting terminals of the comparators and the results of PSPICE simulations for the chosen LPP are given in Tab. 2. As can be seen, the offset voltages of the comparators do not affect the obtained solutions to any appreciable extent. However, the error does tend to increase with increasing offset voltages.

Finally, offset voltages for the opamps emulating the neurons were also considered. Offset voltages were applied at the non-inverting inputs of the opamps and the results of PSPICE simulations were compared with the algebraic solution as given in Tab. 3. As can be seen, the offset voltages of the opamps have little effect on the obtained solutions.

In fact, the realization of unipolar comparators by the use of opamps and diodes in the proposed circuit tends to increase the circuit complexity. The transistor count can be further reduced by utilising voltage-mode unipolar comparators instead of the opamp-diode combination. This also suggests that a real, large scale implementation for solving linear programming problems with high variable counts

Variation in Resistances	+2%	+5%	+10%	-2%	-5%	-10%
PSPICE Simulation Results	$\begin{bmatrix} 100\,\mu V \\ -10.29\,V \\ -6.17\,V \\ 115\,\mu V \end{bmatrix}$	$\begin{bmatrix} 113\,\mu V \\ -10.35\,V \\ -6.23\,V \\ 267\,\mu V \end{bmatrix}$	$\begin{bmatrix} 214\,\mu V \\ -10.77\,V \\ -6.33\,V \\ 332\,\mu V \end{bmatrix}$	$\begin{bmatrix} 98\,\mu V \\ -10.28\,V \\ -6.17\,V \\ 129\,\mu V \end{bmatrix}$	$\begin{bmatrix} 105\,\mu V \\ -9.72\,V \\ -5.91\,V \\ 512\,\mu V \end{bmatrix}$	$\begin{bmatrix} 126\,\mu V \\ -9.89\,V \\ -5.69\,V \\ 875\,\mu V \end{bmatrix}$

Tab. 1. Effect of variation in resistances on the obtained results.

Offset voltage applied at inverting terminal of comparator opamps	-5 mV	-10 mV	-15 mV	+5 mV	+10 mV	+15 mV
PSPICE Simulation Results	$\begin{bmatrix} 103\,\mu V \\ -10.22\,V \\ -6.17\,V \\ 112\,\mu V \end{bmatrix}$	$\begin{bmatrix} 100\,\mu V \\ -10.04\,V \\ -6.11\,V \\ 98\,\mu V \end{bmatrix}$	$\begin{bmatrix} 89\,\mu V \\ -9.92\,V \\ -6.03\,V \\ 77\,\mu V \end{bmatrix}$	$\begin{bmatrix} 127\,\mu V \\ -10.28\,V \\ -6.18\,V \\ 314\,\mu V \end{bmatrix}$	$\begin{bmatrix} 105\,\mu V \\ -10.34\,V \\ -6.29\,V \\ 623\,\mu V \end{bmatrix}$	$\begin{bmatrix} 105\,\mu V \\ -10.84\,V \\ -6.33\,V \\ 746\,\mu V \end{bmatrix}$

Tab. 2. Effect of offset voltages of the opamp-based unipolar comparators on the solution quality.

Offset voltage applied at non-inverting terminal of neuronal opamps	-5 mV	-10 mV	-15 mV	+5 mV	+10 mV	+15 mV
PSPICE Simulation Results	$\begin{bmatrix} 111\,\mu V \\ -10.28\,V \\ -6.17\,V \\ 123\,\mu V \end{bmatrix}$	$\begin{bmatrix} 119\,\mu V \\ -10.29\,V \\ -6.18\,V \\ 221\,\mu V \end{bmatrix}$	$\begin{bmatrix} 189\,\mu V \\ -10.32\,V \\ -6.20\,V \\ 323\,\mu V \end{bmatrix}$	$\begin{bmatrix} 127\,\mu V \\ -10.12\,V \\ -6.13\,V \\ 114\,\mu V \end{bmatrix}$	$\begin{bmatrix} 69\,\mu V \\ -9.98\,V \\ -6.09\,V \\ 223\,\mu V \end{bmatrix}$	$\begin{bmatrix} 10\,\mu V \\ -9.89\,V \\ -6.04\,V \\ 246\,\mu V \end{bmatrix}$

Tab. 3. Effect of offset voltages of the opamp-based neurons on the solution quality.

might be quite different. Alternative realizations based on the differential equations (10) governing the system of neurons are being investigated. Other approaches to obtain the tanh(.) non-linearity include the use of a MOSFET operated in the sub-threshold region [30] and the use of Current Differencing Transconductance Amplifier (CDTA) to provide the same nonlinearity in the current-mode regime [31].

8. Conclusion

In this paper, a CMOS compatible approach to solve a linear programming problem in n variables subject to m linear constraints, which uses n-neurons and m-synapses is presented. Each neuron requires one opamp and each synapse is implemented using a unipolar voltage-mode comparator. This results in significant reduction in hardware over the existing schemes. The proposed network was tested on a sample problem of minimizing a linear function in 4 variables and the simulation results confirm the validity of the approach. Hardware verification for a 2–variable problem further validated the theory proposed.

References

[1] KREYSZIG, E. *Advanced Engineering Mathematics*. 8th ed. New Delhi (India): Wiley-India, 2006.

[2] KAMBO, N. S. *Mathematical Programming Techniques*. New Delhi (India): Affilated East-West Press Pvt Ltd., 1991.

[3] ANGUITA, D., BONI, A., RIDELLA, S. A digital architecture for support vector machines: Theory, algorithm, and FPGA implementation. *IEEE Transactions on Neural Networks*, 2003, vol. 14, no. 5, p. 993 - 1009.

[4] CICHOCKI, A., UNBEHAUEN, R. *Neural Networks for Optimization and Signal Processing*. Chichester (UK): Wiley, 1993.

[5] IQBAL, K., PAI, Y. C. Predicted region of stability for balance recovery: motion at the knee joint can improve termination of forward movement. *Journal of Biomechanics*, 2000, vol. 13, no. 12, p. 1619 - 1627.

[6] ZHANG, Y. Towards piecewise-linear primal neural networks for optimization and redundant robotics. In *IEEE International Conference on Networking, Sensing and Control*. Fort Lauderdale (FL, USA), 2006, p. 374 - 379.

[7] ZHANG, Y., LEITHEAD, W. E. Exploiting Hessian matrix and trust-region algorithm in hyperparameters estimation of Gaussian process. *Applied Mathematics and Computation*, 2005, vol. 171, no. 2, p. 1264 - 1281.

[8] YOUNG, M. R. A minimax portfolio selection rule with linear programming solution. *Management Science*, 1988, vol. 44, no. 5, p. 673 - 683.

[9] BIXBY, R. E., GREGORY J. W., LUSTIG, I. J., MATSTEN, R. E., SHANNO, D. F. Very large-scale linear programming: A case study in combining interior point and simplex methods. *Operations Research*, 1992, vol. 40, no. 5, p. 885 - 897.

[10] PEIDRO, D., MULA, J., JIMENEZ, M., del MAR BOTELLA, M. A fuzzy linear programming based approach for tactical supply chain planning in an uncertainty environment. *European Journal of Operational Research*, 2010, vol. 205, no. 16, p. 65 - 80.

[11] CHERTKOV, M., STEPANOV, M. G. An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes. *IEEE Transactions on Information Theory*, 2008, vol. 54, no. 4, p. 1514 - 1520.

[12] CHVATAL, V., COOK, W., DANTZIG, G. B., FULKERSON, D. R., JOHNSON, S. M. *Solution of a Large-Scale Traveling-Salesman Problem*. Berlin (Germany): Springer, 2010, p. 7 - 28.

[13] KROGH, A. What are artificial neural networks? *Nature Biotechnology*, 2008, vol. 26, no. 2, p. 195 - 197.

[14] TANK, D. W., HOPFIELD, J. J. Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 1986, vol. 33, no. 5, p. 533 - 541.

[15] RAHMAN, S. A., ANSARI, M. S. A neural circuit with transcendental energy function for solving system of linear equations. *Analog Integrated Circuits and Signal Processing*, 2011, vol. 66, no. 3, p. 433 - 440.

[16] ANSARI, M. S., RAHMAN, S. A. A non-linear feedback neural network for solution of quadratic programming problems. *International Journal of Computer Applications*, 2012, vol. 39, no. 2, p. 44 - 48.

[17] ANSARI, M. S., RAHMAN, S. A. A DVCC-based non-linear analog circuit for solving linear programming problems. In *International Conference on Power, Control and Embedded Systems (ICPCES)*. Allahabad (India), 2010, p. 1 - 4.

[18] WEN, U.-P., LAN, K.-M., SHIH, H.-S. A review of Hopfield neural networks for solving mathematical programming problems. *European Journal of Operational Research*, 2009, vol. 198, no. 3, p. 675 - 687.

[19] KENNEDY, M. P., CHUA, L. O. Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, 1988, vol. 35, no. 5, p. 554 - 562.

[20] RODRIGUEZ-VAZQUEZ, A., RUEDA, A., HUERTAS, J. L., DOMINGUEZ-CASTRO, R. Switched-capacitor neural networks for linear programming. *Electronics Letters*, 1988, vol. 24, no. 8, p. 496 - 498.

[21] RODRIGUEZ-VAZQUEZ, A., DOMINGUEZ-CASTRO, R., RUEDA, A., HUERTAS, J. L., SANCHEZ-SENENCIO, E. Nonlinear switched capacitor 'neural' networks for optimization problems. *IEEE Transactions on Circuits and Systems*, 1990, vol. 37, no. 3, p. 384 - 398.

[22] LAN, K.-M., WEN, U.-P., SHIH, H.-S., LEE, E. S. A hybrid neural network approach to bilevel programming problems. *Applied Mathematics Letters*, 2007, vol. 20, no. 8, p. 880 - 884.

[23] MAA, C.-Y., SHANBLATT, M. A. Linear and quadratic programming neural network analysis. *IEEE Transactions on Neural Networks*, 1992, vol. 3, no. 4, p. 580 - 594.

[24] CHONG, E. K. P., HUI, S., ZAK, S. H. An analysis of a class of neural networks for solving linear programming problems. *IEEE Transactions on Automatic Control*, 1999, vol. 44, no. 11, p. 1995 - 2006.

[25] ZHU, X., ZHANG, S., CONSTANTINIDES, A. G. Lagrange neural networks for linear programming. *Journal of Parallel and Distributed Computing*, 1992, vol. 14, no. 3, p. 354 - 360.

[26] XIA, Y., WANG, J. Neural network for solving linear programming problems with bounded variables. *IEEE Transactions on Neural Networks*, 1995, vol. 6, no. 2, p. 515 - 519.

[27] MALEK, A., YARI, A. Primal-dual solution for the linear programming problems using neural networks. *Applied Mathematics and Computation*, 2005, vol. 167, no. 1, p. 198 - 211.

[28] GHASABI-OSKOEI, H., MALEK, A., AHMADI, A. Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems. *Computers & Mathematics with Applications*, 2007, vol. 53, no. 9, p. 1439 - 1454.

[29] RAHMAN, S. A., JAYADEVA, DUTTA ROY, S. C. Neural network approach to graph colouring. *Electronics Letters*, 1999, vol. 35, no. 14, p. 1173 - 1175.

[30] NEWCOMB, R. W., LOHN, J. D. Analog VLSI for neural networks. *The Handbook of Brain Theory and Neural Networks*, p. 86 - 90. Cambridge (MA, USA): MIT Press, 1998.

[31] ANSARI, M. S., RAHMAN, S. A. A novel current-mode non-linear feedback neural circuit for solving linear equations. In *International Conference on Multimedia, Signal Processing and Communication Technologies*. Aligarh (India), 2009, p. 284 - 287.

About Authors...

Syed Atiqur RAHMAN received the B.Sc.(Eng.) in Electrical Engineering in 1988, and M.Sc.(Eng.) in Electronics and Communication in 1994, from Aligarh Muslim University, Aligarh. He was appointed as Lecturer in the Department of Electronics Engineering at AMU, Aligarh in 1988. He joined as Reader in Computer Engineering and then Electronics Engineering in the same University in 1997. He earned his PhD from IIT Delhi, India, in 2007 and is currently working as an Associate Professor in the Department of Electronics Engineering at AMU. His fields of interest are electronic circuits, logic theory and artificial neural networks.

Mohd. Samar ANSARI received B.Tech. degree in Electronics Engineering from the Aligarh Muslim University, Aligarh, India, in 2001 and M.Tech. degree in Electronics Engineering, with specialization in Electronic Circuit and System Design, in 2007. He is currently an Assistant Professor in the Department of Electronics Engineering, Aligarh Muslim University, where he teaches Electronic Devices & Circuits and Microelectronics. His research interests include microelectronics, analog signal processsing and neural networks. He has published around 50 research papers in reputed international journals & conferences and authored 3 book chapters.

Athar Ali MOINUDDIN received the B.Sc.(Eng.) in Electronics Engineering, M.Sc.(Eng.) in Communication & Information Sytems and Ph.D. from Aligarh Muslim University, Aligarh. He was appointed as Associate Professor in Electronics Engineering Department in the same University in 2009. His fields of interest are communications systems and artificial neural networks.