Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematical Modelling 31 (2007) 78–92
APPLIED MATHEMATICAL MODELLING
www.elsevier.com/locate/apm

A novel approach based on recurrent neural networks applied to nonlinear systems optimization

Ivan Nunes da Silva *, Wagner Caradori do Amaral, Lucia Valeria de Arruda

Department of Electrical Engineering, São Paulo State University, UNESP, CP 473, CEP 17033.360 Bauru/SP, Brazil

Received 1 November 2003; received in revised form 1 June 2005; accepted 2 August 2005
Available online 3 October 2005

Abstract

This paper presents an efficient approach based on recurrent neural network for solving nonlinear optimization. More specifically, a modified Hopfield network is developed and its internal parameters are computed using the valid subspace technique. These parameters guarantee the convergence of the network to the equilibrium points that represent an optimal feasible solution. The main advantage of the developed network is that it treats optimization and constraint terms in different stages with no interference with each other. Moreover, the proposed approach does not require specification of penalty and weighting parameters for its initialization. A study of the modified Hopfield model is also developed to analyze its stability and convergence. Simulation results are provided to demonstrate the performance of the proposed neural network.

Keywords: Nonlinear optimization problems; Recurrent neural networks; Hopfield networks; Nonlinear programming

1. Introduction

Artificial neural networks are richly connected networks of simple computational elements modeled on biological processes. These networks have been applied to several classes of optimization problems and have shown promise for solving such problems efficiently. Unlike most of the numerical optimization methods, one of the main reasons to use neural networks in nonlinear optimization is that they can readily be implemented in hardware. Therefore, approaches based on neural networks are powerful tools for solving mathematical optimization problems so that they could then be solved in real time processes.

Mathematical optimization problems have been widely applied in practically all knowledge areas. The constrained nonlinear optimization plays a fundamental role in many problems involved with the areas of sciences and engineering, where a set of design parameters is optimized subject to inequality and/or equality constraints [1].

* Corresponding author. Tel.: +55 14 3103 6115.
E-mail address: ivan@feb.unesp.br (I.N. da Silva).

0307-904X/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2005.08.007

In the neural networks literature, there exist several approaches used for solving nonlinear optimization problems. The first neural approach applied in optimization problems was proposed by Tank and Hopfield in [2], where the network was used for solving linear programming problems. Although the equilibrium point of the Tank and Hopfield network may not be a solution of the original problem, this seminal work has inspired many researchers to investigate other neural networks for solving linear and nonlinear programming problems [3]. Kennedy and Chua in [4] extended the Tank and Hopfield network by developing a neural network for solving nonlinear programming problems. However, the network of Kennedy and Chua, as well as that proposed by Tank and Hopfield, contains penalty parameters that generate approximate solutions only when these parameters were large.

More recently, it was proposed in [5] a recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints. Another neural model for solving convex quadratic programming problems was presented in [6]. In [7], it was developed a multilayer perceptron for nonlinear programming, which transforms constrained optimization problems into a sequence of unconstrained ones by incorporating the constraint functions into the objective function of the unconstrained problem. In [3], a projection neural network with a single-layer structure has also been proposed for solving nonlinear programming problems. Basically, most of these neural networks proposed in the literature for solving nonlinear optimization problems contain some penalty or weighting parameters. The stable equilibrium points of these networks, which represent a solution of the constrained optimization problem, are obtained only when those parameters are properly adjusted, and in this case, both the accuracy and the convergence process can be affected.

In this paper, we have developed a modified Hopfield network not depending on penalty or weighting parameters, which overcomes shortcomings of the previous approaches. This approach has already been applied to solve several other types of optimization problems. In [8], the proposed approach was efficiently used to solve dynamic programming problems. In [9], the modified Hopfield network was applied to solve combinatorial optimization problems. Another approach using the modified Hopfield network presented this paper was also applied in [10] for mapping of robust parameter estimation problems in presence of unknown-bounded-but errors.

Differently of most of the other neural models, the network proposed here is able to treat several kinds of optimization problems using a unique network architecture, such as those referenced in [8–10], which have respectively applied to solve dynamic programming, combinatorial optimization and robust estimation. In this paper, we have proposed a novel extension of the modified Hopfield network in order to treat constrained nonlinear optimization problems.

For this purpose, the organization of the present paper is as follows. In Section 2, the modified Hopfield network is presented. Section 3 contains a convergence analysis associated to the network dynamics on the context of the valid subspace. In Section 4, a mapping of nonlinear optimization problems is formulated using the modified Hopfield network. Simulation results are presented in Section 5 to demonstrate the advanced performance of the proposed approach. In Section 6, the main results of the paper are summarized and some concluding remarks are presented.

2. The modified Hopfield network

An artificial neural network (ANN) is a dynamic system that consists of highly interconnected and parallel nonlinear processing elements that shows extreme efficiency in computation. In this paper, a modified Hopfield network with equilibrium points representing a solution of the constrained optimization problem has been developed. As introduced in [11], Hopfield networks are single-layer networks with feedback connections between nodes. In the standard case, the nodes are fully connected, i.e., every node is connected to all others nodes, including itself. The node equation for the continuous-time network with n-neurons is given by

$$\dot{u}_i(t) = -\eta \cdot u_i(t) + \sum_{j=1}^{n} T_{ij} \cdot v_j(t) + i_i^b, \tag{1}$$

$$v_i(t) = g(u_i(t)), \tag{2}$$

where $u_i(t)$ is the current state of the ith neuron, $v_j(t)$ is the output of the jth neuron, i_i^b is the offset bias of the ith neuron, $\eta \cdot u_i(t)$ is a passive decay term, and T_{ij} is the weight connecting the jth neuron to ith neuron.

In Eq. (2), $g(u_i(t))$ is a monotonically increasing threshold function that limits the output of each neuron to ensure that network output always lies in or within a hypercube. It is shown in [11] that the equilibrium points of the network correspond to values $\boldsymbol{v}(t)$ for which the following energy function associated to the network is minimized:

$$E(t) = -\frac{1}{2}\boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{T} \cdot \boldsymbol{v}(t) - \boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{i}^b. \tag{3}$$

The mapping of constrained optimization problems using a Hopfield network consists of determining the weight matrix $\boldsymbol{T}$ and the bias vector $\boldsymbol{i}^b$ to compute equilibrium points. A modified energy function $E^{\mathrm{m}}(t)$ is used here, which is defined as follows:

$$E^{\mathrm{m}}(t) = E^{\mathrm{conf}}(t) + E^{\mathrm{op}}(t), \tag{4}$$

where $E^{\mathrm{conf}}(t)$ is a confinement term that groups all the constraints imposed by the problem, and $E^{\mathrm{op}}(t)$ is an optimization term that conducts the network output to the equilibrium points. Thus, the minimization of $E^{\mathrm{m}}(t)$ of the modified Hopfield network is conducted in two stages:

(i) Minimization of the term $E^{\mathrm{conf}}(t)$:

$$E^{\mathrm{conf}}(t) = -\frac{1}{2}\boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{T}^{\mathrm{conf}} \cdot \boldsymbol{v}(t) - \boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{i}^{\mathrm{conf}}, \tag{5}$$

where $\boldsymbol{v}(t)$ is the network output, $\boldsymbol{T}^{\mathrm{conf}}$ is weight matrix and $\boldsymbol{i}^{\mathrm{conf}}$ is bias vector belonging to E^{conf}. This corresponds to confinement of $\boldsymbol{v}(t)$ into a valid subspace [10,12] that confines the inequality and equality constraints imposed by the problem.

(ii) Minimization of the term $E^{\mathrm{op}}(t)$:

$$E^{\mathrm{op}}(t) = -\frac{1}{2}\boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{T}^{\mathrm{op}} \cdot \boldsymbol{v}(t) - \boldsymbol{v}(t)^{\mathrm{T}} \cdot \boldsymbol{i}^{\mathrm{op}}, \tag{6}$$

where $\boldsymbol{T}^{\mathrm{op}}$ is weight matrix and $\boldsymbol{i}^{\mathrm{op}}$ is bias vector belonging to E^{op}. This moves $\boldsymbol{v}(t)$ towards an optimal solution (equilibrium points).

Thus, as shown in Fig. 1, the operation of the modified Hopfield network consists of three main steps, which are defined as follows:

Step I: Minimization of E^{conf}, corresponding to the projection of $\boldsymbol{v}(t)$ in the valid subspace defined by

$$\boldsymbol{v}(t+1) = \boldsymbol{T}^{\mathrm{conf}} \cdot \boldsymbol{v}(t) + \boldsymbol{i}^{\mathrm{conf}}, \tag{7}$$

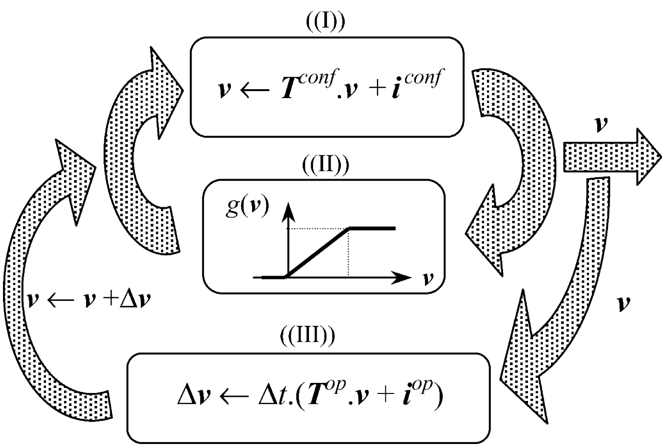

Fig. 1. The modified Hopfield network.

where $\boldsymbol{T}^{\text{conf}}$ is a projection matrix ($\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{T}^{\text{conf}} = \boldsymbol{T}^{\text{conf}}$) and $\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{i}^{\text{conf}} = \mathbf{0}$. This operation corresponds to an indirect minimization process of $E^{\text{conf}}(t)$ using orthogonal projection of $\boldsymbol{v}(t)$ on the feasible set.

Step II: Application of a piecewise activation function constraining $\boldsymbol{v}(t)$ in a hypercube

$$g(v_i) = \begin{cases} \lim\limits_i^{\inf}, & \text{if} \quad v_i < \lim\limits_i^{\inf}, \\ v_i, & \text{if} \quad \lim\limits_i^{\inf} \leqslant v_i \leqslant \lim\limits_i^{\sup}, \\ \lim\limits_i^{\sup}, & \text{if} \quad v_i > \lim\limits_i^{\sup}, \end{cases} \tag{8}$$

where $v_i(t) \in [\lim_i^{\inf}, \lim_i^{\sup}]$. Although $\boldsymbol{v}$ is inside a set with particular structure, the modified Hopfield network can represent a general problem. For example, if $\boldsymbol{v} \in \Re^n$, then $\lim_i^{\inf} = -\infty$ and $\lim_i^{\sup} = \infty$.

Step III: Minimization of E^{op}, which involves updating of $\boldsymbol{v}(t)$ in direction to an optimal solution (defined by $\boldsymbol{T}^{\text{op}}$ and $\boldsymbol{i}^{\text{op}}$), which corresponds to network equilibrium point that represents a solutions for the constrained optimization problem, by applying the gradient in relation to the energy term E^{op}, i.e.,

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t} &= \dot{\boldsymbol{v}} = -\frac{\partial E^{\text{op}}(t)}{\partial \boldsymbol{v}} \\ \Delta\boldsymbol{v} &= -\Delta t \cdot \nabla E^{\text{op}}(\boldsymbol{v}) = \Delta t \cdot (\boldsymbol{T}^{\text{op}} \cdot \boldsymbol{v} + \boldsymbol{i}^{\text{op}}). \end{aligned} \tag{9}$$

Therefore, minimization of E^{op} consists of updating $\boldsymbol{v}(t)$ in the opposite direction of the gradient of E^{op}. These results are also valid when sigmoid-type activation functions are used [13].

As seen in Fig. 1, each iteration represented by the above steps has two distinct stages. First, as described in Step III, $\boldsymbol{v}$ is updated using the gradient of the term E^{op} alone. Second, after each updating given in Step III, $\boldsymbol{v}$ is projected directly in the valid subspace. This second stage is an iterative process, in which $\boldsymbol{v}$ is first orthogonally projected in the valid subspace by applying Step I and then thresholded in Step II, so that its elements lie in the range defined by $[\lim_i^{\inf}, \lim_i^{\sup}]$.

Thus, the mapping of constrained optimization problems using a modified Hopfield network consists of determining the matrices $\boldsymbol{T}^{\text{conf}}$ and $\boldsymbol{T}^{\text{op}}$, and also the vectors $\boldsymbol{i}^{\text{conf}}$ and $\boldsymbol{i}^{\text{op}}$.

In Fig. 2, is illustrated the schematic diagram representing a modified Hopfield network composed by n neurons.

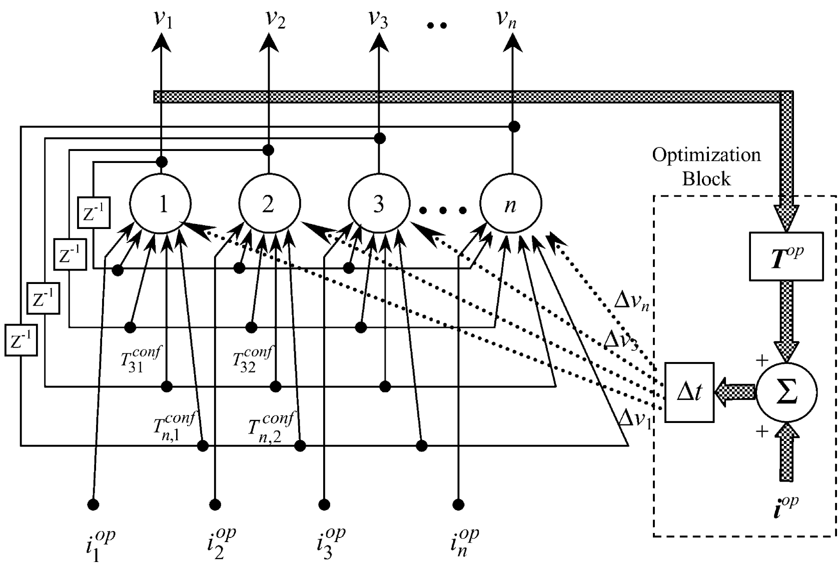

Fig. 2. Schematic diagram of the modified Hopfield network.

3. Analysis of the modified Hopfield network dynamics

The operation of the modified Hopfield network in the early stages of convergence is analyzed in this section. Initially, the component of $\dot{\boldsymbol{v}}$ {defined in Eq. (9)}, which lies in the valid subspace, is defined as $\dot{\boldsymbol{v}}^{\text{conf}}$. Since $\boldsymbol{v}$ is confined to the valid subspace (i.e., $\boldsymbol{v} = \boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{v} + \boldsymbol{i}^{\text{conf}}$), any component of $\dot{\boldsymbol{v}}$ orthogonal to $\dot{\boldsymbol{v}}^{\text{conf}}$ is continually suppressed [14]; hence it is $\dot{\boldsymbol{v}}^{\text{conf}}$, not $\dot{\boldsymbol{v}}$, which best characterizes the overall dynamics of the network. Thus, we have

$$\begin{aligned}\dot{\boldsymbol{v}}^{\text{conf}} &= \boldsymbol{T}^{\text{conf}} \cdot \dot{\boldsymbol{v}} = \boldsymbol{T}^{\text{conf}}(\boldsymbol{T}^{\text{op}}.\boldsymbol{v} + \boldsymbol{i}^{\text{op}}) \\ &= \boldsymbol{T}^{\text{conf}}(\boldsymbol{T}^{\text{op}}(\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{v} + \boldsymbol{i}^{\text{conf}}) + \boldsymbol{i}^{\text{op}}) \\ &= \boldsymbol{T}^{\text{conf}}\boldsymbol{T}^{\text{op}}\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{v} + \boldsymbol{T}^{\text{conf}}(\boldsymbol{T}^{\text{op}} \cdot \boldsymbol{i}^{\text{conf}} + \boldsymbol{i}^{\text{op}}). \end{aligned} \tag{10}$$

The component $\dot{\boldsymbol{v}}^{\text{conf}}$ consists of two parts, a constant term, $\boldsymbol{T}^{\text{conf}}(\boldsymbol{T}^{\text{op}} \cdot \boldsymbol{i}^{\text{conf}} + \boldsymbol{i}^{\text{op}})$, and a term which depends on $\boldsymbol{v}$, $\boldsymbol{T}^{\text{conf}}\boldsymbol{T}^{\text{op}}\boldsymbol{T}^{\text{conf}}\boldsymbol{v}$. These expressions can be simplified by

$$\boldsymbol{T}^{\text{conf}}\boldsymbol{T}^{\text{op}}\boldsymbol{T}^{\text{conf}} = \boldsymbol{A}, \tag{11}$$

$$\boldsymbol{T}^{\text{conf}}(\boldsymbol{T}^{\text{op}} \cdot \boldsymbol{i}^{\text{conf}} + \boldsymbol{i}^{\text{op}}) = \boldsymbol{b}, \tag{12}$$

which gives

$$\dot{\boldsymbol{v}}^{\text{conf}} = \boldsymbol{A} \cdot \boldsymbol{v} + \boldsymbol{b} = \boldsymbol{A} \cdot \boldsymbol{v}^{\text{conf}} + \boldsymbol{b}, \tag{13}$$

where $\boldsymbol{v}^{\text{conf}} = \boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{v}$ and $\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{T}^{\text{conf}} = \boldsymbol{T}^{\text{conf}}$.

With $\boldsymbol{v}$ confined to the valid subspace (i.e., $\boldsymbol{v} = \boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{v} + \boldsymbol{i}^{\text{conf}}$ and $\boldsymbol{T}^{\text{conf}} \cdot \boldsymbol{i}^{\text{conf}} = \boldsymbol{0}$), E^{op} can be expressed as

$$E^{\text{op}} = -\frac{1}{2}\boldsymbol{v}^{\text{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{v} - \boldsymbol{b}^{\text{T}} \cdot \boldsymbol{v}. \tag{14}$$

It is thus apparent, that the dynamics of $\{\dot{\boldsymbol{v}} = \dot{\boldsymbol{v}}^{\text{conf}} = \boldsymbol{A} \cdot \boldsymbol{v} + \boldsymbol{b}\}$ simply results in the steepest descent in relation to E^{op} within the valid subspace, which is consistent with the goal of finding a valid solution which minimizes E^{op}. The general solution of (13) can be expressed by means of the matrix exponential [15]

$$\boldsymbol{v}^{\text{conf}}(t) = \mathrm{e}^{\boldsymbol{A}t}\boldsymbol{v}_0^{\text{conf}} + \int_0^t \mathrm{e}^{\boldsymbol{A}(t-\tau)}\boldsymbol{b} \cdot \mathrm{d}\tau, \tag{15}$$

where $\boldsymbol{v}_0^{\text{conf}}$ is the value of $\boldsymbol{v}^{\text{conf}}$ at time $t = 0$, usually set to be a small random vector. Rewriting the matrix exponentials as power series gives

$$\begin{aligned}\boldsymbol{v}^{\text{conf}}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!}\boldsymbol{A}^k\boldsymbol{v}_0^{\text{conf}} + \int_0^t \sum_{k=0}^{\infty} \frac{(t-\tau)^k}{k!}\boldsymbol{A}^k\boldsymbol{b} \cdot \mathrm{d}\tau = \sum_{k=0}^{\infty} \frac{t^k}{k!}\boldsymbol{A}^k\boldsymbol{v}_0^{\text{conf}} + \sum_{k=0}^{\infty} \frac{\boldsymbol{A}^k\boldsymbol{b}}{k!}\int_0^t (t-\tau)^k\,\mathrm{d}\tau \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!}\boldsymbol{A}^k\boldsymbol{v}_0^{\text{conf}} + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!}\boldsymbol{A}^k\boldsymbol{b}. \end{aligned} \tag{16}$$

To analyze the behavior of $\boldsymbol{v}^{\text{conf}}$ during the convergence process of the network, let us consider the vectors $\boldsymbol{v}^{\text{conf}}$, $\boldsymbol{v}_0^{\text{conf}}$ and $\boldsymbol{b}$ written as terms of their components express on coordinate space generated from normalized eingenvectors of $\boldsymbol{A}$. Thus, suppose $\boldsymbol{A}$ has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, with associated normalized eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$. It is necessary to distinguish between the zero and nonzero eigenvalues of $\boldsymbol{A}$, so the set Z is defined, such as $\lambda_i = 0$ for $i \in Z$ and $\lambda_i \neq 0$ for $i \notin Z$. Let $\boldsymbol{v}^{\text{conf}}$, $\boldsymbol{v}_0^{\text{conf}}$ and $\boldsymbol{b}$ be decomposed along the eigenvectors of $\boldsymbol{A}$ as follows:

$$\boldsymbol{v}^{\text{conf}} = \sum_{i=1}^{n} v_i\boldsymbol{u}_i, \quad \boldsymbol{v}_0^{\text{conf}} = \sum_{i=1}^{n} o_i\boldsymbol{u}_i, \quad \boldsymbol{b} = \sum_{i=1}^{n} b_i\boldsymbol{u}_i, \tag{17}$$

where v_i, o_i and b_i refer respectively to the values of the ith component of $\boldsymbol{v}^{\text{conf}}$, $\boldsymbol{v}_0^{\text{conf}}$ and $\boldsymbol{b}$, represented on coordinate space generated from eigenvectors of $\boldsymbol{A}$. Therefore, from (17), we obtain

$$\boldsymbol{A}^k\boldsymbol{v}_0^{\text{conf}} = \sum_{i=1}^{n} o_i\lambda_i^k\boldsymbol{u}_i, \quad \boldsymbol{A}^k\boldsymbol{b} = \sum_{i=1}^{n} b_i\lambda_i^k\boldsymbol{u}_i.$$

Substituting above equation in (16) results in

$$\begin{aligned}
\boldsymbol{v}^{\text{conf}}(t) &= \sum_{k=0}^{\infty}\frac{t^k}{k!}\sum_{i=1}^{n} o_i\lambda_i^k\boldsymbol{u}_i + \sum_{k=0}^{\infty}\frac{t^{k+1}}{(k+1)!}\sum_{i=1}^{n} b_i\lambda_i^k\boldsymbol{u}_i \\
&= \sum_{i=1}^{n} o_i\boldsymbol{u}_i\sum_{k=0}^{\infty}\frac{t^k\lambda_i^k}{k!} + \sum_{i\notin Z}\frac{b_i\boldsymbol{u}_i}{\lambda_i}\sum_{k=0}^{\infty}\frac{t^{k+1}\lambda_i^{k+1}}{(k+1)!} + \sum_{i\in Z} b_i\boldsymbol{u}_i\sum_{k=0}^{\infty}\frac{t^{k+1}0^k}{(k+1)!} \\
&= \sum_{i=1}^{n} \mathrm{e}^{\lambda_i t} o_i\boldsymbol{u}_i + \sum_{i\notin Z}\frac{b_i\boldsymbol{u}_i}{\lambda_i}\left(\sum_{k=0}^{\infty}\frac{t^k\lambda_i^k}{k!} - 1\right) + \sum_{i\in Z} b_i\boldsymbol{u}_i t \\
&= \sum_{i=1}^{n} \mathrm{e}^{\lambda_i t} o_i\boldsymbol{u}_i + \sum_{i\notin Z}\frac{b_i\boldsymbol{u}_i}{\lambda_i}(\mathrm{e}^{\lambda_i t} - 1) + \sum_{i\in Z} b_i\boldsymbol{u}_i t.
\end{aligned} \tag{18}$$

Eq. (18) is completely general in that it holds for any arbitrary $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{v}_0^{\text{conf}}$. However, the expression can be simplified if $\boldsymbol{A}$ and $\boldsymbol{b}$ are defined as in Eqs. (11) and (12). In this case the eigenvectors of $\boldsymbol{A}$ with corresponding zero eigenvalues are confined to spanning the invalid subspace, whilst $\boldsymbol{b}$ lies wholly within the valid subspace, so $b_i = 0$ for $i \in Z$. Eq. (18) can sequentially be transformed in

$$\boldsymbol{v}^{\text{conf}}(t) = \sum_{i=1}^{n} \mathrm{e}^{\lambda_i t} o_i\boldsymbol{u}_i + \sum_{i\notin Z}\frac{b_i\boldsymbol{u}_i}{\lambda_i}(\mathrm{e}^{\lambda_i t} - 1). \tag{19}$$

It is important to examine the expression for $\boldsymbol{v}^{\text{conf}}(t)$ given by (19) in the limits of small and large t. For small t, we make the approximation following:

$$\mathrm{e}^{\lambda_i t} \approx 1 + \lambda_i t,$$

which gives

$$\boldsymbol{v}^{\text{conf}}(t) \approx \sum_{i=1}^{n} [o_i(1+\lambda_i t) + b_i t]\cdot\boldsymbol{u}_i. \tag{20}$$

Further noting that for a small random $\boldsymbol{v}_0^{\text{conf}}$, the terms o_i are often small in comparison with the b_i. Thus, Eq. (20) transforms in

$$\boldsymbol{v}^{\text{conf}}(t) \approx t\sum_{i=1}^{n} b_i\boldsymbol{u}_i = t\cdot\boldsymbol{b}. \tag{21}$$

So it is apparent that $\boldsymbol{v}^{\text{conf}}$ initially takes off in the direction of the vector $\boldsymbol{b}$. In the limit of large t, Eq. (19) indicates that $\boldsymbol{v}^{\text{conf}}$ will tend towards the eigenvector of $\boldsymbol{A}$ corresponding to the largest positive eigenvalue. In this case, the equilibrium point of the network may be iteratively computed, since the state of the network starting from an arbitrary initial position will converge to the equilibrium point, which is bounded by the hypercube defined by the piecewise activation function defined in (8).

4. Formulation of constrained optimization problems through the modified Hopfield architecture

Consider the following constrained optimization problem, with m-constraints and n-variables, given by the following equations:

$$\text{Minimize}\quad E^{\text{op}}(\boldsymbol{v}) = f(\boldsymbol{v}), \tag{22}$$

$$\text{subject to}\quad E^{\text{conf}}(\boldsymbol{v}):\ h_i(\boldsymbol{v}) \leqslant \boldsymbol{0},\quad i\in\{1\ldots m\}, \tag{23}$$

$$\boldsymbol{z}^{\min} \leqslant \boldsymbol{v} \leqslant \boldsymbol{z}^{\max}, \tag{24}$$

where $\boldsymbol{v}, \boldsymbol{z}^{\min}, \boldsymbol{z}^{\max} \in \Re^n$; $f(\boldsymbol{v})$ and $h_i(\boldsymbol{v})$ are continuous, and all first and second order partial derivatives of $f(\boldsymbol{v})$ and $h_i(\boldsymbol{v})$ exist and are continuous. The vectors $\boldsymbol{z}^{\min}$ and $\boldsymbol{z}^{\max}$ define the bounds on the variables belonging to the vector $\boldsymbol{v}$. The conditions in (23) and (24) define the bounds on the feasible region. The vector $\boldsymbol{v}$ must remain within this region if it is to represent a valid solution for the optimization problem (22). A solution can be obtained by a modified Hopfield network whose valid subspace guarantees the satisfaction of condition (23). Moreover, the initial hypercube represented by the inequality constraints in (24) is directly implemented by the piecewise function given in (8), which is used as neural activation function.

The parameters $\boldsymbol{T}^{\text{conf}}$ and $\boldsymbol{i}^{\text{conf}}$ are calculated by transforming the inequality constraints in (23) into equality constraints by introducing a slack variable $\boldsymbol{w} \in \Re^m$ for each inequality constraint

$$h_i(\boldsymbol{v}) + \sum_{j=1}^{m} \delta_{ij} \cdot w_j = 0, \tag{25}$$

where w_j are slack variables, treated as the variables v_i, and δ_{ij} is defined by

$$\delta_{ij} = \begin{cases} 1, & \text{if} \quad i = j, \\ 0, & \text{if} \quad i \neq j. \end{cases} \tag{26}$$

After this transformation, the problem defined by Eqs. (22)–(24) can be rewritten as

$$\text{Minimize} \quad E^{\text{op}}(\boldsymbol{v}^+) = f(\boldsymbol{v}^+), \tag{27}$$

$$\text{subject to} \quad E^{\text{conf}}(\boldsymbol{v}):\ \boldsymbol{h}^+(\boldsymbol{v}^+) = \boldsymbol{0}, \tag{28}$$

$$z_i^{\min} \leqslant v_i^+ \leqslant z_i^{\max}, \quad i \in \{1 \ldots n\}, \tag{29}$$

$$0 \leqslant v_i^+ \leqslant z_i^{\max}, \quad i \in \{n+1 \ldots N\}, \tag{30}$$

where $N = n + m$, and $\boldsymbol{v}^{+\text{T}} = [\boldsymbol{v}^{\text{T}}\ \ \boldsymbol{w}^{\text{T}}] \in \Re^N$ is a vector of extended variables. It is important to observe that E^{op} does not depend on the slack variables $\boldsymbol{w}$. In [16], has been shown that a projection matrix to the system defined in (28) is given by

$$\boldsymbol{T}^{\text{conf}} = \boldsymbol{I} - \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}} \cdot (\nabla \boldsymbol{h}(\boldsymbol{v}^+) \cdot \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}})^{-1} \cdot \nabla \boldsymbol{h}(\boldsymbol{v}^+), \tag{31}$$

where

$$\nabla \boldsymbol{h}(\boldsymbol{v}^+) = \begin{bmatrix} \frac{\partial h_1(\boldsymbol{v}^+)}{\partial v_1^+} & \frac{\partial h_1(\boldsymbol{v}^+)}{\partial v_2^+} & \cdots & \frac{\partial h_1(\boldsymbol{v}^+)}{\partial v_N^+} \\ \frac{\partial h_2(\boldsymbol{v}^+)}{\partial v_1^+} & \frac{\partial h_2(\boldsymbol{v}^+)}{\partial v_2^+} & \cdots & \frac{\partial h_2(\boldsymbol{v}^+)}{\partial x_N^+} \\ \vdots & \vdots & \ddots & \\ \frac{\partial h_m(\boldsymbol{v}^+)}{\partial v_1^+} & \frac{\partial h_m(\boldsymbol{v}^+)}{\partial v_2^+} & & \frac{\partial h_m(\boldsymbol{v}^+)}{\partial v_N^+} \end{bmatrix} = \begin{bmatrix} \nabla \boldsymbol{h}_1(\boldsymbol{v}^+)^{\text{T}} \\ \nabla \boldsymbol{h}_2(\boldsymbol{v}^+)^{\text{T}} \\ \vdots \\ \nabla \boldsymbol{h}_m(\boldsymbol{v}^+)^{\text{T}} \end{bmatrix}. \tag{32}$$

Inserting the value of (31) in the expression of the valid subspace in (7), we have the following equation:

$$\boldsymbol{v}^+ = [\boldsymbol{I} - \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}} \cdot (\nabla \boldsymbol{h}(\boldsymbol{v}^+) \cdot \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}})^{-1} \cdot \nabla \boldsymbol{h}(\boldsymbol{v}^+)] \cdot \boldsymbol{v}^+ + \boldsymbol{i}^{\text{conf}}. \tag{33}$$

Results of the Lyapunov stability theory [15] can be used in order to develop a deeper understanding of the equilibrium condition. By the definition of Jacobean, when $\boldsymbol{v}$ leads to the equilibrium point $\boldsymbol{v}^{\text{e}} = \boldsymbol{0}(\|\boldsymbol{i}^{\text{conf}}\| \to \boldsymbol{0})$, $\boldsymbol{h}(\boldsymbol{v})$ may be approximated as follows:

$$\boldsymbol{h}(\boldsymbol{v}^+) \approx \boldsymbol{h}(\boldsymbol{v}^{\text{e}}) + \boldsymbol{J} \cdot (\boldsymbol{v}^+ - \boldsymbol{v}^{\text{e}}), \tag{34}$$

where $\boldsymbol{J} = \nabla \boldsymbol{h}(\boldsymbol{v}^+)$.

In the proximity of the equilibrium point $\boldsymbol{v}^{\text{e}} = \boldsymbol{0}$, we obtain the following equation:

$$\lim_{\boldsymbol{v} \to \boldsymbol{v}^e} \frac{\|\boldsymbol{h}(\boldsymbol{v}^+)\|}{\|\boldsymbol{v}^+\|} = 0. \tag{35}$$

Finally, introducing (34) and (35) in equation given by (33), it is obtained the following iterative expression:

$$\boldsymbol{v}^+ \leftarrow \boldsymbol{v}^+ - \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}} \cdot (\nabla \boldsymbol{h}(\boldsymbol{v}^+) \cdot \nabla \boldsymbol{h}(\boldsymbol{v}^+)^{\text{T}})^{-1} \cdot \boldsymbol{h}(\boldsymbol{v}^+). \tag{36}$$

Thus, the expression defined by Step I in Fig. 1 is similar to the above equation.

The parameters $\boldsymbol{T}^{\mathrm{op}}$ and $\boldsymbol{i}^{\mathrm{op}}$ in this case are such that the vector $\boldsymbol{v}^{+}$ is updated in the opposite gradient direction that of the energy function E^{op}. Since conditions (23) and (24) define a bounded region, the objective function (22) has minimum points. Thus, the equilibrium points of the network, which represent those minimum points, can be calculated by assuming the following values to $\boldsymbol{T}^{\mathrm{op}}$ and $\boldsymbol{i}^{\mathrm{op}}$:

$$\boldsymbol{i}^{\mathrm{op}} = -\begin{bmatrix} \frac{\partial f(\boldsymbol{v}^{+})}{\partial v_1^{+}} & \frac{\partial f(\boldsymbol{v}^{+})}{\partial v_2^{+}} & \cdots & \frac{\partial f(\boldsymbol{v}^{+})}{\partial v_N^{+}} \end{bmatrix}^{\mathrm{T}}, \tag{37}$$

$$\boldsymbol{T}^{\mathrm{op}} = \boldsymbol{0}. \tag{38}$$

To demonstrate the advanced behavior of the modified Hopfield network derived in this section and to validate its properties, some examples involving nonlinear optimization problems have been simulated in the next section.

5. Simulation results

In this section, the modified Hopfield network proposed in previous sections has been used to solve constrained optimization problems. We provide four examples to illustrate the effectiveness of the proposed architecture.

Example 1. Consider the following constrained optimization problem proposed in [16], which is composed by inequality constraints and bounded variables:

$$\begin{aligned} \text{Minimize}\quad & f(\boldsymbol{v}) = \mathrm{e}^{v_1} + v_1^2 + 4v_1 + 2v_2^2 - 6v_2 + 2v_3,\\ \text{subject to}\quad & v_1^2 + \mathrm{e}^{v_2} + 6v_3 \leqslant 15,\\ & v_1^4 - v_2 + 5v_3 \leqslant 25,\\ & v_1^3 + v_2^2 - v_3 \leqslant 10,\\ & 0 \leqslant v_1 \leqslant 4,\\ & 0 \leqslant v_2 \leqslant 2,\\ & v_3 \geqslant 0. \end{aligned}$$

This problem has a unique optimal solution $\boldsymbol{v}^* = [0.0\ 1.5\ 0.0]^{\mathrm{T}}$, and the minimal value of $f(\boldsymbol{v}^*)$ at this point is equal to -3.5. To observe the global convergent behavior of the proposed network, we generated 15 initial points randomly distributed between 0 and 5. The bound constraints represented by the last three equations are directly mapped through the piecewise activation function defined in (8). All simulation results obtained by the modified Hopfield network show that the proposed architecture is globally asymptotically stable at $\boldsymbol{v}^*$. Fig. 3 shows the trajectories of the modified Hopfield network starting from $\boldsymbol{v}^0 = [2.33\ 0.31\ 0.16]^{\mathrm{T}}$ and converging towards to optimal vector $\boldsymbol{v}^*$.

The trajectories of the objective function starting from several initial points are illustrated in Fig. 4. All trajectories lead towards the same theoretical minimal value provided by $f(\boldsymbol{v}^*) = -3.5$ when assumed $\Delta t = 0.0001$. These results show the efficiency of the modified Hopfield network for solving constrained nonlinear optimization problems.

To provide a more consistent analysis in relation to the efficiency of the proposed architecture, we make a comparison between the results produced by the modified Hopfield network with those provided by the network developed by Xia et al. in [3] and by the Kennedy and Chua's topology presented in [4] through the following example.

Example 2. Consider the following constrained optimization problem proposed in [3], which is composed by inequality constraints:

$$\begin{aligned} \text{Minimize}\quad & f(\boldsymbol{v}) = v_1^3 + (v_1 - 2v_2)^3 + \mathrm{e}^{v_1+v_2},\\ \text{subject to}\quad & \boldsymbol{v} \in V, \end{aligned}$$

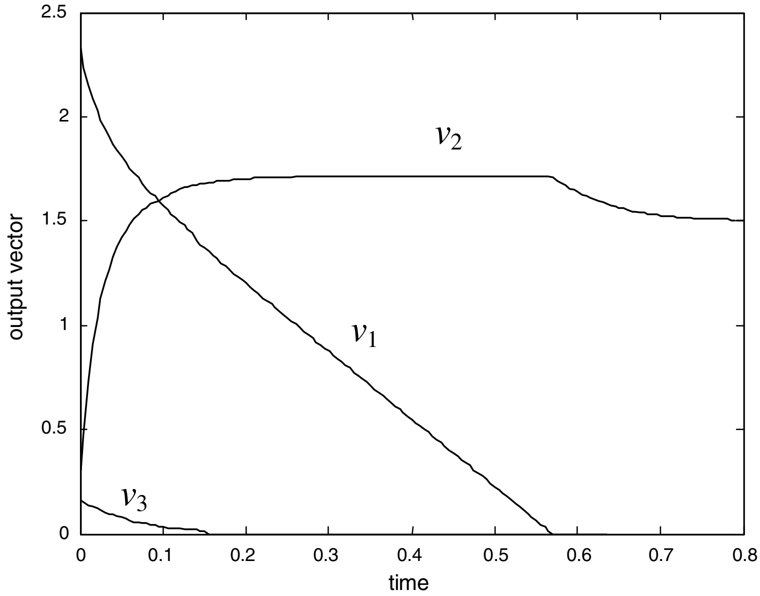

Fig. 3. Transient behavior of the modified Hopfield network in Example 1.

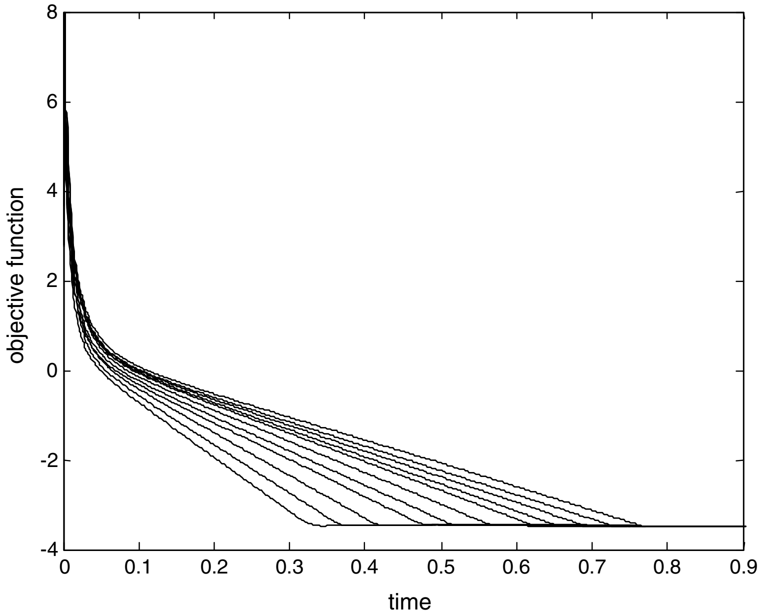

Fig. 4. Evolution of the objective function for 15 initial points in Example 1.

where $V = \{\boldsymbol{v} \in \Re^2 | v_1^2 + v_2^2 \leqslant 1\}$. This problem has a unique optimal solution given by $\boldsymbol{v}^* = [-0.5159 \quad 0.8566]^T$ with $f(\boldsymbol{v}^*) = -9.8075$. All simulation results provided by the modified Hopfield network show that it is globally asymptotically stable at $\boldsymbol{v}^*$.

In Table 1, the results obtained by the modified Hopfield network using $\Delta t = 0.0001$ are compared with those provided by the projection neural network proposed by Xia et al. in [3], and also those given by the nonlinear circuit network developed by Kennedy and Chua in [4]. Six different initial points were chosen, where two points $\{(1,0); (0,-1)\}$ are located in V and four $\{(-2,-2); (2,-2); (2,2); (-2,-2)\}$ are not in V. The

Table 1
Comparison of the simulation results in Example 1

Initial vector	Modified Hopfield Network	Projection neural network	Nonlinear circuit network
$\boldsymbol{v}^{(0)} = [2 \quad 2]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5195 \quad 0.8641]^{\mathrm{T}}$
$\boldsymbol{v}^{(0)} = [-2 \quad 2]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5196 \quad 0.8641]^{\mathrm{T}}$
$\boldsymbol{v}^{(0)} = [-2 \quad -2]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5161 \quad 0.8564]^{\mathrm{T}}$	$\propto$
$\boldsymbol{v}^{(0)} = [1 \quad -1]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5162 \quad 0.8563]^{\mathrm{T}}$	$\propto$
$\boldsymbol{v}^{(0)} = [1 \quad 0]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5162 \quad 0.8564]^{\mathrm{T}}$	$\propto$
$\boldsymbol{v}^{(0)} = [0 \quad -1]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5160 \quad 0.8566]^{\mathrm{T}}$	$\boldsymbol{v} = [-0.5161 \quad 0.8564]^{\mathrm{T}}$	$\propto$

results obtained by the modified Hopfield network are very close to the exact solution. The mean error between the solutions obtained by the network and the exact solution is less than 0.02%. We can verify that all solutions produced by the modified Hopfield network are quite accurate.

According to Table 1 the nonlinear circuit network proposed by Kennedy and Chua can apparently approach $\boldsymbol{v}^*$ in only two cases. This confirmation was also observed in simulations performed in [3]. The projection neural network developed in [3] produces solutions for all cases presented in Table 1, and we can observe that the final solutions depend on their initial values. It is also shown in Table 1 that the modified Hopfield network, independently of the initial values of $\boldsymbol{v}$, has converged to the same final values for all simulations. To illustrate the global convergent behavior of the modified Hopfield network, Fig. 5 shows the trajectories of $\boldsymbol{v}$ starting from several initial points.

It is important to observe that all trajectories starting from the inside or outside of the feasible region V converge to $\boldsymbol{v}^*$. Thus, the proposed approach always converges to the optimal solution, independently whether the chosen initial point is located in the feasible region or not. Therefore, we can conclude that the modified Hopfield network is of high robustness.

Example 3. Consider the following constrained optimization problem proposed in [1], which is composed by inequality and equality constraints:

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{v}) = v_1^3 + 2v_2^2 \cdot v_3 + 2v_3, \\ \text{subject to} \quad & v_1^2 + v_2 + v_3^2 = 4, \\ & v_1^2 - v_2 + 2v_3 \leqslant 2, \\ & v_1, v_2, v_3 \geqslant 0. \end{aligned}$$

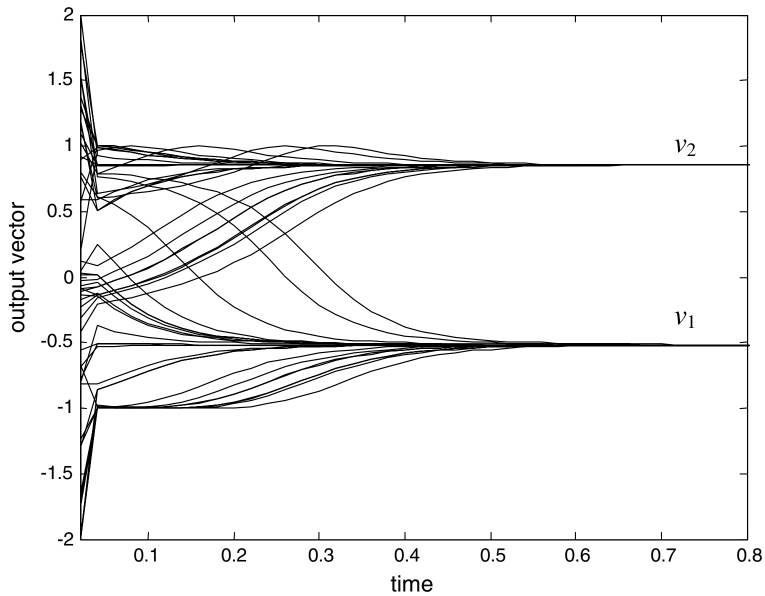

Fig. 5. Trajectories of the modified Hopfield network for 20 initial points in Example 2.

The optimal solution for this problem is given by $\boldsymbol{v}^* = [0.0 \quad 4.0 \quad 0.0]^T$, where the minimal value of $f(\boldsymbol{v}^*)$ at this point is equal to zero. Fig. 6 shows the trajectories of the network variables starting from the initial point $\boldsymbol{v}^0 = [1.67 \quad 1.18 \quad 3.37]^T$. All simulation results obtained by the modified Hopfield network using $\Delta t = 0.001$ show that the proposed architecture is globally asymptotically stable at $\boldsymbol{v}^*$.

The network has also been evaluated for different values of initial conditions. The trajectories of the objective function starting from several initial points are illustrated in Fig. 7. All trajectories lead toward the same

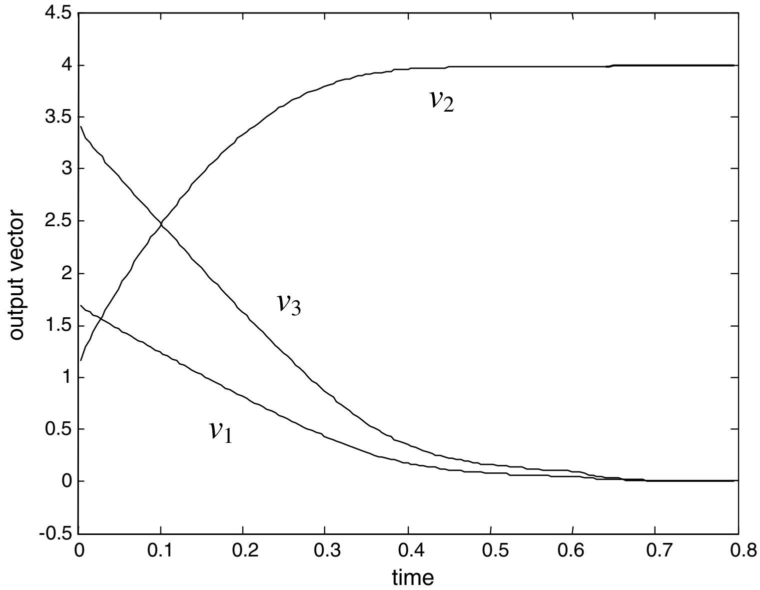

Fig. 6. Transient behavior of the modified Hopfield network in Example 3.

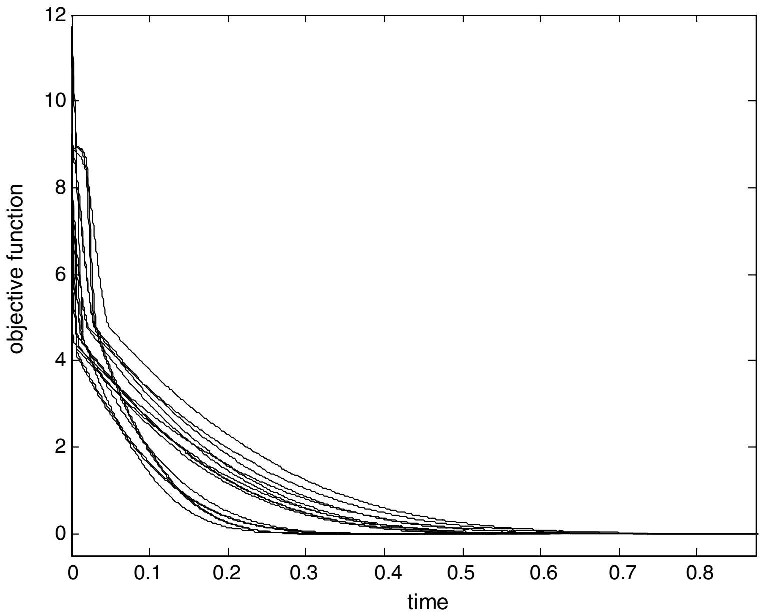

Fig. 7. Evolution of the objective function for 15 initial points in Example 3.

equilibrium point. These results show the ability and efficiency of the modified Hopfield network for solving constrained nonlinear optimization when equality and inequality constraints are simultaneously included in the problem.

In comparison with results obtained by using the multilayer perceptron network proposed in [7], and starting from the same initial points, it was observed that the modified Hopfield Network not only converges more quickly, but also results in higher accuracy.

Example 4. In this example we consider a constrained optimization problem representing the economic power dispatch problem [17].

The economic power dispatch problem consists of determining the optimal combination of power outputs for all generating units, which minimizes the total fuel cost while satisfying the constraints imposed by the system. Considering n generating units, the economic power dispatch problem can be expressed by the following equations:

$$\text{Minimize} \quad f(\boldsymbol{v}) = \sum_{i=1}^{n} f_i(v_i) = \sum_{i=1}^{n} (a_i + b_i \cdot v_i + c_i \cdot v_i^2),$$

$$\text{subject to} \quad \sum_{i=1}^{n} v_i - L - D = 0,$$

$$v_i^{\min} \leqslant v_i \leqslant v_i^{\max},$$

where $f_i(v_i)$ is the fuel cost of the ith generating unit, v_i is the power generated by the ith unit, $v_i^{\min}$ and $v_i^{\max}$ are minimum and maximum generation levels for the ith unit, L is the system total demand, and D is the system loss. The generation parameters are described in Table 2. In this example, the total demand is 850 MW and the system loss is considered despicable [17].

The purpose of this example is also to compare the solutions obtained by the modified Hopfield network with some other neural approaches presented in the literature, which are also used for solving economic power dispatch problems. This problem has a unique optimal solution given by $\boldsymbol{v}^* = [393.17 \quad 334.60 \quad 122.23]^{\mathrm{T}}$. The transient behavior of the generation dispatches obtained by the modified Hopfield network using $\Delta t = 0.001$ is shown in Fig. 8.

The behavior of the cost function is illustrated in Fig. 9. The network has also been evaluated for different values of initial conditions. All trajectories lead towards the same equilibrium point.

The values of the generation dispatches calculated by the modified Hopfield network is shown in Table 3, as well as the dispatches provided by the linear Hopfield network proposed in [18], by the conventional Hopfield method [19] and by the quadratic programming network proposed in [20]. The solution obtained by the modified Hopfield network is equal to the optimal solution.

The simulation results described in Table 3 shows that the modified Hopfield network has presented the best results. From a detailed analysis accomplished on the other models, the aspects that can influence on the their performances are the following: (i) optimization and constraint terms involved in the problem mapping are treated in a single stage, (ii) interference between optimization and constraint terms affects the precision of the equilibrium points, and (iii) the convergence process of the networks depend on the correct adjustment of the weighting constants associated with the energy terms. However, the modified Hopfield

Table 2
Initial parameters of the problem in Example 4

Unit	a_i	b_i	c_i	$v_i^{\min}$ (MW)	$v_i^{\max}$ (MW)
1	561	7.92	0.001562	150	600
2	310	7.85	0.00194	100	400
3	78	7.97	0.00482	50	200

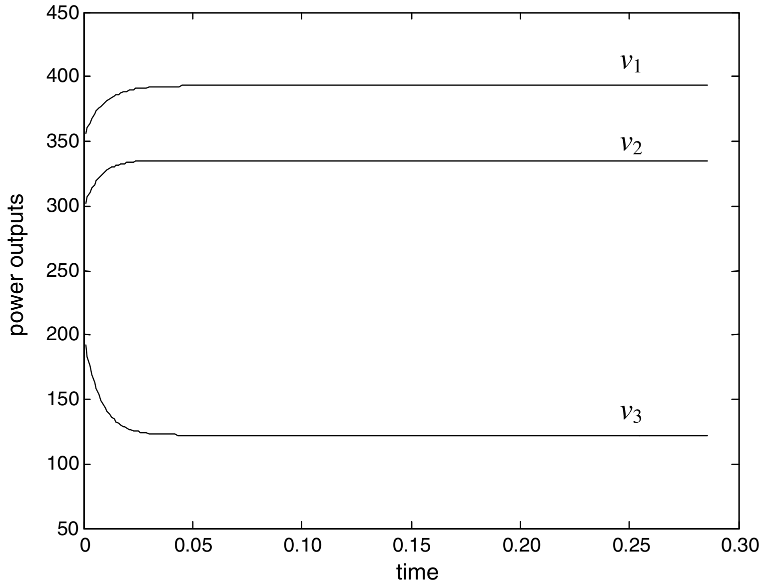

Fig. 8. Transient behavior of the power outputs in Example 4.

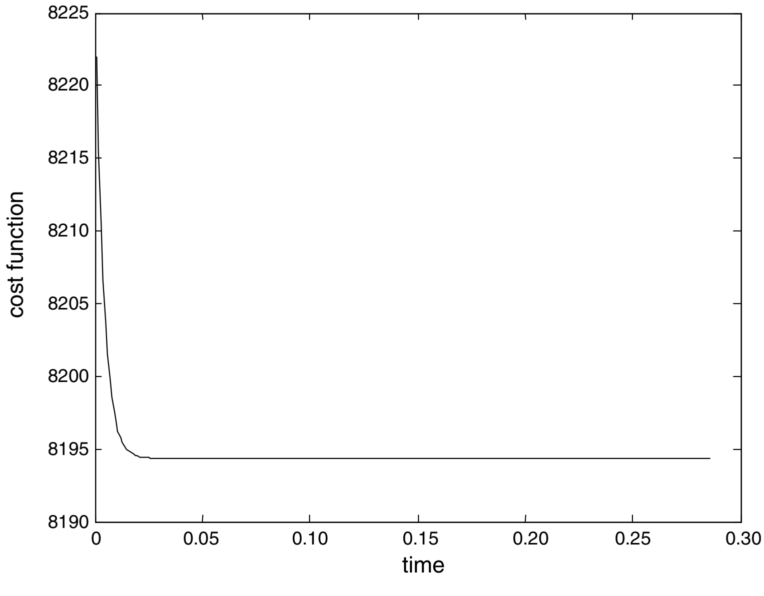

Fig. 9. Evolution of the cost function in Example 4.

Table 3
Comparison of the simulation results in Example 4

Method	v_1 (MW)	v_2 (MW)	v_3 (MW)
Modified Hopfield network	393.17	334.60	122.23
Linear Hopfield network	392.90	334.39	122.14
Conventional Hopfield network	393.80	333.10	122.30
Quadratic programming network	393.10	334.52	122.20
Optimal solution	393.17	334.60	122.23

network presented here has not presented these inconveniences since it treats optimization and constraint terms in different stages.

6. Conclusions

In this paper, we have developed a modified Hopfield network for solving constrained optimization problems. The architecture presented here treats optimization and constraint terms in different stages. The terms $\boldsymbol{T}^{\mathrm{conf}}$ and $\boldsymbol{i}^{\mathrm{conf}}$ (belonging to E^{conf}) of the modified Hopfield network were developed to force the validity of the structural constraints associated with the nonlinear optimization problem, and the terms $\boldsymbol{T}^{\mathrm{op}}$ and $\boldsymbol{i}^{\mathrm{op}}$ (belonging to E^{op}) were developed to find a optimal solution associated with the cost function. In contrast to the other neural approaches used in these types of problems, the modified Hopfield network presented in this paper does not require specification of weighting parameters in its initialization process. Thus, the main advantages of using a modified Hopfield network to solve constrained optimization problems are the following ones: (i) consideration of optimization and constraint terms in distinct stages with no interference with each other, (ii) use of unique energy term (E^{conf}) to group all constraints imposed on the problem, and (iii) lack of need for adjustment of weighting constants for initialization.

The simulation results demonstrate that the proposed network is an efficient method to solve nonlinear optimization problems. From the simulation results we can conclude that the modified Hopfield network proposed in this paper has the advantages of global convergence and high accuracy of solutions. In order to guarantee this global convergent behavior, we also analyzed, for each example presented in the paper, the network convergence from several other points randomly generated. For all simulations, the network always converged to the same optimal solutions.

Some particularities of the neural approach in relation to primal methods normally used in nonlinear optimization are the following: (i) it is not necessary the computation, in each iteration, of the active set of constraints; (ii) the neural approach does not compute Lagrange's multipliers; and (iii) the initial solution used to initialize the network can be outside of the feasible set defined from the constraints.

References

[1] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear Programming, Wiley, New York, 1993.
[2] D.W. Tank, J.J. Hopfield, Simple neural optimization networks: an A/D converter, signal decision network, and a linear programming circuit, IEEE Trans. Circuits Syst. 33 (1986) 533–541.
[3] Y. Xia, H. Leung, J. Wang, A projection neural network and its application to constrained optimization problems, IEEE Trans. Circuits Syst. I 49 (2002) 447–458.
[4] M.P. Kennedy, L.O. Chua, Neural networks for nonlinear programming, IEEE Trans. Circuits Syst. 35 (1988) 554–562.
[5] X.B. Liang, J. Wang, A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints, IEEE Trans. Neural Networks 11 (2000) 1251–1262.
[6] Y.H. Chen, S.C. Fang, Neurocomputing with time delay analysis for solving convex quadratic programming problems, IEEE Trans. Neural Networks 11 (2000) 230–240.
[7] J. Reifman, E.E. Feldman, Multilayer perceptron for nonlinear programming, Comput. Operat. Res. 29 (2002) 1237–1250.
[8] I.N. Silva, L.V.R. Arruda, W.C. do Amaral, An efficient model of neural networks for dynamic programming, Int. J. Syst. Sci. 32 (2001) 715–722.
[9] I.N. Silva, A.N. de Souza, M.E. Bordon, A modified Hopfield model for solving the N-queens problem, in: Proceedings of the IEEE/INNS/ENNS International Joint Conference on Neural Networks, Honolulu, 2000, pp. 509–514.
[10] I.N. Silva, L.V.R. Arruda, W.C. Amaral, A novel approach to robust parameter estimation using neurofuzzy systems, Math. Comput. Simul. 48 (1999) 251–258.
[11] J.J. Hopfield, Neurons with a graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. 81 (1984) 3088–3092.
[12] S.V.B. Aiyer, M. Niranjan, F. Fallside, A theoretical investigation into the performance of the Hopfield model, IEEE Trans. Neural Networks 1 (1990) 53–60.
[13] S. Abe, Global convergence and suppression of spurious states of the Hopfield neural networks, IEEE Trans. Circuits Syst. 40 (1993) 246–257.
[14] A.H. Gee, S.V.B. Aiyer, R.W. Prager, Analytical framework for optimization neural networks, Neural Networks 6 (1993) 79–97.
[15] M. Vidyasagar, Nonlinear Systems Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[16] D.G. Luenberger, Linear and Nonlinear Programming, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
[17] A.F. Wood, B.F. Wollenberg, Power Generation Operation and Control, Wiley, New York, 1984.

[18] C.T. Su, G.J. Chiou, A fast-computation Hopfield method to economic dispatch of power systems, IEEE Trans. Power Syst. 12 (1997) 1759–1764.
[19] J.H. Park, Y.S. Kim, I.K. Eom, K.Y. Lee, Economic load dispatch for piecewise quadratic cost function using Hopfield neural network, IEEE Trans. Power Syst. 8 (1993) 1030–1038.
[20] C.Y. Maa, M.A. Shanblatt, Linear and quadratic programming neural network analysis, IEEE Trans. Neural Networks 3 (1992) 580–594.