Advances in Environmental Biology, 8(21) October 2014, Pages: 1345-1354

AENSI Journals

Advances in Environmental Biology

ISSN-1995-0756 EISSN-1998-1066

Journal home page: http://www.aensiweb.com/AEB/

A New Ranking Model in DEA Based on Euclidean DistanceandSolving It with Neural Network Approach

[1]A. Ghomashi, [2]F. Hosseinzadeh Lotfi, [3]G.R. Jahanshahloo

[1,2,3]Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

ARTICLE INFO

Article history:

Received 11 July 2014

Received in revised form 26 September 2014

Accepted 3 December 2014

Available online 22 December 2014

Keywords:

Recurrent neural network, Data envelopment analysis, Ranking, Stability, Global convergence.

ABSTRACT

One of the important concept in Data Envelopment Analysis (DEA) is ranking. A drawback in almost of existing ranking models is inability to rank non-extreme units. In this paper, we proposed a model to rank extreme and non-extreme DMUs based on Euclidean distance from convex combination of full-inefficient DMUs which first presented by Jahanshaloo and Afzalinezhad[G. R. Jahanshahloo and M. Afzalinejad, "A ranking method based on a full-inefficient frontier," Applied Mathematical Modelling, vol.30, no. 3, pp. 248-260, 2006.]. Then, we use a novel neural network to solve proposed model for ranking. The validity of the proposed model for ranking and proposed neural network to solve it are considered by proof of lemma, theorems and solving numerical examples.

To Cite This Article: A. Ghomashi, F. Hosseinzadeh Lotfi, G.R. Jahanshahloo, A New Ranking Model in DEA Based on Euclidean DistanceandSolving It with Neural Network Approach. ***Adv. Environ. Biol., 8(21),*** *1345-1354, 2014*

INTRODUCTION

DEA is a nonparametric approach in operations research first proposed by Charmes et al in 1978 to estimate the performance evaluation and relative efficiency of a set of homogeneous DMUs such as business units, government agencies, police departments, hospitals, educational institutions and etc[17]. The seminal model of Charnes et al was called CCR model. Banker et al in 1984 developed a variable returns to scale version of the CCR model that was called BCC model [18]. DEA successfully divides DMUs into two categories: efficient DMUs and inefficient DMUs. DEA does this by assigning a relative efficiency score to each DMU such that the DMUs in efficient category have identical relative efficiencies equal to one and the rest have the relative efficiencies between zero and one. The researchers proposed some methods to difference DMUs in efficient category which this concept has named Ranking efficient units in the DEA . There are lots of ranking methods and each of them has special quality and property to rank efficient[21]. A drawback of existing models in ranking is inability to rank non-extreme units. A ranking method to rank non-extreme units is Jahanshahloo and Afzalinejad method[20]. They defined a full-inefficient frontier and ranked DMUs based on their distance from this frontier. In this paper,based on their work, we proposed a model to rank extreme and non-extreme DMU based on Euclidean distance from convex combination of full-inefficient DMUs. Proposed model is a special case of quadratic programming with linearconstraints. We solve this model using a novel neural network approach. The neural networks are computing systems composed of a number ofhighly interconnected simple information processing units, when the dimension and deneness of the structure of optimization problem increases thus neural network can usually solve optimization problems in execution times at the orders of magnitude much faster than most popular optimization algorithms for general-purpose digital[5]. From a mathematical point of view, to formulate an optimization problem in terms of a neural network, there exist three types of methods[2], one approach commonly used in developing an optimization neural network is to first convert the constrained optimization problem into an associated unconstrained optimization problem, and then design a neural network that solves the unconstrained problem with gradient methods. second approach is to construct a set of differential equations such that their equilibrium points correspond to the desired solutions and then find an appropriate Lyapunov function such that all trajectory of the systems converges to some equilibrium points. Last approach which proposed by Xia and Wang, combination of the above two type methods[2]. In this paper, to solve our proposed ranking model we are concerned with solving QP problem with lower model complexity than existing ones. Kennedy and chua[3] presented a neural network for solving the strict convex quadratic programming. Lately, many researchers

Corresponding Author: A. Ghomashi, Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

successively proposed a number of models. In [4], Rodriguez-Vazquez et al. proposed a neural network for exactly solving linear and quadratic programming problems. In 1992, Maa and Shanblatt summarized and proved the computational capability of neural networks to LP and QP [5]. Xia [6], Xia and wang,[7-13], presented several neural networks for solving linear and quadratic programming problems, which have proved to be globally convergent to the exact solutions. In this paper, we propose a neural network model with two layer structure for solving our proposed ranking model. This paper is divided into six sections. In next section preliminary information is introduced to facilitate later discussions. In section III, we proposed a quadratic model to rank efficient DMUs in BCC model based on definition of full-inefficient DMUs .In section V, we proposed a neural network model to solve proposed ranking model and we analyze stability condition and global convergence proposed neural network model. In section V, some simulation examples are discussed. Section VI gives the conclusions of this paper.

2 Preliminaries:

2.1 DEA Preliminaries:

Suppose there are n DMUs with m inputs and s outputs. The input and output vectors of $DMU_j\,(j=1,...,n)$ *are* $X_j=(x_{1j},...,x_{mj})^t, Y_j=(y_{1j},...,y_{sj})^t$, respectively, where $X_j\ge 0, X_j\ne 0, Y_j\ge 0, Y_j\ne 0$.

The PPS is represented as

$T=\{(X,Y)\in R^{m+s}\,|Y\ can\ be\ produced\ \ from\ X\}$

Banker et al[18] inferred the following PPS for BCC model:

$$T_{BCC}=\{(X,Y)\in R_+^{m+s}\ |\ X\ge\sum_{j=1}^{n}\lambda_j X_j,\ Y\le\sum_{j=1}^{n}\lambda_j Y_j,\ ,\sum_{j=1}^{n}\lambda_j=1,\ \lambda_j\ge 0\ \ j=1,...,n\}$$

The BCC model is used for an efficiency measure under the condition of variant returns to scale(VRS)[18]. Additive model in T_{BCC} is called the which is as follows[19]:

$$z_{ADD}^*=Max\{\sum_{i=1}^{m}S_i^-+\sum_{r=1}^{s}S_r^+\ |\ \sum_{j=1}^{n}\lambda_j X_j+S^-=X_p,\ \sum_{j=1}^{n}\lambda_j Y_j-S^+=Y_p,\ ,\sum_{j=1}^{n}\lambda_j=1,\ \lambda_j\ge 0\ \ j=1,...,n\} \tag{1}$$

Where $S^-\in R^m\ and\ S^+\in R^s$.

Definition 2.1 DMU_p *is ADD-efficient if and only if* $z_{ADD}^*=0$ *[19].*

Definition 2.2 *if* DMU_p *is ADD-efficient then* DMU_p *is called BCC-efficient otherwise* DMU_p *is called BCC-inefficient[19].*

Definition 2.3 *Reference set of* DMU_p *is defined the following form[19]:*

$Ref_p=\{j\,|\,\lambda_j^*\ne 0\ in\ one\ of\ \ optimal\ solutions of\ (1))\}$

Ref_p helps us to construct a benchmark for DMU_p .

Lemma 2.1 *Let* $k\in Ref_p$ *then* DMU_k *is efficient DMU.*

Proof: see[19]

Definition 2.4 *Suppose* $S\subseteq R^{m+s}$ *as the convex hull of all the DMUs. The following set is called full-inefficient frontier[20].*

$F(S)=\{(X,Y)\in S\,|\,\forall(X^{'},Y^{'})\in R^{m+s}((-X^{'},Y^{'})�(-X,Y)\Rightarrow(X^{'},Y^{'})\in S)\}\subseteq S$

Each DMU belonged to $F(S)$ is called a full-inefficient DMU.

2.2 Functions:

Definition 2.5 *A function* $F:R^m\to R^m$ *is said to be Lipschitz continuous with constant* $L>0$ *[14] if for each pair of points* $x,y\in R^m$

$\|F(x)-F(y)\|\le L\|x-y\|.$

Definition 2.6 *Let* $X=\{x\in R^m\,|\,l_i\le x\le u_i,\forall i\in N-\{1,...,m\}\ \}$*,* $P_X:R^m\to X$ *is called a projection operator and defined the following form[14]:*

$P_X(x)=arg\ min\|x-y\|, y\in X$

Indeed $P_X(x)$ is a projection of vector x to set X . Since X is a box set, the projection operator P_X can be stated by

$P_X(x)=[P_X(x_1),...,P_X(x_m)]^{'}$

where $P_X(x_i)$ is a piecewise function. For $i \in \{1,...,m\} - N, P_X(x_i) = x_i$ and for $i \in N$, $P_X(x_i)$ is defined the following form[14]:

$$P_X(x_i) = \begin{cases} l_i, & x_i < l_i \\ x_i, & l_i \le x_i \le u_i \\ u_i, & x_i > u_i \end{cases}$$

Lemma 2.2 *Let* $X \in R^n$ *be a closed convex set.Then*

$$\|P_X(x) - P_X(y)\| \le \|x - y\|, \quad \forall x, y \in R^n$$

where $P_X(x)$ is a projection operator on X.

Proof. See[14].

Definition 2.7 *The finite-dimensional variational inequality problem* $VI(F,K)$ *is to determine a vector* $x^* \in K \subseteq R^n$*, such that[14]*

$$F(x^*)^T(x - x^*) \ge 0, \qquad \forall x \in K \tag{2}$$

where F is a given continuous function from K to R^m, K is a given closed convex set.

Theorem 2.1 *Assume that* K *is closed and convex. Then* $x^* \in K$ *is a solution of the variational inequality problem* $VI(F,K)$ *if and only if for any* $\gamma > 0$, x^* *is a fixed point of the map*

$$P_K(I - \gamma F) : K \to K$$

that is

$$P_K(x^* - \gamma F(x^*)) = x^* \tag{3}$$

Proof: see[14].

Lemma 2.3 *Let* $U : D \subseteq R^m \to R^m$ *be differentiable on an open convex set* D. *Let* $g : R^m \to R$ *given by*

$$g(x) = \int_0^1 (x - x_0)^T U(x_0 + t(x - x_0))dt$$

If the jacmatrix of $U(x)$ is symmetric for all $x \in D$, then $\nabla g(x)^T = U(x)$ where $\nabla g(x)$ is the gradient of g.

Proof: see[1].

Theorem 2.2 *For all* $y \in R^m$ *and all* $x \in X \subseteq R^m$

$$(y - P_X(y))^T(P_X(y) - x) \ge 0$$

Proof. See[14].

2.3 Differential Equation Preliminaries:

Definition 2.8 *Take the following dynamical system*

$$\dot{x} = f(x(t)), \qquad x(t_0) = x_0 \in R^n \tag{4}$$

where $f : R^n \to R^n$. x^* is called an equilibrium point of (4) if [15]

$$f(x^*) = 0.$$

Theorem 2.3 *Assume that* f *in (4) is a continuous mapping, then for arbitrary* $t_0 \ge 0$ *and* $x_0 \in R^n$ *there exists a local solution* $x(t)$ *to (4) where* $t \in [t_0, \tau]$ *for some* $\tau > t_0$. *Furthermore if* f *is locally Lipschitzian continuous at* x_0 *then the solution is unique, and if* f *is Lipschitzian continuous in* R^n *then* τ *can be extended to* $+\infty$ *[15].*

Theorem 2.4 *Let* x^* *be an equilibrium point of (4) and* $X \subset R^n$ *be an open neighborhood of* x^*, *if* $V : R^n \to R$ *is a continuously differentiable function over* X *and* V *satisfies in the following conditions:*

- $V(x^*) = 0, \quad \dfrac{dV(x^*)}{dt} = 0$
- $\dfrac{dV(x(t))}{dt} \le 0, \; V(x) > 0, \qquad \forall x \in X - \{x^*\}$
- $\|x\| \to \infty \Rightarrow \|V(x)\| \to \infty$

then x^* is a lyapunov stable equilibrium and the solution always exist globally,if

$$\frac{dV(x(t))}{dt} < 0 \quad \forall x \in X - \{x^*\}$$

then x^* is a globally asymptotically stable equilibrium[15].

3 Ranking Model:

Based on full-inefficient DMU definition, we use the following model to identify full-inefficient DMU:

$$z_p^* = Max\{\sum_{i=1}^{m} S_i^- + \sum_{r=1}^{s} S_r^+ \mid \sum_{j=1}^{n} \lambda_j X_j - S^- = X_p,\ \sum_{j=1}^{n} \lambda_j Y_j + S^+ = Y_p,\ ,\sum_{j=1}^{n} \lambda_j = 1, \quad (5)$$
$$\lambda_j \geq 0 \ \ j = 1,...,n\}$$

Where $S^- \in R^m$ *and* $S^+ \in R^s$.

Lemma 3.1 *if* $z_p^* = 0$ *then* DMU_p *is full-inefficient DMU.*

Proof: Based on definition 2.4 proof is complete.

Lemma 3.2 *Each convex combination of full-inefficient DMU is inefficient.*

Proof: Based on definition 2.3 and lemma 2.1 proof is complete.

Let $RI = \{j \in \{1,...,n\} \mid z_j^* = 0\}$ and $RE = \{j \in \{1,...,n\} \mid w_{ADD}^* = 0\}$, we proposed the following model to rank of efficient DMUs:

$$z_p^{**} = Max \left\| \begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} X_p \\ Y_p \end{pmatrix} \right\|_2^2 \qquad p \in RE$$
$$s.t. \ \sum_{j \in RI} \lambda_j X_j = X,$$
$$\sum_{j \in RI} \lambda_j Y_j = Y, \quad (6)$$
$$\sum_{j \in RI} \lambda_j = 1,$$
$$\lambda_j \geq 0 \ \ j = 1,...,n$$

Regarding to (6), if $z_p^{**} > z_q^{**}(p,q \in RI)$ then DMU_p has better performance respect to DMU_q since it has more distance from convex hull of full-inefficient DMUs. From Fig. 1 it can be seen that A, B and C are BCC-efficient and F, G, H are full-inefficient and Z^{**} presents Euclidean distance of $DMUB$ from convex combination of full-inefficient DMUs.

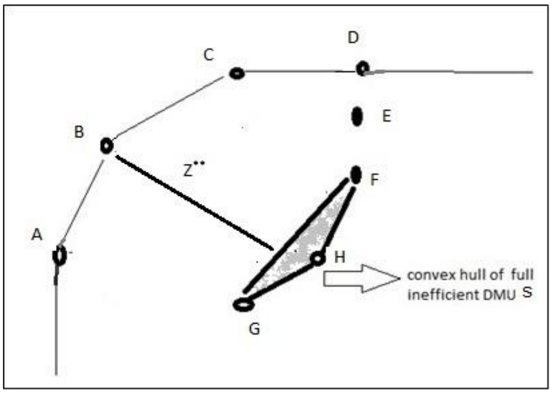

Fig. 1: Geometry illustrative of proposed model for ranking

proposed model(6) is a special case of general form of quadratic programming with linear constraints which matrix in objective function is identity matrix. There exists models to solve it, in next section we proposed a novel neural network model to solve(6) with low complexity.

4 Neural Network Model:

Consider the following QP problem:

$$\begin{aligned} &\mathrm{Min}\ c^T x + \frac{1}{2} x^T x \\ &S.t. \\ &\quad b_1 \le Ax \le b_2 \\ &\quad l \le x \le u \end{aligned} \tag{7}$$

where $c, x \in R^n, A \in R^{m\times n}, b_1$ and $b_2 \in R^m$.

(7) is a special case of general form of quadratic programming which $Q = I_{n\times n}$ so Q in (7) is a symmetric and positive definite matrix. The Lagrangian function of (7) is the following form[16]:

$$L(x, y, u) = c^T x + \frac{1}{2} x^T x - u^T (Ax - y) \tag{8}$$

where $x \in X = \{x \mid l \le x \le u\}$, $y \in Y = \{y \mid b_1 \le y \le b_2\}$ and $u \in R^m$.

Definition 4.1 $(\bar{x}, \bar{y}, \bar{u})$ *is called a saddle point of (8) if* $\bar{x} \in X$, $\bar{y} \in Y$ *and*

$$L(\bar{x}, \bar{y}, u) \le L(\bar{x}, \bar{y}, \bar{u}) \le L(x, y, \bar{u}) \tag{9}$$

for all $x \in X$ and $y \in Y$ [16].

Lemma 4.1 $(\bar{x}, \bar{y}, \bar{u})$ *is a saddle point of (8) if and only if* $\bar{x}$ *are optimal solutions of (7).*

Proof: see[16].

substituting (8) into (9), we have

$$\begin{aligned} c^T \bar{x} + \frac{1}{2} (\bar{x})^T \bar{x} - u^T (A\bar{x} - \bar{y}) &\le c^T \bar{x} + \frac{1}{2} (\bar{x})^T \bar{x} - (\bar{u})^T (A\bar{x} - \bar{y}) \\ &\le c^T x + \frac{1}{2} x^T x - (\bar{u})^T (Ax - y) \end{aligned} \tag{10}$$

for all $x \in X$, $y \in Y$ and $u \in R^m$. From the first inequality in (10) we have

$$(u - \bar{u})^T (A\bar{x} - \bar{y}) \ge 0, \quad \forall u \in R^m \tag{11}$$

so we have

$$A\bar{x} = \bar{y} \tag{12}$$

Define

$$f(x) = c^T x + \frac{1}{2} x^T x - (\bar{u})^T Ax \tag{13}$$

From second inequality in (10) and definition of $f(x)$ we have

$$f(\bar{x}) - f(x) \le (\bar{u})^T (y - \bar{y}) \qquad \forall x \in X, \ \ \forall y \in Y \tag{14}$$

from the above inequality we have

$$f(\bar{x}) - f(x) \le 0, \quad \forall x \in X \tag{15}$$

since if there exist a $\tilde{x} \in X$ such that $f(\bar{x}) - f(\tilde{x}) > 0$ then we have $(\bar{u})^T (y - \bar{y}) > 0, \forall y \in Y,$ which is contradictive when $y = \bar{y}$, so we have

$$(\bar{u})^T (y - \bar{y}) \le 0, \quad \forall y \in Y \tag{16}$$

Using (16), Definition 2.7 and theorem 2.1, we have $(\bar{y}, \bar{u})$ is the solution the following equation:

$$y = P_Y (y - u) \tag{17}$$

On the other hand (15) implies that $\bar{x} \in X$ minimizer of $f(x)$ thus

$$\nabla f(\bar{x}) = \bar{x} + c - A^T \bar{u} = 0 \tag{18}$$

(12),(17) and (18) imply that $\bar{x} \in X$ is optimal solution of (7) if and only if there exist $\bar{y}$ and $\bar{u}$ such thatsatisfying the following equations:

$$\begin{aligned} &Ax - y = 0 \\ &P_Y (y - u) - y = 0 \\ &x + c - A^T u = 0 \end{aligned} \tag{19}$$

By substituting first equation into second equation in (19) we have

$$\begin{aligned} &P_Y (Ax - u) - Ax = 0 \\ &x + c - A^T u = 0 \end{aligned} \tag{20}$$

Lemma 4.2 $\bar{x} \in X$ *is optimal solution of (7) if and only if there exist* $\bar{u}$ *such that* $(\bar{x}, \bar{u})$ *satisfies in (20).*

Proof: Using lemma 4.1 and (21), proof is complete.

In this, Using lemma 4.2, we proposed the following recurrent neural network for solving (7):

• state equation

$$\frac{du}{dt} = \lambda(P_Y(Ax - u) - Ax) \tag{21}$$

• output equation

$$x = P_X(A^T u - c) \tag{22}$$

where $\gamma > 0$ is scalar and . The system described by Eq. (21) can be applied for solving (7) and can be easily realized by a recurrent neural network. The proposed neural network can be implemented by using a simple hardware only without analog multipliers for the variables or the penalty parameter. The operator P_Y *and* P_X may be implemented by using a piecewise activation function.

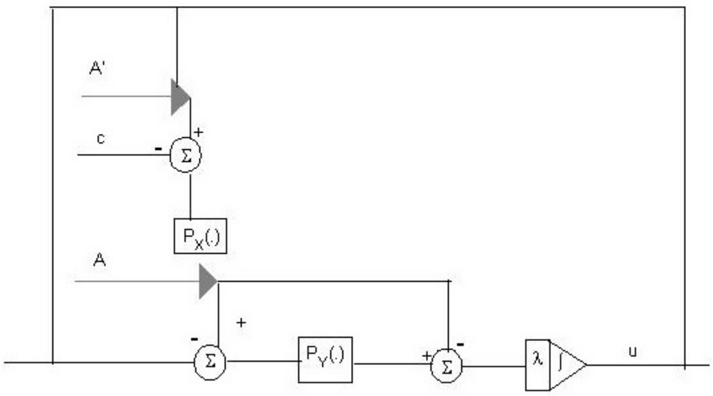

Fig. 2: Architecture of the recurrent neural network (21).

Corollary 4.1:

Right hand side of (21a) is Lipschitz continuous function.

Proof. Let the right hand side of (21a) be denoted by $L(u)$ and $h(u) = x = P_X(A^T u - c)$ by lemma 2.2, for any $\hat{u}$ and $\bar{u}$,we have:

$$\begin{aligned}
&\|L(\hat{u}) - L(\bar{u})\| \\
&= \left\|\lambda\left(P_Y(Ah(\hat{u}) - \hat{u}) - Ah(\hat{u})\right) - \lambda\left(P_Y(Ah(\bar{u}) - \bar{u}) - Ah(\bar{u})\right)\right\| \\
&= \left\|\lambda\left(P_Y(Ah(\hat{u}) - \hat{u}) - P_Y(Ah(\bar{u}) - \bar{u}) + Ah(\bar{u}) - Ah(\hat{u})\right)\right\| \\
&\le \left\|\lambda\left(Ah(\hat{u}) - \hat{u} + \bar{u} - Ah(\bar{u}) + Ah(\bar{u}) - Ah(\hat{u})\right)\right\| \\
&= \lambda\|(\hat{u} - \bar{u})\|
\end{aligned}$$

Which gives the desired results.

Lemma 4.3 *For each initial point* $u(t_0) \in R^m$*, there exist a unique continuous solution* $u(t)$ $(t \in [t_0, \tau))$ *for (21a) and the equilibria of neural network in (21a) correspond to unique optimal solution of (7).*

Proof: Theorem 2.3 and corollary 4.1 yield a unique continuous solution $u(t)$ over $[t_0, \tau)$ exist for (21a). Since the feasible set of problem(7) is nonempty and objective function in (7) is strictly convex function so there exista unique optimal solution for (7), then using lemma 4.2 and (21a), proof is complete.

5 Convergence Analysis:

In this section, we prove globally asymptotically convergent of (21a).

Definition 5.1 *The neural network in (21a) is said to be stable in the sense of Lyapunov and globally convergent, globally asymptotically stable, if the corresponding dynamic system is so.[1]*

Theorem 5.1 *The proposed neural network in (21a) is stable in the sense of Lyapunov and is globally asymptotically convergent to the unique solution of (7). Moreover, the convergence rate of the neural network in (21a) increase as* λ *increases.*

Proof. By Lemma 4.3, we know that for each initial point $u(t_0) \in R^m$, there exist a unique continuous solution $u(t)$ $(t \in [t_0, \tau))$ for (21a). Let u^* be equilibria point of (21a),define a lyapunov function below:

$$V(u(t)) = \int_0^1 (u(t) - u^*) I(u^* + s(u(t) - u^*)) ds, \quad \forall t \ge t_0, \tag{23}$$

where $I(.)$ is identity function. Let $u = u(t)$, then time derivative of V along the trajectory of (21a) as follows:

$$\frac{d}{dt} V(u) = \frac{dV}{du} \frac{du}{dt} \tag{24}$$

on the other hand by theorem 2.2 we have:

$$(y - P_Y(y))^T (P_Y(y) - u) \ge 0, \quad \forall y \in R^m, \forall u \in Y$$

Let $y = Ax - u$ and $u = Ax$ then

$$(Ax - u - P_Y(Ax - u))^T (P_Y(Ax - u) - Ax) \ge 0$$

where $x = P_X(A^T u - c)$ the above inequality rewritten as follows:

$$I(u)^T (P_Y(Ax - u) - Ax) \le -\|P_Y(Ax - u) - Ax\|^2$$

since the jacobian matrix of $I(u)$ in (22) is symmetric, form lemma2.3 we have

$$\frac{dV}{du} = I(u)$$

Then

$$\frac{dV}{du} \frac{du}{dt} = \lambda I(u)^T (P_Y(Ax - u) - Ax) \le -\|P_Y(Ax - u) - Ax\|^2 \le 0$$

so we have

$$\{u(t) \mid t_0 \le t < \tau\} \subset P_0 = \{u \in R^m \mid V(u) \le V(u(t_0))\}$$

since P_0 is bounded and $\{u(t) \mid t_0 \le t < \tau\} \subset P_0$, $u(t)$ is bounded and thus $\tau = \infty$. Moreover

$$\frac{dV(u)}{dt} = 0 \Leftrightarrow P_Y(Ax - u) - Ax = 0 \Leftrightarrow \frac{du}{dt} = 0 \; (25)$$

so by applying the theorem 2.4 and lemma 4.2, we get result that the proposed neural network is globally asymptotically convergent to the unique solution of (7).

Since $\frac{dV}{dt} < 0$ then we can result that as λ increases, the convergence rate of the neural network in (21a) increases. This proof is completed.

6 Illustrative Examples:

In this section, we demonstrate the effectiveness and performance of the proposed model for ranking and proposed neural network to solve it with two illustrative examples. The ordinary differential equation solver engaged in ode23 in matlab 2012.

Example 1 *Fifteen DMUs were evaluated in terms of three inputs* (x_1, x_2, x_3) *and three output* (y_1, y_2, y_3) *that defined in table 1.*

Table 1: Data set in Example 1

DMUs	x_1	x_2	x_3	y_1	y_2	y_3
1	225935.0	405.0	1000.0	178285.0	4967.0	3388.0
2	102200.0	179.0	575.0	75526.0	3808.0	2083.0
3	94900.0	135.0	388.0	63868.0	5435.0	1246.0
4	51100.0	95.0	263.0	31835.0	1222.0	559.0
5	43800.0	43.0	186.0	19360.0	1112.0	373.0
6	31755.0	36.0	116.0	23372.0	2095.0	275.0
7	31390.0	28.0	99.0	17798.0	462.0	277.0
8	29930.0	31.0	159.0	14067.0	794.0	184.0
9	27740	41	226	15895	1130	386
10	23360.0	39.0	153.0	18127.0	1143.0	269.0
11	20440.0	31.0	94.0	14860.0	949.0	157.0
12	10585.0	15.0	67.0	4498.0	148.0	29.0
13	8030.0	15.0	62.0	8311.0	335.0	70.0
14	11680.0	26.0	110.0	9449.0	435.0	161.0
15	11315.0	20.0	92.0	4797.0	71.0	53.0

The results of proposed ranking model, non radial AP model[21](nr-AP) and jahanshahloo-afzalinejad ranking model[20](JA) is provided in table 2. Fig.3 shows that transient behavior of the neural network of (21a) in terms of $w(t)$. As can be seen from Fig. 3, the proposed neural network model is globally convergent to the optimal solution.

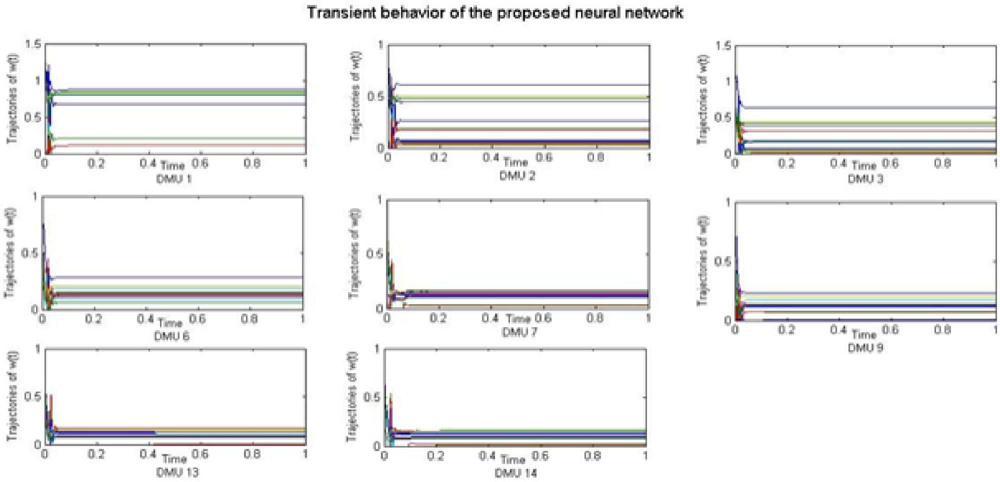

Fig. 3: Transient behavior of the neural network of (21a) in terms of $w(t)$ in example 1

Table 2: Comparison of three models in Example 1

DMUs	Rank based on		
	our model	nr-AP	JA
1	1	2	8
2	3	1	7
3	2	3	6
6	4	5	5
7	7	6	4
9	8	4	3
13	5	7	2
14	6	8	1

Example 2 *The inputs and outputs of seven DMUs which each DMU consumes two inputs* (x_1, x_2) *to produce four outputs* (y_1, y_2, y_3, y_4) *is presented in table 3.*

Table 3: Data set in Example 2

DMUs	x_1	x_2	y_1	y_2	y_3	y_4
1	2.0	2.0	2.0	2.0	2.0	2.0
2	2.0	2.0	2.0	3.0	2.0	2.0
3	2.0	2.0	2.0	2.0	3.0	2.0
4	2.0	2.0	2.0	2.0	2.0	3.0
5	1.0	1.0	2.0	2.5	3.5	2.0
6	1.0	3.0	2.0	2.0	2.0	4.0
7	2.0	1.0	2.0	2.5	2.25	3.0

The results of running the proposed ranking model, (nr-AP) and (JA) are summarized in table 4. Fig. 4 shows that transient behavior of the neural network of (9) in terms of $w(t)$ which is globally convergent to the optimal solution.

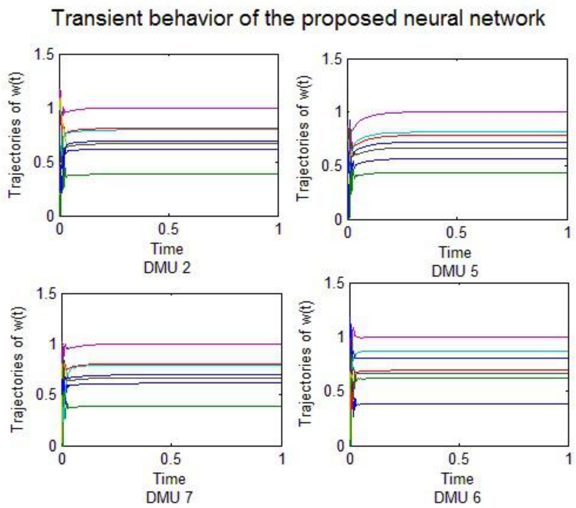

Fig. 4: Transient behavior of the neural network of (21a) in terms of $w(t)$ in example 2

Table 4: Comparison of three models in Example 2

DMUs	Rank based on		
	our model	nr-AP	JA
2	3	4	4
5	1	3	3
6	4	1	2
7	2	2	1

Conclusion:

In this paper, a new quadratic model introduced to rank non-extreme and extreme efficient units in DEA. Proposed model is based on full-inefficient unit in DEA. Then ,we proposed a novel neural network model to solve quadratic programming with linear constraints which in objective function,Q=I, and apply to solve proposed quadratic model. The neural network model in this paper has low complexity respect to semilar existing model. It is shown here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to the optimal solutions. Finally, examples are provided to show the effectiveness of the proposed model for ranking and neural network model to solve proposed quadratic model

REFERENCES

[1] Xia, Y.S., H. Leung and J. Wang, 2002. A projection neural network and its application to constrained optimization problems, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 49(4): 447-458.
[2] Xia, Y.S. and J. Wang, 1998. A general methodology for designing globally convergent optimization neural networks, IEEE Trans. Neural Networks, 9(6): 1331-1343.
[3] Kennedy, M.P. and L.O. Chua, 1988. Neural networks for nonlinear programming, IEEE Trans. Circuits Syst., 35: 554-562.
[4] Rodr´ýguez-V´azquez, A., R. Dom´ýnguez-Castro, A. Rueda, J.L. Huertas and E. S´anchez-Sinencio, 1990. Nonlinear switched-capacitor 'neural networks' for optimization problems, IEEE Trans. Circuits Syst., 37: 384-397.
[5] Maa, C.Y. and M.A. Shanblatt, 1992. Linear and quadratic programming neural network analysis, IEEE Trans. Neural Networks, 3: 580-594.
[6] Xia, Y., 1996. A new neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Networks, 7: 1544-1547.
[7] Wu, X., Y. Xia, J. Li and W.K. Chen, 1996. A high performance neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Networks, 7: 643-651.
[8] Xia, Y. and J. Wang, 2000. A recurrent neural network for solving linear projection equations, Neural Networks, 13: 337-350.
[9] Xia, Y. and J. Wang, 2000. Global exponential stability of recurrent neural networks for solving optimization and related problems, IEEE Trans. Neural Networks, 11: 1017-1022.
[10] Xia, Y.S and J. Wang, 2000. On the stability of globally projected dynamic systems, J. Optim. Theory Appl., 106(1): 129-150.

[11] Xia, Y.S and J. Wang, 2004. A general projection neural network for solving monotone variational inequality and related optimization problems, IEEE Trans. Neural Networks, 15(2): 318-328.
[12] Xia, Y.S and J. Wang, 2005. A recurrent neural network for solving nonlinear convex programs subject to linear constraints, IEEE Trans. Neural Networks, 16(2): 1045-1053.
[13] Xia, Y., G. Feng, J. Wang, 2004. A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations.Neural Netw, 17(7): 1003-1015.
[14] Kinderlehrer, D., G. Stampacchia, 1980. An Introduction to Variational Inequalities and Their Applica- tions, Academic Press, New York.
[15] Zabczyk, J., 1992. Mathematical control theory: an introduction, Birkhuser, Boston.
[16] Bazaraa, M.S. and C. Shetty, 1979. Nonlinear programming, theory and algorithms, Jon wiley and sons, New york.
[17] Charnes, A., W.W. Cooper and E. Rhodes, 1978. Measuring the efficiency of decision making units. European Journal of Operational Research, 2: 429-444.
[18] Banker, R.D., A.Charnes and W.W. Cooper, 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science, 30(9): 1078-1092.
[19] Cooper, W.W., L.Seiford and K.Tone, 2005. Introduction to Data Envelopment Analysis and its Uses: With DEA-Solver Software and References.New York: Springer.
[20] Jahanshahloo, G.R. and M. Afzalinejad, 2006."A ranking method based on a full-inefficient frontier," Applied Mathematical Modelling, 30(3): 248-260.
[21] Hosseinzadeh Lotfi, F., G.R. Jahanshahloo, M. Khodabakhshi, M. Rostamy-Malkhlifeh, Z. Moghaddas and M. Vaez-Ghasemi, 2013." A Review of Ranking Models in Data Envelopment Analysis", Journal of Applied Mathematics.