Neural Processing Letters
https://doi.org/10.1007/s11063-019-10068-y

Projection Recurrent Neural Network Model: A New Strategy to Solve Weapon-Target Assignment Problem

Alireza Shojaeifard[1] · Ali Nakhaei Amroudi[1] · Amin Mansoori[2] · Majid Erfanian[3]

Abstract
In the present research, we are going to obtain the solution of the Weapon-Target Assignment (WTA) problem. According to our search in the scientific reported papers, this is the first scientific attempt for resolving of WTA problem by projection recurrent neural network (RNN) models. Here, by reformulating the original problem to an unconstrained problem a projection RNN model as a high-performance tool to provide the solution of the problem is proposed. In continuous, the global exponential stability of the system was proved in this research. In the final step, some numerical examples are presented to depict the performance and the feasibility of the method. Reported results were compared with some other published papers.

Keywords Weapon-target assignment problem · Nonlinear optimization problem · Projection recurrent neural network · Global exponential stability · Projection function

1 Introduction

The weapon-target assignment (WTA) problem is to find a proper assignment of weapons to targets with the objective of minimizing the expected damage of own-force assets. There are two versions of the WTA problem: the static WTA problem [1] and the dynamic WTA problem [2]. In the static WTA problem, the weapons are launched at the same time, while they are launched asynchronously in the dynamic WTA problem. In this paper, the former

✉ Amin Mansoori
am.ma7676@yahoo.com; a-mansoori@um.ac.ir

Alireza Shojaeifard
ashojaeifard@ihu.ac.ir

Ali Nakhaei Amroudi
kpnakhaei@ihu.ac.ir

Majid Erfanian
erfaniyan@uoz.ac.ir

1 Department of Mathematics and Statistics, Imam Hossein Comprehensive University, Tehran, Iran

2 Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran

3 Department of Science, School of Mathematical Sciences, University of Zabol, Zabol, Iran

Published online: 22 June 2019

is investigated. This is an NP-complete problem [3,4]. Some methods such as combinatorial optimization [5], pseudo-boolean programming [6] (and the references therein) are given to solve these types of problems. These methods are based on graph search approaches and usually result in exponential computational complexities. As a consequence, it is difficult to solve these types of problems directly while the number of targets or weapons are large. Genetic algorithms (GAs) have been applied to solve this problem [7–9]. Even though those approaches could find the best solution in those simulated cases, the search efficiency did not seem good enough. The dynamic system method is one of the efficient approaches for solving programming problems. In fact, recurrent neural network (RNN) models are well tool to transfer the optimization problems into a dynamic system. The main idea of such method for solving the mathematical programming problem is to applied a non-negative function which is called energy function and a dynamic system. An important requirement is that the energy function decreases monotonically as the dynamic system approaches an equilibrium point. Note that, the nature of the dynamic system method is parallel and distributed which is the main advantage of this scheme.

The pioneering works on RNN models to optimization are for Hopfield and Tank [10,11]. Neurodynamic optimization has received great success in recent years [12–24]. For instance, a general projection NN model to solve extended linear-quadratic optimization problems with linear constraints was given by Hu [12], Sun et al. [13] developed an other efficient model to solve second-order cone constrained variational inequality (VI), Eshaghnezhad et al. [14] developed an RNN for solving non-linear pseudo-monotone projection equation, Effati et al. [15] gave a projection type NN model to solve bilinear programs, He et al. [16] proposed an inertial projection NN to solve VIs, Mansoori et al. [17,18] presented an efficient NN model to solve the absolute value equations, Cheng et al. [19] proposed a neutral-type delayed projection NN for solving nonlinear variational inequalities, Zhang [20], developed an artificial NN method for the fuzzy shortest path problem, and Liu et al. [21] gave a continuous-time RNNs with time-varying thresholds. Also, projection NNs have also been applied to the field of control engineering, for example, kinematic control of redundant manipulators [25], time-scale expansion-based approximated optimal control [26], robot manipulators [27,28] and mobile robot manipulators [29].

Motivated by the former discussion, in spite of the fact that the several success in RNNs, it has some limits in solving zero-one programs with general convex objective functions. The WTA problem is a zero-one optimization problem with linear constraints. There are not an attempt on WTA problem by RNN. In this research, we try to develop a novel projection RNN for this problem. In fact, this paper is going to show the existence of the RNN model for WTA problem. For this purpose, we assume the unconstrained form of the problem. Then, the novel proposed projection RNN model constructed used to solve the WTA problem. On the final step, the globally exponentially stability of the novel proposed projection RNN is stated.

2 Weapon-Target Assignment Problem Formulation

In this section a mathematical formulation of the WTA problem is presented.

On modern battlefields, it is an important task for battle managers to make a proper WTA to defend own-force assets. As an example in considering anti-aircraft weapons of naval battle force platforms, threat targets may be launched from surface ships, aircrafts, or submarines. These targets have different probabilities of killing to platforms which are dependent on the

Table 1 Notation and variables defined in WTA

	Definition
W	The number of weapon types
T	The number of targets that must be engaged
u_j	The value of target j. It is determined during the threat evaluation phase and used to priorities target engagement
p_{ij}	The probability of destroying target j by a single weapon of type i
x_{ij}	A binary decision variable indicating the number of weapons of type i assigned to target j

target types. Thus, a WTA decision-aided system is strongly desired in helping and training planners to make proper decisions on the battlefield. To formulate the WTA problem, we use some notation and variables defined in Table 1. Throughout the paper, in the WTA problem, we assume that there are W weapons and T targets and all weapons must be assigned to targets. In fact, this assumption is further restricted to $W = T$ and that all targets must also be assigned. Such a restricted assumption, in fact, has largely reduced the search space, but also may restrict the diversity of solutions. Also, the individual probability of killing (p_{ij}) by assigning the ith weapon to the jth target is known for all i and j. This probability defines the effectiveness of the jth weapon to destroy the ith target. Hence, the WTA problem is to minimize the following cost function [30]:

$$F = \sum_{i=1}^{T} u_i \left(\prod_{j=1}^{W} (1 - p_{ij})^{x_{ij}} \right), \tag{1}$$

where $\prod_{j=1}^{W}(1 - p_{ij})^{x_{ij}}$ verifies the overall probability of the ith target being destroyed and u_i is the expected damage value of the ith target to the asset. Since, all weapons must be assigned to targets therefore,

$$\sum_{i=1}^{T} x_{ij} = 1, \quad \text{for } j = 1, 2, \ldots, W, \tag{2}$$

where x_{ij} is a Boolean value indicating whether the jth weapon is assigned to the ith target. $x_{ij} = 1$ indicates that the jth weapon is assigned to the ith target. Thus, Eq. (1) summarizes the overall damage for all targets. Moreover, Eq. (2) can be rewritten as follow when $W = T = N$:

$$\sum_{i=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N,$$
$$\sum_{j=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N.$$

Summarizing the former discussion, the WTA problem can be modelled as the following zero-one nonlinear optimization problem:

$$\begin{aligned} \min \quad & F = \sum_{i=1}^{T} u_i \left(\prod_{j=1}^{W} (1 - p_{ij})^{x_{ij}} \right) \\ s.t. \quad & \sum_{i=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N, \\ & \sum_{j=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N, \\ & x_{ij} \in \{0, 1\}^{N^2}. \end{aligned} \tag{3}$$

Because of the total unimodality property of the constraint coefficient matrix defined in (3) [31], the integrality constraint in the WTA problem formulation can be equivalently replaced with the non-negativity constraint. In fact, in the original problem if a unique optimum exists we can transfer the zero-one constraint with the non-negativity constraint [31]. By the above discussion the following equivalent nonlinear programming problem can be formulated:

$$\begin{aligned} \min \quad & F = \sum_{i=1}^{N} u_i \left(\prod_{j=1}^{N} (1 - p_{ij})^{x_{ij}} \right) \\ s.t. \quad & \sum_{i=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N, \\ & \sum_{j=1}^{N} x_{ij} = 1, \quad i = 1, 2, \ldots, N, \\ & x_{ij} \geq 0. \end{aligned} \tag{4}$$

Next result shows that the cost function F in (4) is convex.

Theorem 2.1 *The function $F : [0, 1]^{N^2} \to \mathbb{R}$ is convex (note that $x_{i,j} \geq 0$ can not exceed from 1).*

Proof Since the sum of convex functions is convex so we just need to indicate that the function $\prod_{j=1}^{N} (1 - p_{ij})^{x_{ij}}$ is convex for every i. Consider the function $f(x) = \prod_{j=1}^{N} (1 - p_{ij})^{x_j}$. Assume that $y, z \in [0, 1]^N$ and also, $0 \leq \lambda \leq 1$. We have,

$$\lambda f(y) + (1 - \lambda) f(z) = f(z) \left(\lambda \frac{f(y)}{f(z)} + (1 - \lambda) \right).$$

Also, we can get:

$$\begin{aligned} f(z) \left(\frac{f(y)}{f(z)} \right)^{\lambda} &= f(z) \frac{f(y)^{\lambda}}{f(z)^{\lambda}} \\ &= f(y)^{\lambda} f(z)^{1-\lambda} \\ &= \left(\prod_{j=1}^{N} (1 - p_{ij})^{y} \right)^{\lambda} \left(\prod_{j=1}^{N} (1 - p_{ij})^{z} \right)^{1-\lambda} \end{aligned}$$

$$= \prod_{j=1}^{N} (1 - p_{ij})^{\lambda y + (1-\lambda) y}$$
$$= f(\lambda y + (1-\lambda) y).$$

Therefore, it yields $f(z) \left(\frac{f(y)}{f(z)}\right)^{\lambda} = f(\lambda y + (1-\lambda) y)$. Furthermore, the function $\left(\frac{f(y)}{f(z)}\right)^{\lambda}$ for very λ is convex and for every $0 \leq \lambda \leq 1$ gives $\left(\frac{f(y)}{f(z)}\right)^{\lambda} \leq \lambda \frac{f(y)}{f(z)} + (1-\lambda)$. Applying the fact that $f(z) \geq 0$ provides:

$$\begin{aligned} f(\lambda y + (1-\lambda) y) &= f(z) \left(\frac{f(y)}{f(z)}\right)^{\lambda} \\ &\leq \lambda \frac{f(y)}{f(z)} + (1-\lambda) \\ &= \lambda f(y) + (1-\lambda) f(z). \end{aligned}$$

Thus, it states that f is convex and the result follows. □

We summarized the above discussion in next remark.

Remark 2.2 According to Theorem 2.1 the cost function of (4) is convex. Since the constraints of (4) are linear so the WTA problem (4) is a convex nonlinear optimization problem. In next section we consider the general form of the convex nonlinear programming problem with linear constraints and propose a projection RNN model to solve the original problem.

3 Projection Recurrent Neural Network Model

Consider the following convex nonlinear programming problem with linear constraints:

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{5}$$

where $f(\cdot)$ is twice continuously differentiable and convex in $\mathbb{R}^n$. The problem (5) can be rewritten as an unconstrained optimization below:

$$\begin{aligned} \min \quad & F(x) = f(x) + \frac{\lambda}{2} \|Ax - b\|^2 \\ s.t. \quad & x \geq 0. \end{aligned} \tag{6}$$

Since the problem is convex so the gradient of $F(\cdot)$ equal to zero ($\nabla F(x) = 0$) is both the necessary and the sufficient conditions for optimality. Here, we need some requirements from projection function. We summarized them in the following lemma.

Lemma 3.1 ([32,33]) *Let Ω be a closed convex subset of $\mathbb{R}^n$. The projection map $P_{\Omega}(\cdot)$ is defined as follows:*

$$P_{\Omega}(x) = arg \min_{y \in \Omega} \|x - y\|.$$

Then, the projection map has the following properties:

1. $\|P_\Omega(x) - P_\Omega(y)\| \le \|x - y\|, \ \forall x, y \in \mathbb{R}^n$.
2. $x \in \Omega \Longleftrightarrow P_\Omega(x) = x$.
3. *$x^* \in \Omega$ is an optimal solution of the problem* $\min \{F(x) \mid x \in \Omega\}$*, if and only if, for every $\alpha > 0$, x^* satisfies in the projection equation $P_\Omega[x^* - \alpha\nabla F(x^*)] = x^*$.*
4. $(v - P_\Omega(v))^T (w - P_\Omega(v)) \le 0, \ \forall v \in \mathbb{R}^n, \ \forall w \in \Omega$.

Hence, the projection RNN model associated with (6) can be described as follow:

$$\frac{dx}{dt} = P_\Omega(x) - \alpha\nabla F(P_\Omega(x)) - x, \tag{7}$$

where $\alpha > 0$ and $P_\Omega(\cdot)$ is the projection function. Next theorem gives an important result.

Theorem 3.2 *x^* is the optimal solution of* (6)*, if and only if, x^* is the equilibrium point of the projection RNN model* (7).

Proof Let x^* be the equilibrium point of the projection RNN model (7). Therefore, $\frac{dx^*}{dt} = 0$ and then $P_\Omega(x^*) - \alpha\nabla F(P_\Omega(x^*)) - x^* = 0$ or $P_\Omega(x^*) - x^* = \alpha\nabla F(P_\Omega(x^*))$. Employing part (2) from Lemma 3.1 yields $\nabla F(x^*) = 0$ which is both the necessary and the sufficient conditions for optimality. For the converse, let that x^* is the optimal solution of (6). In accordance with part (3) from Lemma 3.1 we have $P_\Omega[x^* - \alpha\nabla F(x^*)] = x^*$. Since $x^* \in \Omega$ we have:

$$P_\Omega[P_\Omega(x^*) - \alpha\nabla F(P_\Omega(x^*))] = P_\Omega(x^*),$$

or it can be rewritten as $P_\Omega(x^*) - \alpha\nabla F(P_\Omega(x^*)) = x^*$ or $P_\Omega(x^*) - \alpha\nabla F(P_\Omega(x^*)) - x^* = 0$. This implies that $\frac{dx^*}{dt} = 0$. Thus, the proof is completed. □

4 Stability Analysis

Here, we study the stability analysis of the proposed model (7). We start this section with a lemma providing the Lipschitz property of the right hand side of the proposed model (7).

Lemma 4.1 *Right hand side of* (7) *is Lipschitzian.*

Proof At first we have:

$$\nabla F(P_\Omega(x)) = \nabla f(P_\Omega(x)) + \lambda(AP_\Omega(x) - b).$$

Also, from mean value theorem we have:

$$\|\nabla f(x) - \nabla f(y)\| \le \|\nabla^2 f(cx + (1-c)y)\| \|x - y\|.$$

Assume that $x, y \in \mathbb{R}$ and $M = \sup_{x\in\Omega} \|\nabla^2 f(x)\|$. According to the above relations we have:

$$\begin{aligned}
&\Big\| P_\Omega(x) - \alpha\nabla f(P_\Omega(x)) - \alpha\lambda(AP_\Omega(x) - b) - P_\Omega(y) + \alpha\nabla f(P_\Omega(y)) + \alpha\lambda(AP_\Omega(y) - b) \Big\| \\
&\quad \le \Big\| P_\Omega(x) - P_\Omega(y) \Big\| + \alpha\lambda\|A\| \Big\| P_\Omega(x) - P_\Omega(y) \Big\| + \Big\| \alpha\nabla f(P_\Omega(x)) - \alpha\nabla f(P_\Omega(y)) \Big\| \\
&\quad \le \|x - y\| + \alpha\lambda\|A\|\|x - y\| + \alpha \Big\| \nabla^2 f\left(cP_\Omega(x) + (1-c)P_\Omega(y)\right) \Big\| \|x - y\| \\
&\quad \le (1 + \alpha\lambda\|A\|)\|x - y\| + \alpha M\|x - y\| \\
&\quad = (1 + \alpha\lambda\|A\| + \alpha M)\|x - y\|.
\end{aligned}$$

Thus, the right hand side of model (7) is Lipschitz continuous function with constant $1 + \alpha\lambda\|A\| + \alpha M$. □

Lemma 4.2 *There exists a unique solution $x(t)$ for model (7).*

Proof According to Lemma 4.1, the right hand side of model (7) has the Lipschitz continuous property therefore, Lemma 6.6 completes the proof for some $\tau > t_0$ over $[t_0, \tau)$ as $\tau \to \infty$.

Before proving the global exponential stability theorem of model (7) we need a useful lemma.

Lemma 4.3 *The mapping $g(x) = -\alpha\nabla F(P_\Omega(x)) + P_\Omega(x)$ for $\alpha < \frac{2}{\max_{x\in\Omega}\|\nabla^2 F(x)\|^2}$ is contractive.*

Proof Based on part (2) from Lemma 3.1 and the mean value theorem we have:

$$\begin{aligned}&\Big\| -\alpha\nabla F(P_\Omega(x)) + P_\Omega(x) - (-\alpha\nabla F(P_\Omega(y)) + P_\Omega(y))\Big\|\\ &\quad\le \max_{x,y\in\Omega}\Big\| -\alpha\nabla^2 F(cx + (1-c)y) + I\Big\|\|x-y\|.\end{aligned}\tag{8}$$

Since mapping F is twice continuously differentiable so $\nabla^2 F(\cdot)$ is a symmetric positive definite matrix and provides:

$$\max_{x\in\Omega}\Big\| -\alpha\nabla^2 F(x) + I\Big\| = \max_{\sigma}|\sigma\alpha - 1|,$$

where $\sigma > 0$ is the eigenvalue of the $\nabla^2 F(\cdot)$. From the fact that, $\alpha < \frac{2}{\max_{x\in\Omega}\|\nabla^2 F(x)\|^2}$ we get:

$$-1 < \sigma\alpha - 1 \le \alpha \max_{x\in\Omega}\Big\| -\alpha\nabla^2 F(x)\Big\| - 1 < 1.$$

Hence, giving:

$$\max_{x\in\Omega}\Big\| -\alpha\nabla^2 F(x) + I\Big\| = \max_{\sigma}|\sigma\alpha - 1| = \eta < 1.$$

Applying equation (8) yields:

$$\Big\| -\alpha\nabla F(P_\Omega(x)) + P_\Omega(x) - (-\alpha\nabla F(P_\Omega(y)) + P_\Omega(y))\Big\| \le \eta\|x-y\|.$$

This implies that the mapping $g(x) = -\alpha\nabla F(P_\Omega(x)) + P_\Omega(x)$ is contractive. □

Theorem 4.4 *The proposed model (7) is globally exponentially stable.*

Proof Theorem 3.2 and Lemma 4.2 give the existence and the uniqueness of the solution of (7) over $[t_0, \infty)$. Consider the following equations:

$$\begin{aligned}x(t) &= e^{-I(t-t_0)}x_0 + \int_{t_0}^{t} e^{-I(t-s)}\left(-\alpha\nabla F(P_\Omega(x)) + P_\Omega(x)\right)ds,\\ x^* &= e^{-I(t-t_0)}x^* + \int_{t_0}^{t} e^{-I(t-s)}\big(-\alpha\nabla F(P_\Omega(x^*)) + P_\Omega(x^*)\big)ds.\end{aligned}$$

Combining the above equations gives:

$$x(t) - x^* = e^{-I(t-t_0)}(x_0 - x^*) + \int_{t_0}^{t} e^{-I(t-s)} \big(-\alpha \nabla F(P_\Omega(x)) + P_\Omega(x) \\ - (-\alpha \nabla F(P_\Omega(x^*)) + P_\Omega(x^*))\big)\, ds,$$

or it can be rewritten as follow:

$$\begin{aligned} \|x(t) - x^*\| &\le e^{-(t-t_0)}\|x_0 - x^*\| + \int_{t_0}^{t} e^{-(t-s)} \Big\| - \alpha \nabla F(P_\Omega(x)) \\ &\quad + P_\Omega(x) - (-\alpha \nabla F(P_\Omega(x^*)) + P_\Omega(x^*)) \Big\| ds, \\ &\le e^{-(t-t_0)}\|x_0 - x^*\| + \max_{x\in\Omega} \Big\| - \alpha \nabla^2 F(x) + I \Big\| \int_{t_0}^{t} e^{-(t-s)} \|x(s) - x^*\| ds \\ &= e^{-(t-t_0)}\|x_0 - x^*\| + \eta \int_{t_0}^{t} e^{-(t-s)} \|x(s) - x^*\| ds. \end{aligned}$$

Note that, we use Lemma 4.3 in the second inequality. The above inequality can be expressed as follow:

$$\|x(t) - x^*\| e^t \le e^{t_0}\|x_0 - x^*\| + \eta \int_{t_0}^{t} e^{s} \|x(s) - x^*\| ds,$$

employing the Gronwall inequality [34] verifies that:

$$\|x(t) - x^*\| \le e^{-(t-t_0)(1-\eta)}\|x_0 - x^*\|.$$

Since $\eta < 1$, thus, the result follows. □

5 Simulation Results

This section, gives some simulation results to show the performance of the proposed model. The codes are developed using symbolic computation software MATLAB (ode45) and the calculations are implemented on a machine with Intel core 7 Duo processor 2 GHz and 8 GB RAM.

Example 5.1 Consider the WTA problem with $N = 5$. We set the random parameters as follow:

$$u = (1, 3, 5, 2, 4), \quad p_{i,j} = \begin{bmatrix} 0.6 & 0.5 & 0.8 & 0.7 & 0.4 \\ 0.9 & 0.5 & 0.7 & 0.8 & 0.4 \\ 0.8 & 0.9 & 0.4 & 0.5 & 0.8 \\ 0.8 & 0.9 & 0.8 & 0.6 & 0.4 \\ 0.7 & 0.4 & 0.5 & 0.6 & 0.7 \end{bmatrix}$$

By using the proposed method and applying the above parameters the optimal solution of the problem is:

$$x_{1,4} = x_{2,1} = x_{3,2} = x_{4,3} = x_{5,5} = 1, \quad F = 2.7.$$

Figure 1 displays the transient behaviour of the continuous-time projection RNN model (7) in Example 5.1.

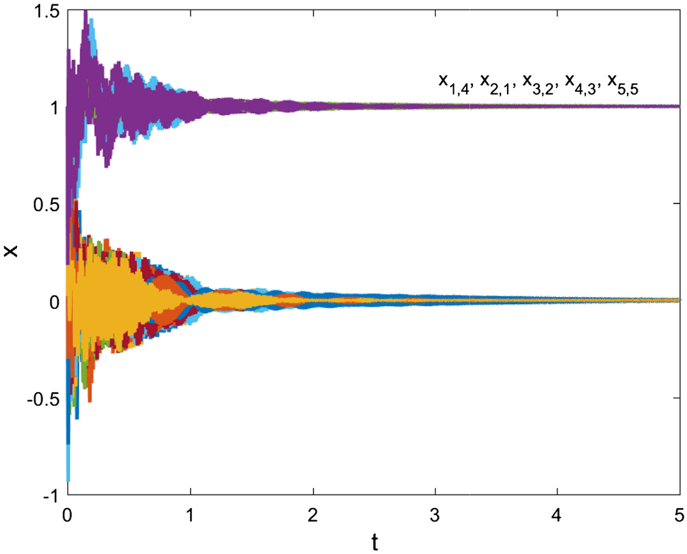

Fig. 1 Transient behaviour of the continuous-time model (7) in Example 5.1

Example 5.2 In this example we consider the WTA problem with $N = 10$ and randomized parameters. Also, we compare the proposed method with some other methods such as general Genetic Algorithm (GA) [35] and GA with greedy eugenics [9]. At first, we shall introduce them in the following.

1. *General GA* GAs have been considered as a class of general-purpose search strategies for optimization problems. A general GA is shown as follows [35]. Let $P(t)$ and $C(t)$ be parents and offspring in generation t.

```
Procedure: General GA
Begin
t ← 0;
Initialize P(t);
Evaluate P(t);
While (not matched with the termination
conditions) do
Recombine P(t) to yield C(t);
Evaluate C(t);
Select P(t+1) from P(t) and C(t);
t ← t+1;
End
End
```

The GA starts with a set of randomly selected chromosomes as the initial population that encodes a set of possible solutions. Variables of a problem are represented as genes in a chromosome, and chromosomes are evaluated according to their fitness values, which are obtained by evaluating the considered fitness function or cost function such as (3). Recombination typically involves two genetic operators: (1) crossover and (2) mutation. Genetic operators alter the composition of genes to create new chromosomes referred to as offspring. The selection operator is an artificial version of natural selection, a

Table 2 Comparison results with randomized parameters with $N = 10$

Method	Operator	Convergence (%)	CPU-time (s)
General GA	OCP	10	–
	MCX	20	–
	EX	40	–
GA with greedy eugenics	OCP	100	76.55
	MCX	100	50.68
	EX	100	20.28
Proposed model (7)	–	Globally	20.23

Darwinian survival of the fittest among populations, to create populations from generation to generation. Chromosomes with better fitness have higher probabilities of being selected in the next generation. After several generations, GA can converge to the best solution.

2. *GA with greedy eugenics* In this algorithm instead of using random mechanisms or some powerful global search mechanisms to find a neighbour, it uses a greedy reformation scheme in the place of eugenics. In other words, the eugenic scheme is to greedily reform the current chromosome. Even though greedy algorithms may have a great possibility to be trapped into a local optimum, due to crossover and mutation operations and the parallel search nature used in GAs, the search can easily escape from local optima. Hence, if a greedy eugenics is used, it can find locally good solutions quickly and will not be trapped in local optima. The concept of eugenics is illustrated by the following simple algorithm [7].

```
Procedure: Simple eugenics
Begin
While (the eugenic process has not been
stopped) do
Generate a neighbourhood solution π′
If C(π′) < C(π) then π = π′
End
End
```

The comparison results with $N = 10$ are presented in Table 2. In Table 2 we use some abbreviation that may be introduced. The one-cut-point (OCP) crossover, inversion mutation, and roulette selection operators are employed as the basic operations in GA [35]. The OCP operator is to randomly generate OCP and to swap the cut parts of two parents so as to generate offspring. The concept of EX (crossover operator) is to construct offspring with possibly good genes from parents. EX is modified from CX in [7]. In the CX operator, the information contained in both parents is preserved. EX also adopts the similar concept, except it preserves only those genes supposed to be "good". MCX (modified CX) is similar to CX adopted in [7]. MCX keeps genes with the information contained in both parents and the other genes are randomly selected from parents. In other words, MCX is CX without the repairing process. According to Table 2 one can find out that the GA with greedy eugenics is more efficient than general GA. However, the proposed method in this paper is better that two other algorithms. The RNN models are stable which means get the optimal solution but the GAs are iterative algorithms and does not get the optimum. For $N = 50$ and $N = 80$ the comparison results are presented in Table 3. The reported results in Table 3 is the standard

Table 3 Comparison results with randomized parameters for $N = 50$ and $N = 80$

Algorithms	$W = T = 50$	$W = T = 80$
Simulated annealing (SA)	50.8541	50.8137
General GA	47.6715	52.9217
GA with simple eugenics	30.7316	60.7415
GA with SA as eugenics	27.981	55.215
GA with immunity as eugenics	25.8734	30.2742
GA with greedy eugenics	19.1175	25.973
Proposed model (7)	0	0

deviation of each algorithm. It easy to check that the proposed method in this paper is more efficient than the other methods. The reason is that our model is globally convergent to the optimal solution of the problem. In fact, the model is globally exponentially stable.

6 Conclusion and Discussion

Here, we state a brief description about some issues for the proposed approach, findings of the paper, and conclusion.

Through the paper, a novel dynamic model for solving the WTA problem was proposed. The given model was solved by an Ordinary Differential Equation (ODE). Note that, the proposed dynamic model is a one-layer design. Reported results were compared with some other methods. Moreover, the dynamic system approach does not depend on the initial point (i. e., we obtain the unique solution of the problem whether we choose the initial point from outside of the convergence region or not). According to the theoretical results and numerical experiments we can summarize the advantages of the paper as follow:

- In comparison with the other existing approaches, using of dynamic system models give less CPU-Time.
- As stated before, the dynamic system approach does not depend on the initial points. This advantage is shown by testing all simulation results with random initial points. This is because, the model is globally convergent to the optimal solution of the problem.
- The proposed model has the global exponential stability property.

At last, the work is in progress to extend the other recurrent models for solving other optimization problems. As stated before, the concepts of projection RNNs are applied to other fields of study such control engineering. Since RNN models are high-performance tools for obtaining the solutions to the mentioned problems, we hope that our results would have some degree of impact on the practices of other disciplines.

Appendix: Dynamical System

Here, we give some requirements from dynamical system.

Definition 6.1 (*Equilibrium point* [34,36]) In the following dynamical system:

$$\dot{x} = f(x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^n, \tag{9}$$

where f is a function from $\mathbb{R}^n$ to $\mathbb{R}^n$, x^* is called an equilibrium point of (9), if $f(x^*) = 0$.

Definition 6.2 (*Stability in the sense of Lyapunov* [34,36]) Suppose $x(t)$ is a solution for (9). Equilibrium point x^* is said to be stable in the sense of Lyapunov, if for any $x_0 = x(t_0)$ and any $\varepsilon > 0$, there exists a $\delta > 0$, such that:

$$\|x(t) - x^*\| < \varepsilon, \quad \forall\, t \geq t_0, \quad \|x(t_0) - x^*\| < \delta.$$

Definition 6.3 (*Asymptotic stability* [34,36]) An isolated equilibrium point x^* is said to be *asymptotically stable*, if in addition to being Lyapunov stable, it has the property that $x(t) \to x^*$ as $t \to \infty$ for all $\|x(t_0) - x^*\| < \delta$.

Definition 6.4 (*Globally exponentially stable* [34,36]) A dynamic system is said to be globally exponentially stable with degree ζ at x^* if every trajectory starting at any initial point $x(t_0) \in \mathbb{R}^n$ satisfies:

$$\|x(t) - x^*\| \leq \mu \|x(t_0) - x^*\| e^{-\zeta(t-t_0)}, \quad \forall\, t \geq t_0,$$

where ζ and μ are positive constants independent of the initial points.

It is clear that the exponential stability implies the globally asymptotical stability and the system converges arbitrarily fast.

Lemma 6.5 ([34,36])

1. *An isolated equilibrium point x^* is Lyapunov stable if exists a Lyapunov function over some neighbourhood Ω of x^*.*
2. *An isolated equilibrium point x^* is asymptotically stable if there is a Lyapunov function over some neighbourhood Ω of x^* such that $\frac{dE}{dt} < 0$ for all $x \neq x^* \in \Omega$.*

Lemma 6.6 ([34,36]) *Assume that $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous mapping. Then, for arbitrary $t_0 \geq 0$ and $x_0 \in \mathbb{R}^n$ there is a local solution $x(t)$ for the dynamical system* (9) *with $t \in [t_0, \tau)$ for some $\tau \geq t_0$. Furthermore, if f is locally Lipschitzian continuous at x_0 then the solution is unique, and if f is Lipschitzian continuous in $\mathbb{R}^n$ then τ can be extended into ∞.*

References

1. denBroeder GG Jr, Ellison RE, Emerling L (1959) On optimum target assignments. Oper Res 7(3):322–326
2. Soland RM (1987) Optimal terminal defense tactics when several sequential engagements are possible. Oper Res 35(4):537–542
3. Meter AW, Preston FL (1990) A suite of weapon assignment algorithms for a SDI mid-course battle manager: AT&T bell laboratories
4. Lloyd SP, Witsenhausen HS (1986) Weapon allocation is NP-complete. In: IEEE summer simulation conference, Reno, NV, USA
5. Bjorndal AMH et al (1995) Some thoughts on combinatorial optimization. Eur J Oper Res 83(2):253–270
6. Hammer PL (1965) Some network flow problems solved with pseudo-boolean programming. Oper Res 13:388–399
7. Merz P, Freisleben B (2000) Fitness landscape analysis and memetic algorithms for quadratic assignment problem. IEEE Trans Evolut Comput 4:337–352
8. Lee Z-J, Lee C-Y, Su S-F (2000) A fuzzy-genetic based decision-aided system for the naval weapon-target assignment problems. In: Proceedings of 2000 R.O.C. automatic control conference, pp 163–168
9. Lee Z-J, Su S-F, Lee C-Y (2003) Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Trans Syst Man Cybern Part B Cybern 33(1):113–121

10. Hopfield JJ, Tank DW (1985) Neural computation of decisions in optimization problems. Biol Cybern 52:141–152
11. Tank DW, Hopfield JJ (1986) Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit. IEEE Trans Circuits Syst 33:533–541
12. Hu X (2009) Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints. Neurocomputing 72:1131–1137
13. Sun J, Chen J-S, Ko C-H (2012) Neural networks for solving second-order cone constrained variational inequality problem. Comput Optim Appl 51:623–648
14. Eshaghnezhad M, Effati S, Mansoori A (2017) A neurodynamic model to solve nonlinear pseudo-monotone projection equation and its applications. IEEE Trans Cybern 47(10):3050–3062
15. Effati S, Mansoori A, Eshaghnezhad M (2015) An efficient projection neural network for solving bilinear programming problems. Neurocomputing 168:1188–1197
16. He X, Huang T, Yu J, Li C, Li C (2017) An inertial projection neural network for solving variational inequalities. IEEE Trans Cybern 47(3):809–814
17. Mansoori A, Erfanian M (2018) A dynamic model to solve the absolute value equations. J Comput Appl Math 333:28–35
18. Mansoori A, Eshaghnezhad M, Effati S (2018) An efficient neural network model for solving the absolute value equations. IEEE Trans Circuits Syst II Express Briefs 65(3):391–395
19. Cheng L, Hou Z-G, Tan M (2008) A neutral-type delayed projection neural network for solving nonlinear variational inequalities. IEEE Trans Circuits Syst II Express Briefs 55(8):806–810
20. Eshaghnezhad M, Rahbarnia F, Effati S, Mansoori A (2018) An artificial neural network model to solve the fuzzy shortest path problem. Neural Process Lett. https://doi.org/10.1007/s11063-018-9945-y
21. Liu D, Hu S, Wang J (2004) Global output convergence of a class of continuous-time recurrent neural networks with time-varying thresholds. IEEE Trans Circuits Syst II Express Briefs 51(4):161–167
22. Mohammadi M, Mansoori A (2018) A projection neural network for identifying copy number variants. IEEE J Biomed Health Inform. https://doi.org/10.1109/JBHI.2018.2871619
23. Mansoori A, Effati S (2019) An efficient neurodynamic model to solve nonlinear programming problems with fuzzy parameters. Neurocomputing 334:125–133
24. Mansoori A, Eshaghnezhad M, Effati S (2019) Recurrent neural network model: a new strategy to solve fuzzy matrix games. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2018.2885825
25. Zhang Y, Chen S, Li S, Zhang Z (2018) Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. IEEE Trans Ind Electron 65(6):4909–4920
26. Zhang Y, Li S (2018) Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks. IEEE Trans Syst Man Cybern Syst 48(11):1957–1967
27. Chen D, Zhang Y (2017) A hybrid multi-objective scheme applied to redundant robot manipulators. IEEE Trans Autom Sci Eng 14(3):1337–1350
28. Chen D, Zhang Y, Li S (2018) Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances. Neurocomputing 275:845–858
29. Chen D, Zhang Y (2018) Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators. IEEE Trans Neural Netw Learn Syst 29(9):4385–4397
30. Hosein PA (1989) Class of dynamic non linear resource allocation problems. In: Ph.D Thesis, Massachusetts Institute of Technology, MA, USA
31. Bazaraa MS, Shetty C, Sherali HD (1979) Nonlinear programming, theory and algorithms. Wiley, New York
32. Facchinei F, Pang J (2003) Finite-dimensional variational inequalities and complementarity problems. Springer, New York
33. Nagurney A (2002) Variational inequalities, Isenberg School of Management University of Massachusetts
34. Miller RK, Michel AN (1982) Ordinary differential equations. Academic Press, New York
35. Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York
36. Khalil HK (1996) Nonlinear systems. Prentice-Hall, Michigan