Solving Weapon-Target Assignment Problem Using Discrete Particle Swarm Optimization*

Xiangping Zeng1,2, Yunlong Zhu1, Lin Nan1, Kunyuan Hu1, Ben Niu1,2 and Xiaoxian He1,2

1Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang
2Graduate school of the Chinese Academy of Sciences, Beijing
{zengxiangping, ylzhu, nl}@sia.cn

Abstract -This paper presents a discrete particle swarm optimization (DPSO) to solve weapon-target assignment (WTA) problem. The proposed algorithm sponges the advantages of PSO and GA. Originally the greedy searching strategy is introduced into DPSO in which a priority set is constructed to control the local search and converge to the global optimum efficiently. Then the particles would be updated based on the information of priority set. Furthermore, the concept of “permutation” is employed to the update strategy. Finally, particles will be reinitialized as long as they are stagnated in the search space. The experimental results illustrate that the DPSO is a promising optimization method, which is especially useful for optimization problem with discrete variables.

Index Terms -Weapon-target assignment, particle swarm optimization, greedy searching strategy.

I. 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. This is an NP-complete problem that rapidly becomes intractable with growing targets size [1, 2]. Various methods have been proposed to solve such problems [3-6]. Most of them are based on graph search approaches and usually result in exponential computational complexity. Recently, genetic algorithms (GA) have been widely used as search algorithms in WTA and have also demonstrated satisfactory performance [7-11]. Even though GA could find the best solution in those simulated cases [7-11], there are still some problems such as premature convergence and slow evolution. Thus, we propose a new heuristic approaches based on PSO to solve WTA problem.

The Particle Swarm Optimization (PSO), originally introduced in terms of social and cognitive behaviour by Kennedy and Eberhart in 1995 [12, 13], has come to be widely used in many areas [14-16]. The PSO method is very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution.

The rest of the paper is organized as follows. Section II defines the WTA problems. Section III introduces standard PSO. The DPSO for WTA is detailed in section IV. In section V, the DPSO is employed to solve WTA problems and experimental results are reported as compared with some other algorithms. Finally, Section VI concludes the paper.

II. WEAPON-TARGET ASSIGNMENT (WTA) PROBLEMS

On modern battlefields, it is an important task for battle managers to make a proper WTA to defend own-force assets. These targets have different probabilities of killing to platforms that are dependent on the target types. In our study, two following assumptions are made. (1) There are $W$ weapons and $T$ targets and all weapons must be assigned to targets. (2) The individual probability of killing ($K_{ij}$) by assigning the $i$th target to the $j$th weapon is known for all $i$ and $j$. This probability defines the effectiveness of the $j$th weapon to destroy the $i$th target. The overall probability of the $i$th target being destroyed is $\prod_{j=1}^{W}(1-K_{ij})^{X_{ij}}$ where $X_{ij}$ is a Boolean value indicating whether the $j$th weapon is assigned to the $i$th target. $X_{ij}=1$ indicates that the $j$th weapon is assigned to the $i$th target. Then, the considered WTA problems are to minimize the following cost function.

$$F(\pi) = \sum_{i=1}^{T} EDV(i) \times \left[ 1 - \prod_{j=1}^{W}(1-K_{ij})^{X_{ij}} \right]$$ (1)

subject to the assumption that all weapons must be assigned to targets, that is

$$\sum_{i=1}^{T} X_{ij} = 1, \quad j = 1,2,...,W$$ (2)

where $EDV(i)$ is the expected damage value of the $i$th target to the asset, $\pi$ is a feasible assignment list and $\pi(j)=i$ indicates that assigning the $i$th target to the $j$th weapon.

III. STANDARD PARTICLE SWARM OPTIMIZATION

The PSO is motivated from the simulation of simplified social behavior [17, 18]. Supposed that the search space is $D$-dimensional, then the $i$th individual, which is called particle can be represented by a $D$-vector as $X_i=(x_{i1}, x_{i2},...,x_{id})^T$, the velocity of the particle $i$ is represented as $V_i=(v_{i1}, v_{i2},...,v_{id})^T$.

---

*Project supported by the National Natural Science Foundation, China (No. 70431003) and the National Basic Research Program, China (2002CB312204).
In each generation, the best previous experience of the \( i \) th particle is denoted as \( p_{\text{best}_i} = (p_{i1}, p_{i2}, \ldots, p_{id}) \) and \( g_{\text{best}} \) is the best global experience of all particles. Thus, the velocity and position of each particle is adjusted by the following formulae

\[
V_{i}^{t+1} = V_{i}^{t} + c_1 \text{Rand}_i((p_{\text{best}_i} - X_{i}^{t}) + c_2 \text{Rand}_2(g_{\text{best}} - X_{i}^{t})) \quad (3)
\]

\[
X_{i}^{t+1} = X_{i}^{t} + V_{i}^{t+1} \quad (4)
\]

where \( i = 1, 2, \ldots, N \), and \( N \) is the size of all the swarm, \( t \) represents the iteration steps, \( c_1 \) and \( c_2 \) are positive constants, \( \text{Rand}_i() \) and \( \text{Rand}_2() \) are random numbers, uniformly distributed in \([0, 1]\).

In (3), the first term represents the previous velocity. The second term is known as the “cognitive” component, denoting the personal thinking of each particle. The third term is known as the “social” component, which represents the collaborative effect of the particles in finding the global optimal solution.

IV DPSO FOR WTA

As a novel computing technique, the PSO has succeeded in solving many continuous problems, but has done few with optimization problem with discrete variables [19, 20]. Due to the continuous search, it is not easy applied to WTA problems. Therefore, a discrete PSO model is presented here.

In this method, each particle corresponds to a candidate solution of the underlying problem. Thus, let each particle represents a decision for weapon-target assignment using a vector of \( W \) elements, and each element is an integer value between 1 to \( T \). Fig. 1 illustrates the \( i \) th particle corresponding to a weapon-target assignment in which eight weapon are assigned to eight targets, and particle \( i_6 = 2 \) means that target 6 is assigned to weapon 2.

<table>
<thead>
<tr>
<th>( i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>particle ( _i )</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ \text{particle}_{i_6} = 2 \]

Fig. 1 An example for a particle

The velocity of each particle is also considered as a vector by permutation of two particles’ positions. Suppose \( \vec{X} \) and \( \vec{Y} \) are particle’s positions, then the operation \( \vec{X} \oplus \vec{Y} \) can yield the \( \vec{V} \). Note that the symbol \( \oplus \) denotes the operation of positions.

To enhance the performance of convergence, Greedy Searching Strategy (GSS) is applied.

1. **Greedy Searching Strategy (GSS)**

![GSS Pseudocode](image)

The GSS is to find a better candidate nearby the current one before moving to the next stage of search. In this research, we view the GSS as a major role nailing down a local optimum for the solution provided by PSO. The idea is to greedily assign targets to weapons that have the highest \( EDV(i) \times K_{ij} \). Thus, a threshold \( Def \) should be predefined and priority set \( \phi(j) (j = 1, 2, \ldots, W) \) be constructed for each weapon to store the qualification targets. The GSS is illustrated in the Fig. 2.

2. **Permutation**

Due to GSS may have a great possibility to be trapped into a local optimum, two probabilities are employed to update strategy.

1) One is Fixed Probability (FP), which denotes \( W \times FP \) elements are chosen randomly from \( g_{\text{best}} \). Then these elements are kept to permute corresponding elements of current particle. That aims at enhancing convergence of the global optimum of the search.

2) The other is Unfixed Probability (UFP), which denotes \( W \times UFP \) elements are chosen randomly from \( p_{\text{best}} \). Then the corresponding elements of current particle are determined based on theirs corresponding priority sets \( \phi \). It should make sure that the least repeated elements should be preserved in the new particle. The major consideration of this modification is to avoid premature convergence.

An example illustrates how permutation is implemented. Supposed there are six weapons and six targets. One generation, if \( UFP = 0.33, FP = 0.33 \), unfixed positions, e.g., the second and the fourth positions, have the priority sets \( \phi(2) = \{1, 4, 5\}, \phi(4) = \{2, 5, 6\} \) respective, and fixed positions are the first and the fifth positions. Thereafter particles are
After permuting particles’ positions, current particle is

Current particle = 6 1 3 5 2 4

Now even though GSS is used, it can control the local search and converge to the global optimum solution efficiently.

3. DPSO

Then, the operation of permutation is applied to the proposed algorithm. The particles should be manipulated according to the following equations at each iteration step,

\[ V_{i+1}^t = V_i^t \oplus \sum_{i=1}^{int(W \times UFP)} \text{Rand}(i) \times pbest_i^t \oplus \sum_{j=1}^{int(W \times FP)} \text{Rand}(2j) \times gbest^t \]  

\[ X_{i+1}^t = X_i^t + V_{i+1}^t \]  

where \( W \) is the size of all the weapons, \( t \) represents the iteration steps, \( pbest \) is previous best experience of the particle, \( gbest \) is best global experience of all particles, \( UFP \) is unfixed probability, and \( FP \) is fixed probability.

Now, the pseudocode for this proposed algorithm is given in Fig. 3.

```
Begin
    Construct priority set \( \phi \) for each weapon;
    Initialize current particle based on the information of \( \phi \);
    Loop for \( t = 1 \) to maximum iteration or until convergence
        For \( i = 0 \) to int \((W \times UFP)\)
            permute current particle according to the second term of (5);
        End for
        For \( j = 0 \) to int \((W \times FP)\)
            permute current particle according to the third term of (5);
        End for
        Compute current fitness value;
        Update particles according to (6);
    End Loop
    Output \( gbest \) fitness value and \( gbest \) particle;
End
```

Fig. 3 Pseudocode for the DPSO

V EXPERIMENTATION AND RESULTS

To illustrate the proposed algorithm, some simulations for WTA problems are performed in this section. The fitness value of each solution is evaluated using the fitness function (1). All the simulations are coded in java programming language on a Pentium 1-GHz, 256MB machine running under windows environment. And they use the same initial population under the same case, which are randomly generated. And simulations are conducted for ten trials and the average results are reported.

The first simulation with ten weapons and ten targets is used to investigate the performances of parameters in DPSO and the results are shown in Table 1.

In Table 1, 100% means that all tests can converge to the best fitness value. It can be seen that different parameters have gained the same best fitness value, but used different CPU time. So the well defining parameters result in better results and faster convergence. The best results have been gained while \( UFP = 0.5 \) and \( FP = 0.33 \) which will be used in the following simulations.

Next, to measure the efficiency of DPSO, results are compared with General GA, GA with greedy eugenics [21]. In this case, these parameters used are the crossover probability \( p_c = 0.8 \), the mutation probability \( p_m = 0.4 \) in General GA and GA with greedy eugenics respective.

60% means that six tests can converge to the best fitness value in Table 2. It can be seen that DPSO always use the least CPU time when three algorithms all gain the best fitness.

Finally, simulations with increasing numbers of weapon and target are conducted and best fitness values obtained by various search algorithms are listed in Table3.

By comparing the statistical data, it is evident that DPSO can converge better fitness values than other algorithms with the same time. Furthermore, three fitness curves of various
algorithms for $W = 20$ and $T = 20$ have been showed in Fig.4 to test the convergent performance. The results indicate that DPSO outperform General GA and GA with greedy eugenics in terms of better convergence and higher quality of the solution.

In conclusion, the results clearly indicate the superior of DPSO to the other algorithms in the CPU time and convergence efficiency.

VI CONCLUDING REMARKS

In this paper, a DPSO has been presented and employed to solve WTA problems. With this method, the “social” part and the “cognitive” part of the particle swarm have been considered to update strategy. It can efficiently control the local search and converge to the global optimum solution by employing a GSS. It can easily used to solve optimization problem with discrete variables. The better performance of DPSO can also be ascribed to its simplicity.

The performance of DPSO algorithm is evaluated in comparison with other algorithms. The simulation results show that DPSO outperforms that of other existing algorithms in most of the test cases. The DPSO runs faster and is indeed capable of obtaining higher quality solutions. It is an attractive alternative for solving the WTA problems.

REFERENCES