Proceedings of the 34th Chinese Control Conference
July 28-30, 2015, Hangzhou, China

Efficiently solving multi-objective dynamic weapon-target assignment problems by NSGA-II

LI Juan[1,2], CHEN Jie[1,2], XIN Bin [1,2] ⊠

1. School of Automation, Beijing Institute of Technology, Beijing 100081, P. R. China
E-mail: 00134476@bit.edu.cn

2. Key Laboratory of Complex System Intelligent Control and Decision, Beijing 100081, P. R. China
E-mail: brucebin@bit.edu.cn

Abstract: A multi-objective dynamic weapon-target assignment (MODWTA) problem with three competing objectives, resource constraints, feasibility constraints and fire transfer constraints is studied in this paper. The weapon-target assignment (WTA) problem is formulated into a multi-objective constrained combinatorial optimization problem. Apart from maximizing damage to hostile targets, the research in this paper follows the principle of minimizing ammunition consumption and total operational time under the consideration of limited resource constraints, feasibility constraints and fire transfer constraints. Because of these competing objectives and rigorous constraints, the WTA problem becomes more complicated. In order to tackle the two challenges, the well-known non-dominated sorting genetic algorithm with elitist strategy, namely NSGA-II, is adopted according to the specific structure of the problem to achieve efficient problem solving. Besides, the proposed NSGA-II is compared with a multi-objective Monte Carlo random sampling method, which shows the superiority of the proposed MODWTA algorithm. The numerical simulation results show that the proposed NSGA-II algorithm effectively finds the approximate Pareto front within acceptable time.

Key Words: multi-objective optimization problem (MOP), dynamic weapon-target assignment (DWTA), fire transfer, NSGA-II, combinatorial optimization

1 Introduction

Traditionally, assigning weapons to enemy targets in a certain combat situation largely depends on human decision makers. However, even an experienced decision maker can not always maximize the damage to hostile targets with limited weapon resources. A decision aid which can automatically and effectively provide optimal or near-optimal solutions will be very useful in the future warfare. Furthermore, since modern warfare is network-centric and information-dependent, battle command should be improved to match with its accuracy. An automatic decision aid can not only strengthen the accuracy and rapidity of involved forces, but also reduce dependency on human decision makers. Such a valuable decision aid is a feasible and desirable solution to the WTA.

The WTA problem is a classical constrained optimization problem originating from the military operations research, which aims to meet one or more desired operational goals via assigning a limited number of weapons to enemy targets. Solving WTA problems is an effective way to improve the operational effect without increasing resource consumption or higher technical support, so it becomes indispensable in reality [1].

The basic WTA problem has been proved to be NP-complete. It has two versions: static WTA (SWTA) and dynamic WTA (DWTA). In SWTA, all weapons engage targets in a single stage. On the contrary, DWTA is a multi-stage problem and the outcomes of previous stages must have effects on the subsequent decision making. The feasibility constraints which are primarily related to the time window of weapons and targets are main features for distinguishing DWTA from SWTA. The feasibility constraints greatly increase the complexity of the DWTA problem. The goal of DWTA is to find a global optimal assignment for the whole combat process, or equivalently, to allocate a right amount of weapons at appropriate stages [2].

In the past, most literatures on WTA only considered optimizing the total expected damage to targets regardless of ammunition consumption and time consumption in a combat situation. While in an actual combat situation, a proper WTA should also minimize ammunition and total operational time consumption [3]. In this paper, we model WTA as a three-objective constrained optimization problem. Then the prevailing evolutionary multi-objective optimizer – NSGA-II is employed to solve the problem. An advantage of this approach is that it is able to provide an entire front of approximate Pareto optimal solutions, therefore benefiting human decision-makers to have a global view of candidate desirable WTA schemes.

In previous research, J. L. Guo [3], Q. Tang [4] and D. P. Lötter [5] all modeled the weapon-target (WTA) problem as a multi-objective problem (MOP) and utilized NSGA-II to solve it. Among them, Q. Tang et al. regarded minimizing shooting time as an objective; D. P. Lötter proposed to take the cost of weapons into account and solved it via three approaches, including analytical hierarchical process (AHP), functional utility theory and the NSGA-II. However, these studies are all about SWTA problems.

The scenario considered in this paper is delineated as follows. At certain time, the defender detects T hostile targets, and the defender has W weapons to intercept the targets. Besides, before these offensive targets break through the de-

Corresponding author: XIN Bin (brucebin@bit.edu.cn)

This work was supported by the National Natural Science Foundation of China (61304215), Projects of Major International (Regional) Joint Research Program NSFC (Grant No. 61120106010), Beijing Outstanding Ph.D. Program Mentor (Grant No. 20131000704), Beijing Education Committee Cooperation Building Foundation Project (CSYS 100070417), Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61321002), Program for Changjiang Scholars and Innovative Research Team in University (Grant No. IRT1208).

fense and escape, there are at most S stages available for the defender to use its own weapons to hit the targets. In the DWTA literature, most researchers often adopt a Shoot-Look-Shoot (SLS) engagement policy which is a tradeoff between defense effects and defense costs [6].

The contribution of this paper is threefold. Firstly, apart from maximizing damage to hostile targets, this paper regards minimizing ammunition consumption and total operational time as two other objectives. Secondly, this paper first proposes the fire transfer constraint which makes the DWTA model more realistic. Thus, a three-objective constrained combinational optimization model is formulated for the DWTA problem. At last, a modified NSGA-II which is fit for the specific structure of the problem is proposed to achieve efficient problem solving.

This paper is organized as follows. Section 2 models DWTA as a multi-objective constrained combinatorial optimization problem. Section 3 introduces the multi-objective optimizer NSGA-II, and also presents its adaptation to DWTA problem solving. Simulation results are illustrated in section 4. Section 5 concludes the paper.

2 Problem Formulation

DWTA models depend on many factors, such as defense strategies and features of targets and weapons, so different combat scenarios need different models. The scenario considered in this paper is described as above, and next we model it into the mathematical form.

2.1 Objective Function 1 – Damage Efficiency

Hosein and Athans proposed two versions of WTA models: asset-based model and target-based model [2]. This paper mainly focuses on the target-based WTA. Firstly, a WTA decision should be made to destroy as many targets as possible. The formulation of the objective function for stage t is shown as follows:

$$D_t(X^t) = \sum_{j=1}^{T(t)} v_j (1 - \prod_{s=t}^{S} \prod_{i=1}^{W(t)} (1 - p_{ij}(s))^{x_{ij}(s)}) \quad (1)$$

where t and s are the indexes of defense stages, $X^t = [X_t, X_{t+1}, ..., X_s]$ with $X_t = [x_{ij}(t)]_{W \times T}$ is the decision matrix at stage t, and $x_{ij}(t)$ is a binary decision variable taking a value of one (i.e., $x_{ij}(t) = 1$) if weapon i is assigned to target j at stage t, or zero (i.e.,$x_{ij}(t) = 0$) otherwise. $W(t)$ and $T(t)$ represent the remaining number of the weapons and targets at stage t, respectively ($W(1) = W, T(1) = T$). v_j means the threat value of target j. $p_{ij}(s)$ denotes the probability that weapon i destroys target j at stage s, which is also called kill probability.

2.2 Objective Function 2 – Ammunition Consumption

How to achieve the goal of minimizing ammunition cost without losing damage efficiency is what we are after. Ammunition consumption is calculated under the premise of given targets and desired damage degree [1, 5], so we choose the overall cost of ammunition as the second objective.

$$C_{ammu} = \sum_{s=1}^{S} \sum_{j=1}^{T} \sum_{i=1}^{W} \beta_i u_{ij}(s) x_{ij}(s) \quad (2)$$

where $u_{ij}(s)$ denotes the ammunition consumption when weapon i is allocated to target j at stage s,β_i represents the unit economic cost of the ammunition that weapon i consumes. If all weapons are the same, β_i is assumed to be constant (e.g., one unit).

2.3 Objective Function 3 – Total Operational Time

On the basis of the definition of WTA, the final decision should contain weapon-target assignment and fire time scheduling [1]. However, the aforementioned WTA decision is restricted within certain time window in which only a part of weapons are available to engage the targets [6], and the issue of shooting time is left to human decision makers. In general, the sooner the targets are shot, the better the combat effect will be. Therefore, the time consumption through all stages is taken as the third objective [4, 7].

$$C_{time} = \sum_{s=1}^{S} \sum_{j=1}^{T} \sum_{i=1}^{W} t_{ij}(s) x_{ij}(s) \quad (3)$$

where $t_{ij}(s)$ denotes the time that weapon i takes to destroy target j at stage s. Besides, only when certain damage degree is reached, can each target be considered as destroyed. So for convenient comparison, if weapon i can not destroy target j when assigned to it at stage s, we set $t_{ij}(s)$ as $+\infty$ [7].

The kill probability matrix $P(s) = [p_{ij}(s)]_{T \times W}$, ammunition consumption matrix $U(s) = [u_{ij}(s)]_{T \times W}$ and time consumption matrix $Time(s) = [t_{ij}(s)]_{T \times W}$ all can be calculated under given combat situations.

The three objectives are conflicting in the sense that when assigning more weapons to a target, the expected damage efficiency will be higher, but this will, in turn, result in a higher overall ammunition consumption and time cost. Hence, the aim is to find an acceptable trade-off between the cost and the efficiency.

2.4 Constraints

1) Resource Constraints

$$\sum_{i=1}^{W} x_{ij}(t) \le m_j \quad \forall j \in I_j, \forall t \in I_t \quad (4)$$

$$\sum_{j=1}^{T} x_{ij}(t) \le n_i \quad \forall i \in I_i \quad (5)$$

$$\sum_{j=1}^{T} \sum_{t=1}^{S} x_{ij}(t) \le N_i \quad \forall i \in I_i \quad (6)$$

$$I_i = \{1, 2, ..., W\}; I_j = \{1, 2, ..., T\}; I_t = \{1, 2, ..., S\}$$

Constraint (4) limits the ammunition cost for each target at each stage. Constraint (5) reflects the capability of weapons of firing at multiple targets at the same time. Actually, most weapons can fire only one target, while for special cases, the value of n_i may be larger than two. In these cases, these weapons can be regarded as n_i independent weapons, so it is assumed that $n_i = 1, \forall i \in I_i$. Constraint (6) indicates the amount of available ammunitions of weapon i.

2) Feasibility Constraints

$$x_{ij}(t) \le f_{ij}(t) \quad \forall i \in I_i, \forall j \in I_j, \forall t \in I_t \quad (7)$$

Where $f_{ij}(t)$ is a binary variable. If weapon *i* can not shoot target *j* at stage *s* for various reasons (e.g., the targets being beyond the range of the weapon), then $f_{ij}(t) = 0$, otherwise $f_{ij}(t) = 1$.The constraints are important features of DWTA against SWTA,and the feasibility matrix should be updated after each stage. Besides, they also increase the complexity of DWTA and the difficulty of generating feasible solutions.

3) Fire Transfer Constraints

The fire transfer constraints widely exist in actual operations, especially when one weapon is scheduled to hit multiple targets in a given order. Fire transferring is mainly caused by position and direction changes of targets. The exact definition of fire transferring time is the minimum time required from the end of shooting in one direction to the start of shooting in another direction. Fire transferring time is an important indicator of the mobility and continuous salvo capability of the fire power. If the time interval between two contiguous engagements is less than the corresponding fire transferring time, at most only one of the two engagements can be implemented.

Taking the situation involving *W* weapons, *T* targets, and *S* stages as an example to illustrate fire transferring, we define the fire transfer feasibility matrix as:

$$FTF = [ftf(w_i, s_m, s_n, t_j, t_k)]$$

where $ftf(w_i, s_m, s_n, t_j, t_k)$ is a binary variable. $ftf(w_i, s_m, s_n, t_j, t_k) = 0$ means that when weapon w_i is shooting target t_j at stage s_m (i.e., $x_{w_i t_j}(s_m) = 1$), the weapon can not be used to shoot target t_k at stage s_n (i.e., $x_{w_i t_k}(s_n) = 0$) and vice versa. $ftf(w_i, s_m, s_n, t_j, t_k) = 1$ means no such restrictions.

$FTF:$	w_1				$\cdots$	w_W
	(s_1, s_2)	(s_1, s_3)	$\cdots$	(s_{n-1}, s_n)	$\cdots$	$\cdots$
(t_1, t_2)	1	0		1		
(t_2, t_1)	0	1		0	$\cdots$	$\cdots$
(t_1, t_3)	1	1		1		
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
(t_{T-1}, t_T)	0	0		1		
(t_T, t_{T-1})	1	0		1	$\cdots$	$\cdots$

After giving the following fire transfer feasibility matrix FTF, the fire transferring constraint can be formulated as follows:

$$x_{w_i t_j}(s_m) x_{w_i t_k}(s_n) \leq ftf(w_i, s_m, s_n, t_j, t_k) \qquad (8)$$

which indicates that weapon w_i has a fire transfer constraint for targets t_j at stage s_m and target t_k at stage s_n.

Remark:

In contrast to feasibility constraints which limit the use of individual weapon to engage specific targets at specific stages, the fire transfer constraints impose restrictions on different operations of the same weapon (especially those aimed at different targets).

From above, the optimization model for the abovementioned DWTA problem can be formulated as follows:

$$\left\{ \begin{array}{l} \max \ f_1 = D_t(X^t) \\ \min \ f_2 = C_{ammu} \\ \min \ f_3 = C_{time} \end{array} \right.$$
$$s.t.(4), (5), (6), (7), (8)$$

3 NSGA-II

Kalyanmoy Deb et al. proposed the non-dominated sorting genetic algorithm with elitist strategy (i.e., NSGA-II) which incorporates an elitism strategy and uses crowding distance to maintain population diversity [1, 8].

3.1 Non-dominated Sorting

For each individual *p* of population *P*, we calculate the above-mentioned three-dimensional objective vector $\vec{f} = [\ D_t(X^t) \quad C_{ammu} \quad C_{\cos t}\ ]$, and initialize two entities $S_p = \emptyset, n_p = 0$ where S_p is composed of all individuals that are being dominated by individual *p*; n_p is the number of individuals which dominate individual *p*. Any individual in the first non-dominated front will have $n_p = 0$. Next we visit each individual $q \in S_p$ with $n_p = 0$ and reduce the corresponding n_p by one. After doing that, individuals with $n_p = 0$ will belong to the second non-dominated front. The lower non-dominated rank an individual has, the higher superiority the individual will have.

3.2 Diversity Preservation Strategy – Crowding Distance

It is desired that an evolutionary algorithm (EA) maintains a good spread of solutions. So, NSGA-II adopts the concept of crowding distance to maintain population diversity and keep a good spread of solutions in the objective space [8].

1) Crowding Distance

For a certain individual, we calculate the average distance of two points on either side of this point along each of the objectives as the crowding distance [4, 8]. Besides, the crowding distance between two individuals in different fronts is meaningless. We take bi-objective as an example to illustrate the geometric explanation of crowding distance,as shown in Fig. 1. The computational complexity of the above process is $O(M \cdot N \cdot logN)$ [8].

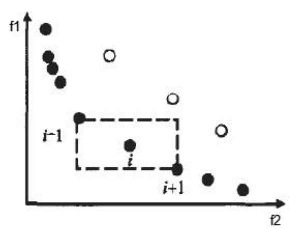


Fig. 1: Crowding distance calculation in the bi-objective situation

2) Crowding Distance Comparison Operator

After the non-dominated sorting, each individual *i* in the population is assigned two metrics: p_{rank} and $d_p(F_{p_{rank}})$, where p_{rank} refers to the non-dominated rank of individual *p*, and $d_p(F_{p_{rank}})$ denotes the crowding distance of individual *p* in the $p_{rank}th$ front. Then define the comparison operator $\prec$:

$$\begin{aligned} i \prec j \Leftrightarrow & [(i_{rank} < j_{rank}) \\ & or(\ i_{rank} = j_{rank} \ and \ d_i(F_{i_{rank}}) > d_j(F_{j_{rank}}))] \end{aligned}$$

which means as to two individuals with different non-dominated ranks, we prefer individuals with a smaller non-dominated rank. Otherwise, if they locate in the same front, individuals with greater crowding distance will be selected.

3.3 Encoding

This paper adopts the decimal encoding. The length of a chromosome is the number of weapons. Each weapon is regarded as a genetic locus to form a chromosome, and the genic value of each genetic locus indicates the number of the target to which the weapon is assigned. Take the following chromosome ($W = 5, T = 6$) as an example to illustrate the encoding method shown in Fig. 2.

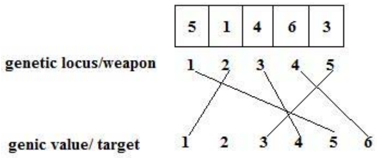


Fig. 2: Encoding scheme

Accordingly, the chromosome corresponds to a decision matrix *X* as follows:

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

3.4 Genetic Operators

All individuals are sorted based on the non-domination and with crowding distances assigned, whereafter, genetic operations should be performed. This paper adopts the binary tournament selection based on comparison operator $\prec$ [9]. Then the inversion recombination and random mutation operators are designed to generate an offspring population *Q* of size *N*, in accordance with specific conditions. The inverse recombination operator means reversing genic values between two randomly selected genetic loca, which will form a new chromosome and guarantee the satisfaction of three resource constraints. Random mutation is a simple but effective way of keeping the diversity of the population. In the multi-objective DWTA problem situation, a genetic locus is randomly chosen and the corresponding value is replaced with a random number in the range of $\{0, 1, ..., T\}$ with a user-defined mutation probability.

In the proposed NSGA-II algorithm, the elitism feature is ensured [1, 5, 8], since all the current and previous best individuals will survive. The pseudo code of NSGA-II is presented in Fig. 3.

Main loop of NSGA-II [9], [11]

```
NSGA-II ()
Randomly initialize a parent population P_0
    P_0 = (F_1, F_2, ...) =non_domination_sort_mod( P_0 )
For each F_i ∈ P_0
    Crowding_distance_assignment( F_i )
    t=0
    while (1)
    Use selection, crossover and mutation to create a new population Q_t .
        R_t = P_t ∪ Q_t // Elitism Strategy //
        F = (F_1, F_2, ...) =non_domination_sort_mod( R_t )
        Let P_{t+1} = ∅, i = 1
        While(| P_{t+1} | + | F_i |< N )
            Crowding_distance_assignment( F_i )
            P_{t+1}=P_t ∪ F_i, i = i + 1
        End
        P_{t+1}=P_{t+1} ∪ F_i[1: (N−| P_{t+1} |)]
        t = t +1
    End
Return F_i
End
```

Fig. 3: Peseudo-code of NSGA-II

As to the DWTA problem considered in this paper, there is a cycle computation: **Decision Making** $\rightarrow$ **Damage Assessment** $\rightarrow$ **Decision Making**. Specifically, a WTA solution is obtained at certain stage, after that there will be a damage assessment procedure during which there may occur a number of new targets or some old targets exit. Following with that, a new decision making process will be triggered except the computational complexity is decreased since the reduction of the numbers of weapons and targets.

4 Simulation Results

4.1 Parameter Settings

In this section, we first randomly generate a set of required data which largely accord with real combat situations. And three types of data for different combat situations, i.e. small, medium and large scale cases are used to compare the performance of NSGA-II with that of the Monte Carlo method.

For each situation we generate the $P(s)$, $U(s)$, $Time(s)$, $F(s)$ and $FTF(s)$, and each element of these matrices is randomly generated within a given range related to the problem scale. Then the value vector of targets and other parameters are given. In what follows, we take the small-scale situation as an example to illustrate the performance of NSGA-II and the Monte Carlo method.

The related parameters of small-scale WTA problems are as follows: the number of targets, weapons and stages are 5, 3 and 3, respectively; the threat value vector of hostile targets is $v = [0.93, 0.82, 0.65, 0.61, 0.57]$ [1]; $n_i = [1, 1, 1], m_j = [2, 2, ..., 2]_{1\times 15}, N_i = [2, 2, 2]$. For NSGA-II, the population size is 100, the maximum number of iterations is 150, and the crossover probability and mutation probability are 0.9 and 0.05, respectively. Besides, the following also introduces medium-scale and large-scale DWTA instances. The numbers of targets, weapons and stages are 12/50, 20/50 and 5/8 in medium/large-scale cases, respectively.

4.2 Performance Metrics

In order to assess the performance of multi-objective optimization algorithms, the inverted generational distance (*IGD*) metric [10] are frequently used. For instance, the *IGD* metric is computed as follows:

$$IGD(P^*, P) = \frac{\sum\limits_{v \in P^*} d(v, P)}{|P^*|}$$

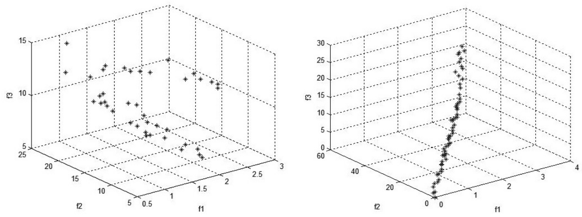
where P^* is a set of uniformly distributed points in the objective space along the true Pareto front (PF) or nearly true PF when it is hard or impossible to get the true PF; *P* is an approximate PF of the true PF; $d(v, P)$ is the minimum Euclidean distance between *v* and elements in *P*, and $|P^*|$ represents the number of points in P^*. If $|P^*|$ is large enough to represent the true PF well, $IGD(P^*, P)$ could measure both the diversity and convergence of *P* in a sense. As to the value of $IGD(P^*, P)$, the lower the better.

4.3 Experimental Results

1) Effectiveness of NSGA-II

The distributions of initialized solutions and near-optimal solutions (i.e., near-optimal Pareto front) obtained via the NSGA-II are shown in Fig. 4(a) and Fig. 4(b), respectively.

Fig. 4(a) and Fig. 4(b) visually show that with the increase of iterations, the spread of optimal solutions becomes convergent. Since the three objective functions are competing with each other, there is no one solution that can optimize them simultaneously. What we except to get is a set of optimal solutions named Pareto optimal solutions. Finally, the decision maker can choose a solution in the Pareto front according to their own preference.



(a) Initial distribution (b) Final distribution

Fig. 4: Solution distribution in objective space under small-scale cases via NSGA-II

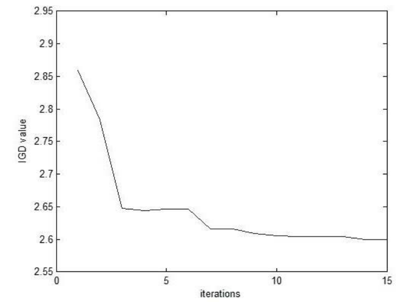


Fig. 5: The track of IGD values with the increase of iterations under small-scale cases via NSGA-II

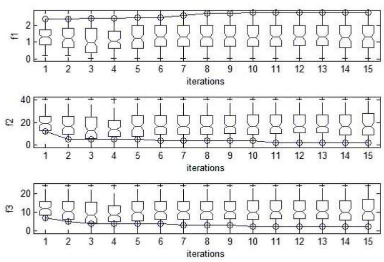
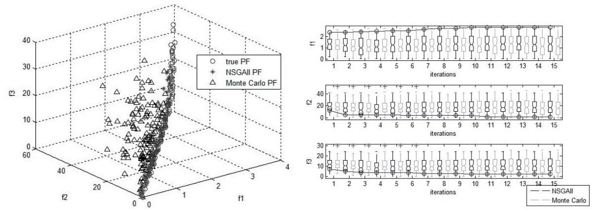


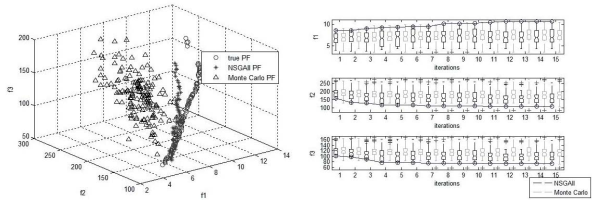
Fig. 6: Boxplots of three objective functions under small-scale cases via NSGA-II

The inverted generational distance ($IGD(P^*, P)$) metric is adopted to measure both the diversity and the convergence of the proposed algorithm. As to the DWTA problems, it is hard to obtain the true PF. To address this issue, we run each instances for 20 times, execute the non-dominated sorting on these solutions obtained earlier and select 500 evenly distributed points in Pareto front as the true Pareto front P^*. For visual convenience, the *IGD* values during the first 15



(a) Solution distribution in objective space (b) Boxplots of three objective functions

Fig. 7: Comparison results under small-scale cases via NSGA-II and the Monte Carlo method



(a) Solution distribution in objective space (b) Boxplots of three objective functions

Fig. 8: Comparison results under medium-scale cases via NSGA-II and the Monte Carlo method

iterations are recorded, and the evolution of *IGD* values with the number of iterations is presented in Fig. 5. Besides, we present the box-plot of the three objectives with the number of iterations as shown in Fig. 6. In Fig. 6, the middle line and the notches on each box represent the median and a robust estimate of the uncertainty of a set of solutions. The track of *IGD* values in combination with the box-plot can confirm the convergence of NSGA-II.

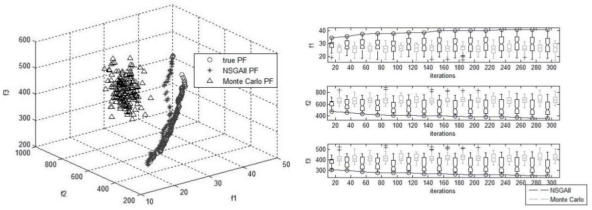
2) Superiority of NSGA-II over the Monte Carlo method

The Monte Carlo method is employed as a comparison against the NSGA-II to demonstrate the superiority of NSGA-II in solving DWTA problems. The Monte Carlo method is achieved by combing individuals with smaller non-dominated rank with random generated ones together and selecting a group of individuals with smaller non-dominated rank. We execute 10 independent runs of both NSGA-II and the Monte Carlo method and record the mean of *IGD* values as shown in Table 1.

Table 1: The mean *IGD* values of NSGA-II and the Monte Carlo method

Method / Scale	NSGA-II	Monte Carlo
Small-scale	**0.1810**	2.4370
Medium-scale	**4.6475**	6.7100
Large-scale	**5.8918**	84.6086

Fig. 7, Fig. 8 and Fig. 9 demonstrate that the approximate Pareto front obtained via NSGA-II is closer to the nearly true Pareto front and spreads more evenly compared with that obtained by the Monte Carlo method. Besides, these box-plots and Fig. 10 all confirm the convergence of NSGA-II and its better performance than the Monte Carlo method. Table 1 shows that the NSGA-II has a better performance for the DWTA instances under three scales with comparison to the



(a) Solution distribution in objective space (b) Boxplots of three objective functions

Fig. 9: Comparison results under large-scale cases via NSGA-II and the Monte Carlo method

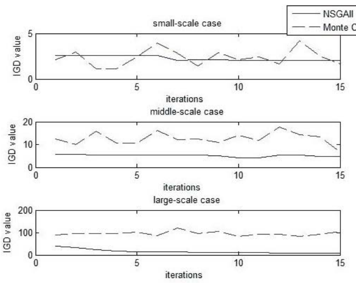


Fig. 10: The track of *IGD* values via two method

Monte Carlo method because it obtains lower *IGD* values.

3) Time performance analysis

The primary aim of solving DWTA problems is to provide a set of near-optimal or acceptable real-time decisions for aiding operational command, so the time performance of algorithms has to meet the actual requirements.

Table 2: The mean of time cost of NSGA-II under three scales (unit: second)

Scale / Time	Small	Medium	Large
Total	6.16700	20.82044	91.17560
Non-dominated sorting/per	0.02376	0.03494	0.10202
Non-dominated sorting /total	3.56802	10.47862	30.61572
Genetic operators/per	0.01496	0.02822	0.16622
Genetic operators/total	2.24436	8.45712	49.86444

Table 2 demonstrates that the total time required for DWTA problem solving increases greatly with the increase of problem size. In more specific terms, the time required for the non-dominated sorting (including the selection) and genetic operators (only including the recombination and the mutation) per iteration are almost the same for the three scales. The total time cost of the non-dominated sorting is slightly larger than that of genetic operators.

All experiments are performed on DELL PC with Intel Core i5-2400 CPU, 3.10 GHz and 4.0 GB of RAM via MATLAB language. In real combat situation, the time consumption should be limited within seconds or even within milliseconds. Apparently, the data shown in Table 2 is unqualified. However, this problem can be settled via two aspects: firstly, we can use lower level languages such as C++ which may obtain better time performance. Secondly, employing parallel computations on high-performance hardware platform such as FPGA, which can also reduce the processing time in order to satisfy the real-time requirement.

5 Conclusions and Future Works

In this paper, the DWTA problem is formulated into a multi-objective optimization problem. And apart from traditional resource constraints, the fire transfer constraints are introduced, which makes the model more practical. Then the NSGA-II is adopted according to the specific structure of the problem to achieve efficient problem solving. Finally, the numerical simulations with instances under three scales are performed to analyze the feasibility of the NSGA-II. Besides, The NSGA-II is compared with the Monte Carlo method to illustrate its superiority. The computational results show that the NSGA-II can effectively find the approximate Pareto front within acceptable time and provide desirable supports for the commanders.

In the future, we may focus on another kind of multi-objective evolutionary algorithms (MOEAs) – MOEA/D (Multi-objective Evolutionary Algorithm based on Decomposition) which is a decomposition-based one rather than the domination-based. The comparative performances of NSGA-II and MOEA/D on solving the DWTA problems will be studied in the future works.

References

[1] H. B. Cui. Study on firepower optimal distribution of artillery group based on NSGA-II. Master's thesis, Changsha, Hunan, P. R. China: National University of Defense Technology, 2010. (in Chinese).

[2] B. Xin, J. Chen, J. Zhang, L. H. Dou, Z. H. Peng. Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 2010, 40(6): 649-662.

[3] J. L. Guo, P. Fu. Multi-objective optimization of weapon-target assignment (WTA) using NSGA-II. *Fire Control & Command Control*, 2010, 35(3): 1-5. (in Chinese).

[4] Q. Tang, X. Yang. Air defense fire distribution based on NSGA-II and multi-attribute decision-making method. *Command Control & Simulation*, 2011, 33(1): 30-34.

[5] D. P. Lötter. Modeling weapon assignment as a multiobjective decision problem. Master's Thesis, Department of Logistics, Stellenbosch University, 2012.

[6] B. Xin, J. Chen, Z. H. Peng, L. H. Dou, J. Zhang. An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2011, 41(3): 598-606.

[7] Y. X. Wang. Air defense firepower assignment model and algorithm under the principle of earlier damage. Master's Thesis, Nanjing, P. R. China: Nanjing University of Science and Technology, 2008. (in Chinese)

[8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.

[9] R. B. Agarwal, K. Deb, R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 1995, 9(6): 115-148.

[10] Q. Zhang, H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.