Efficient Multi-objective Evolutionary Algorithms for Solving the Multi-stage Weapon Target Assignment Problem: A Comparison Study

Juan Li [1], Jie Chen [2], Bin Xin[3], and Lu Chen [4]
School of Automation
Beijing Institute of Technology
State Key Laboratory of Intelligent Control and Decision of Complex Systems
Beijing, P.R.China
[1]00134476@bit.edu.cn; [2]chenjie@bit.edu.cn; [3*] brucebin@bit.edu.cn; [4]3120140357@bit.edu.cn;

***Abstract*—The weapon target assignment (WTA) problem is a fundamental problem arising in defense-related applications of operations research. The multi-stage weapon target assignment (MWTA) problem is the basis of the dynamic weapon target assignment (DWTA) problem which commonly exists in practice. The MWTA problem considered in this paper is formulated as a multi-objective constrained combinatorial optimization problem with two competing objectives. Apart from maximizing the damage to hostile targets, this paper follows the principle of minimizing the ammunition consumption. Decomposition and Pareto dominance both are efficient and prevailing strategies for solving multi-objective optimization problems. Three competitive multi-objective optimizers: DMOEA-εC, NSGA-II, and MOEA/D-AWA are adopted to solve multi-objective MWTA problems efficiently. Then comparison studies among DMOEA-εC, NSGA-II, and MOEA/D-AWA on solving three different-scale MWTA instances are done. Three common used performance metrics are used to evaluate the performance of each algorithm. Numerical results demonstrate that NSGA-II performs best on small-scale and medium-scale instances compared with DMOEA-εC and MOEA/D-AWA, while DMOEA-εC shows advantages over the other two algorithms on solving the large-scale instance.**

Keywords*—*multi-stage weapon target assignment (MWTA); multi-objective constrained optimization problem; multi-objective optimization; decomposition; ε-constraint; combinatorial optimization

I. INTRODUCTION

The weapon target assignment (WTA) problem is a fundamental problem arising in defense-related applications of military operations research. It deals with the problem of how to obtain a weapon-target pair or a set of weapon-target pairs that meet decision makers' operational goals regarding combating effects and expenditures [1]. The WTA problem is a classical constrained combinatorial optimization problem and has been proved to be NP-complete [2]. Hosein et al. [3-5] grouped the WTA problem into two categories: the static WTA (SWTA) problem and the dynamic one (DWTA). In SWTA, all weapons engage with targets in a single stage. On the contrary, DWTA is a multi-stage problem where some weapons engage with targets at one stage and then the strategy for the next stage is decided based on the assessment of the former stage. The DWTA problem is a global decision-making process. It takes the whole defense effects through all stages into account, incorporates the concept of time windows. A time window is a time interval that spans from the time that a certain target reaches the combat zone of a weapon to the time that the target escapes from the weapon's combat zone. In actual combat situations, targets come from different directions at different speeds, thus each weapon-target-stage pair will be assigned a time window. The term "stage" here is regarded as the period that spans from the beginning to the completion of hitting a batch of targets. Since the battle situation may change with the arrival of new incoming targets and changes of intents or states of existing targets, time windows for the same weapon-target pair vary with stages, which is a typical feature that distinguishes SWTA and DWTA.

DWTA is much more complicated than SWTA. Previous works are mainly focused on SWTA, while the information age entails DWTA in modern and even future warfare. In real combat situations, after making decisions at one stage, there will be a damage assessment during which a number of new targets may appear or some old targets may exit [6]. Following that, a new decision making process of the next stage will be triggered. This process is the same as the previous one, except that the computational complexity decreases due to the reduction of the number of weapons and targets. Thus, there is a cyclic computation process: "Decision Making → Damage assessment → Decision Making" in an actual DWTA process. The multi-stage weapon target assignment (MWTA) problem falls between the SWTA one and the DWTA one. It also takes time windows into account but does not possess the dynamic process like the DWTA does. In a word, the multi-stage version of the WTA problem lays the foundation of the dynamic one.

According to different combat situations and missions, the WTA problem can be categorized into two types: the target-based one and the asset-based one. The aim of the target-based WTA problem is to maximize the expected damage of the targets, while the goal of the asset-based one is to minimize the expected loss of protected assets. There is no essential distinction between the two types, and the target-based model can be regarded as a special case of the asset-based one. The target-based one is taken into consideration in the following.

The objective of a traditional WTA problems is focused on operational effects, such as maximizing the expected damage of targets and minimizing the expected loss of protected assets. However, a proper WTA scheme should not only satisfy certain operational requirements, but also minimize the operational

This work was supported in part by the National Natural Science Foundation of China under Grant 61673058, 61304215 and U1609214, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61321002, in part by the National Science Fund for Distinguished Young Scholars under Grant 60925011, in part by the Projects of Major International (Regional) Joint Research Program NSFC under Grant 61120106010, in part by the Beijing Outstanding Ph.D. Program Mentor under Grant 20131000704, and in part by the Research Fund for theDoctoral Program of Higher Education of China under Grant 20131101120033. Corresponding Author: Bin Xin (brucebin@bit.edu.cn)

cost. The two goals are conflicted in the sense that the more ammunition cost is, the better the operational effect is. Thus the WTA problem in practice is a multi-objective optimization problem (MOP).

Various multi-objective evolutionary algorithms (MOEAs) have been widely accepted as major approaches for approximating the true Pareto front (PF) for MOPs [7-8]. Based on the selection strategy, these algorithms can be categorized into three classes: Pareto dominance-based [9-11], performance indicator-based [12-13], and decomposition-based [14-20]. Among them, the Pareto dominance-based approaches optimize conflicting objectives simultaneously by domination behaviors on solutions and are the most classical ones. The decomposition-based approaches decompose an MOP into a series of single-objective subproblems and optimize them concurrently. They are growing in popularity and become major methodologies for the multi-objective optimization thanks to their good properties. In this paper, DMOEA-εC [21], NSGA-II [10], and MOEA/D-AWA [18] are adopted to solve the multi-objective MWTA problems.

Similar to our previous research on the MWTA problem [23-24], this paper also models it as a two-objective constrained problem. The aim of this paper is to apply three state-of-the-art multi-objective optimizers, including DMOEA-εC, NSGA-II, and MOEA/D-AWA, to solve the proposed multi-objective MWTA problems and present a comparison study. The contribution of this paper are in the following aspects:

- DMOEA-εC is a newly proposed competitive multi-objective evolutionary algorithm in our recent work. It has been compared with some advanced MOEAs on several continuous and 0/1 knapsack test problems and has shown its advantages over the other approaches. However, the MWTA problem is far more complicated than these benchmark problems. Thus this paper intends to demonstrate the effectiveness of DMOEA-εC on solving the MWTA problems. And we compare it with NSGA-II and MOEA/D-AWA on solving three different-scale MWTA instances.
- Numerical results demonstrate that NSGA-II performs best on small-scale and medium-scale instances compared with DMOEA-εC and MOEA/D-AWA, while DMOEA-εC shows advantages over the other two algorithms on solving the large-scale instance. Besides, NSGA-II runs about twice as fast as DMOEA-εC and MOEA/D-AWA with the same number of function evaluations for the same scale problem on average, and DMOEA-εC runs slightly faster than MOEA/D-AWA.

The rest of this paper is organized as follows. In Section II, the mathematical formulation of the multi-objective MWTA problem is given. Section III provides algorithmic descriptions of DMOEA-εC, NSGA-II, and MOEA/D-AWA. Section IV describes test instances and parameter settings of MWTA problems and presents comparison results about DMOEA-εC against NSGA-II and MOEA/D-AWA on the three different-scale MWTA instances. Section V concludes the paper and points out future works.

II. DESCRIPTION AND MATHEMATICAL FORMULATION OF MWTA PROBLEMS

Models of WTA problems depend on many factors, e.g., defense strategies and features of targets and weapons. The scenario considered in this paper is delineated as follows. At a certain time, the defender detects T hostile targets and has W weapons to intercept targets. Besides, before these offensive targets break through the defense and escape, there are at most S stages available for the defender to use its own weapons to hit the targets. The above-mentioned combat scenario is very common, e.g., in air-defense-oriented naval group combating.

A. Objective Functions

Given the set of targets and the set of available weapons, one aims to find the optimal assignment of weapons to targets, such that the damage to the targets is maximized and the cost of operations is minimized. The MWTA problem is formulated as a multi-objective problem from the perspective of target-based models under the above-described scenario.

1) Expected damage of targets

Based on characteristics of the target-based model, the first objective is to maximize the total expected damage on coming targets through all stages. The formulation of the expected damage at stage t is expressed as:

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

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 i at stage t, or zero (i.e., $x_{ij}(t) = 0$) otherwise. $W(t)$ and $T(t)$ represent the remaining number of 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. The threat value vector $\vec{v} = [v_j]_{1 \times T}$ and the kill probability matrix at any stage s, denoted by $P(s) = [p_{ij}(s)]_{T \times W}$, can be obtained in advance based on the theory of shooting and performances of weapons.

2) Ammunition consumption

Apart from satisfying the tactical requirement, a WTA decision should also cut down the operational costs. Thus, the goal of minimizing over all ammunition consumption through all stages is regarded as the second objective function.

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

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 are assumed to be constant (e.g., one unit). The economic cost vector $\vec{\beta} = [\beta_i]_{1 \times W}$ and the ammunition consumption matrix corresponding to stage s denoted by $U(s) = [u_{ij}(s)]_{T \times W}$ should be given in advance.

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

B. Model Constraints

1) Resource constraints

$$\sum_{i=1}^{W} x_{ij}(t) \le m_j \quad \forall j \in I_j, \forall t \in I_t \tag{3}$$

$$\sum_{j=1}^{T} x_{ij}(t) \le n_i \quad \forall i \in I_i, \forall t \in I_t \tag{4}$$

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

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

Constraints (3)-(5) are resource constraints. Constraint (3) means that at each stage at most m_j weapons can be used to destroy target j. It limits the ammunition cost for each target at each stage. The value of m_j usually depends on the performance of available weapons and the tactical preferences of commanders. Constraint (4) reflects the capability of weapons of firing at multiple targets at the same time. To be more precise, weapon i can fire at most n_i targets at the same time. Actually, most weapons can fire at only one target, while for special cases, e.g., in artillery-based defense systems, the value of n_i may be larger than two. In these special cases, weapons can be regarded as n_i independent weapons, so it is assumed that $n_i = 1, \forall i \in I_i$. Constraint (5) indicates the amount of available ammunitions of weapon i.

2) Feasibility constraints

$$x_{ij}(t) \le f_{ij}(t) \quad \forall i \in I_i, \forall j \in I_j, \forall t \in I_t \tag{6}$$

where $f_{ij}(t)$ is a binary variable. If weapon i can not shoot target j at stage t for various reasons (e.g., the target is being beyond the range of the weapon), then $f_{ij}(t) = 0$, otherwise $f_{ij}(t) = 1$. This constraint is an important feature of MWTA against SWTA, since MWTA considers the influence of time windows on the engagement feasibility of weapons. Due to the time-dependent property of the engagement feasibility, the feasibility matrix $F(s) = [f_{ij}(s)]_{W \times (T \times S)}$ should be updated after each stage.

Based on the above description, the multi-objective MWTA model is formulated as follows:

$$P1: \begin{cases} \min \quad f_1 = 1/D_t(\mathrm{X}^t) \\ \min \quad f_2 = C_{ammu} \end{cases}$$

$$s.t. \quad (3),(4),(5) \text{ and } (6)$$

III. HEURISTIC ALGORITHMS

Numerous classical intelligent optimization algorithms have been applied to solve WTA problems whether it is static or dynamic, single-objective or multi-objective. Three advanced MOP optimizers: DMOEA-εC, NSGA-II and MOEA/D-AWA are adopted to solve the two-objective MWTA problem proposed in this paper.

A. DMOEA-εC

The decomposition-based multi-objective evolutionary algorithm with the ε-constraint framework (DMOEA-εC) [21] is a newly proposed multi-objective solver in our recent work. It has been compared with six state-of-the-art MOEAs on both continuous and 0/1 knapsack benchmark problems and has shown its advantages over competitive approaches. The weighting method and the ε-constraint method are two basic generation methods [22]. They are often used as elements of more developed methods. DMOEA-εC takes inspiration from the ε-constraint method and is a decomposition-based MOEA. The ε-constraint method selects one of the objectives as the main objective, while transforming the other non-main objectives to constraints and associating each non-main objective with an upper bound coefficient. DMOEA-εC explicitly decomposes an MOP into a series of scalar constrained optimization subproblems by selecting one of the objectives as the main objective function and associating each subproblem with an upper bound vector. Let $\boldsymbol{\varepsilon}^{1},...,\boldsymbol{\varepsilon}^{N}$ be a set of evenly spread upper bound vectors, and the neighborhood of an upper bound vector $\boldsymbol{\varepsilon}^{i}$ is defined as a set of its several closest upper bound vectors in $\{\boldsymbol{\varepsilon}^{1},...,\boldsymbol{\varepsilon}^{N}\}$. The neighborhood of the *i*th subproblem consists of all subproblems with the upper bound vectors from the neighborhood of $\boldsymbol{\varepsilon}^{i}$ and is denoted as *B(i)*. These subproblems are optimized collaboratively by an EA based on Deb's feasibility rule [23]. And each subproblem is optimized by using information only from its neighboring subproblems. When two solutions are compared, the following criteria are followed：

1) Any feasible solution is preferred to any infeasible solution.

2) Among two feasible solutions, the one having better objective function value is preferred.

3) Among two infeasible solutions, the one having smaller constraint violation is preferred.

Under the ε-constraint framework, DMOEA-εC tends to retain feasible solutions for each subproblem. A feasible solution usually has good objective values for non-main objectives but bad objective value for the main objective, since these objectives contract each other. Thus it will be bad for the optimization of the main objective function. Thus a main objective alternation strategy is proposed and used to tackle this problem periodically. After the main objective alternation strategy is utilized, a solution which is good for the current subproblem will no longer perform well since the objective function of this subproblem has been changed. Thus a solution-to-subproblem matching proceduree is designed to place the nearest solution to each subproblem. It uses the distance value between a solution and a subproblem as the criterion and is utilized after the main objective alternation strategy. When a new solution is generated, it may perform badly for the current subproblem but perform well for another subproblem. In order to avoid wasting potentially useful solutions and make best use of them, the subproblem-to-solution matching procedure which uses the constraint violation value as the criterion is proposed to find a subproblem with the minimum constraint violation value for the new solution. The two matching procedures strike a balance between convergence and diversity.

In DMOEA-εC, an external archive population (EP) is maintained in addition to the evolving population. Thus when a new solution is generated, the EP should be updated. And if the number of individuals in EP exceeds S, EP is

Algorithm 1 The framework of DMOEA-εC

1: Initialize N evenly spread upper bound vectors; randomly initialize the evolving population $\{\mathbf{x^1},...,\mathbf{x^N}\}$ and set $\mathbf{FV}^i = \mathbf{F}(\mathbf{x}^i)$; extract nondominted individuals and denote the set of them as *EP*; initialize $\mathbf{z}^*$ and $\mathbf{z^{nad}}$; set *gen=0*, *n=N*.
2: Use the solution-to-subproblem matching procedure to match solutions with subproblems.
3: **for** i = 1 to N do
4: Set the neighborhood of the *i*th subproblem *B(i)*.
5: **end for**
6: **if** *gen* is a multiple of IN_m then
7: Alternate the main objective.
8: Use the distance value between a solution and a subproblem to match solutions with subproblems.
9: **end if**
10: **while** $n \le NFE$ do
11: **for** $i \in I$ do
12: $P = \begin{cases} B(\mathrm{i}), & \text{if } rand < \delta \\ \{1, 2, ..., N\}, & \text{otherwise} \end{cases}$
13: Reproduction: select parent individuals from P randomly and apply certain reproduction operator to generate a new solution y.
14: *n=n+1*.
15: Repair: if y is infeasible, repair it.
16: Update the approximated ideal point $\mathbf{z}^*$.
17: Use the subproblem-to-solution mathching procedure to find a subproblem k for y.
18: Compare y with neighboring solutions of the subproblem k and update these neighboring solutions by using the feasibility rule.
19: Update the external archive EP and prune it by using the farthest-candidate approach.
20: Update the approximated nadir point $\mathbf{z^{nad}}$.
21: **end for**
22: *gen=gen+1*.
23: **end while**

pruned until its size equals to S. In order to maintain a good spread of obtained nondominated solutions, several crowded comparison mechanisms have been proposed. The farthest-candidate approach [24] inspired by the best-candidate sampling algorithm [25] in sampling theory is adopted here to prune EP. In the farthest-candidate approach, boundary points (solutions with the minimum and maximum objective values) are selected first. Then the candidate point among the unselected points which is farthest from the selected points is chosen iteratively. In this way, a set of evenly distributed nondominated solutions will be selected from a set of alternative nondomianted solutions.

The following notations will be used in the description of DMOEA-εC:

- N : the number of upper bound vectors, which is the same as the population size;
- T : neighborhood size;
- δ : probability of selecting mate solutions from its neighborhood;
- n_r : maximum number of replacement;
- IN_m : iteration interval of alternating the main objective function;
- S : maximum size of the external archive population;
- NFE : maximum number of function evaluations.

The algorithmic description of DMOEA-εC is presented in Algorithm 1. In this algorithm, *rand* is a uniformly randomly distributed value in [0, 1].

B. NSGA-II

The non-dominated sorting genetic algorithm with elitist strategy (NSGA-II) was firstly proposed by Deb et al. [10]. It is a computationally fast and elitist MOEA based on the non-dominated sorting approach. NSGA-II optimizes all objectives simultaneously by assigning two attributes to each solution i. One is the non-domination rank (i_{rank}) which is based on non-dominated sorting behaviors among solutions of a population F, and the other is the crowding distance ($d_i(F_{i_{rank}})$) which is used to preserve the diversity of the population. With the two attributes, a comparison operator is presented as:

$$i \prec \qquad {}_{rank} < j_{rank}) \\ or(\, i_{rank} = j_{rank} \;\; and \;\; d_i(F_{i_{rank}}) > d_j(F_{j_{rank}}))] \,.$$

Then the selection operator based on the comparison operator is performed on a mating pool. Specifically, combine the parent and offspring populations (namely, the elitist strategy) and select the best (with respect to fitness and spread) solutions. The above strategies guide the selection process at the various stages of the algorithm toward a uniformly spread Pareto optimal front. Readers are suggested to refer to [10] for more details on NSGA-II.

C. MOEA/D-AWA

Recently, the multi-objective evolutionary algorithm based on decomposition (MOEA/D) in which the decomposition idea is applied instead of the classical Pareto dominance relation has achieved a great success in the field of evolutionary multi-objective optimization. It decomposes a MOP into a set of scalar subproblems using uniformly distributed aggregation weight vectors and optimizes them concurrently. Generally, the uniformity of weight vectors in MOEA/D can ensure the diversity of the Pareto optimal solutions based on the assumption that the PF is close to the hyper-plane in the objective space. However, the basic assumption might be violated in the case that the PF of the target MOP is complex, i.e. the PF is discontinuous or has a shape of sharp peak or long tail. Therefore, some studies have been done to refine the weight vectors in MOEA/D [14, 16-18].

An improved MOEA/D with adaptive weight vector adjustment (MOEA/D-AWA) proposed by Qi et al. [18] is one of such studies. An adaptive weight vector adjustment (AWA) strategy is introduced to obtain the uniformly distributed PF of the target MOP. It is natural to have an idea of designing an AWA strategy to regulate the distribution of weight vectors of subproblems by removing subproblems from the crowded regions and adding new ones into the sparse regions periodically. Thus an uniformly distributed PF can be obtained.

Firstly, the AWA removes subproblems located in the crowded regions. The crowd degree of a region is measured by the vicinity distance which evaluates the sparsity level of a solution among the current population. Next, the elite population is deployed as a guidance of helping add new subproblems into the real sparse regions of the complex PF rather than the discontinuous parts which are pseudo sparse regions. If an elite individual is located in a sparse region of the evolving population, it will be recalled into the evolving population and a new weight vector will be generated and added to the subproblem set. MOEA/D-AWA has the same framework as the version of MOEA/D with the dynamic resource allocation (DRA) strategy [16]. In this version, a utility function is defined and computed for each subproblem. Computational efforts are distributed to each of the subproblems based on their utility function values. The major difference between MOEA/D and the suggested MOEA/D-AWA lies on the periodical update of the weight vectors during the search process.

Experimental results indicate that the MOEA/D-AWA outperforms the MOEA/D on solving problems whose PFs are discontinuous, uniform or may be low tail. So in the following, the MOEA/D-AWA with decimal coding is adopted to solve the multi-objective MWTA problems. With references of [14], [16] and [18], readers can have a deep understanding of the MOEA/D-DRA and MOEA/D-AWA. Similar to our previous works [28-29], the two point crossover, one point mutation and the random repair mechanism are also used in DMOEA-εC, NSGA-II, and MOEA/D-AWA.

IV. Numerical Experiments

A. Test Instances and Parameter Settings

Since the actual data is difficult to obtain, in this section, three instances with different problem scales, i.e. small-scale, medium-scale and large-scale cases are randomly generated. The number of weapons/targets/stages in small-scale, medium-scale and large-scale instances are 3/5/3, 20/12/5 and 50/50/8, respectively. The value vector of targets comes from reference [28-29]. For each instance, the weapon kill probability matrix $P(s)$ and the ammunition consumption matrix $U(s)$ are randomly generated within a given range related to the problem scale, which are exactly the same as reference [28-29]. A summary of all the common parameters for each instance is given in TABLE I:

TABLE I. Common Parameter Settings for Three MWTA Instances

Instance	n_i	m_j	N_i	pop	gen
Small-scale	1	1	2	100	100
Medium-scale	1	2	2	100	150
Large-scale	1	2	2	100	200

In addition to these common parameters, the remaining parameters include: the neighbor size is set as T=10, the probability of selecting mate solutions from neighborhood is set as $\delta = 0.9$, the maximal number of replacement is set as $n_r = 1$, the iteration interval of utilizing the dynamic resource allocation strategy is set as 10, the maximal number of subproblems adjusted is set as 1, the ration of the iterations to evolve with only MOEA/D is set as 0.7, the iteration interval of alternating the main objective function is set as $IN_m = 10$, and the size of the external population is set as 150 for all test instances.

B. Performance metrics

No single one performance metric can provide a comprehensive measure on the performance of an MOEA [30]. In the following experimental studies, the following two performance metrics are used.

1) Inverted generational distance (IGD):

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

P^* denotes 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 approximation of the true PF; $d(v, P)$ is the minimum Euclidean distance between v and elements 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. The smaller the value of $IGD(P^*, P)$ is, the better performance of P is. Since a lower value requires that every point in P must be very close to the true PF and cannot miss any part of the whole PF.

2) Set coverage (C-metric):

$$C(P_1, P_2) = \frac{|\{\mu \in P_2 \mid \exists\, v \in P_1 : v \text{ dominates } \mu\}|}{|P_2|}$$

P_1 and P_2 are two approximations to the true PF. $C(P_1, P_2)$ is a binary performance metric and is defined as the percentage of solutions in P_2 that are dominated by at least one solution in P_1.

3) Time performance

The primary aim of solving MWTA problems is to lay a foundation for the DWTA problems whose goal is to provide a set of near-optimal or acceptable real-time decisions for aiding operational command. So in a real application, the time performance of algorithms has to meet the requirement of actual needs. The last part evaluates time performances of DMOEA-εC, NSGA-II, and MOEA/D-AWA.

C. Comparison Results

DMOEA-εC and two competitors (i.e., NSGA-II and MOEA/D-AWA) are tested on three different-scale MWTA instances (small-scale, medium-scale, and large-scale ones) to show the effectiveness of the proposed DMOEA-εC and give a comparison study. DMOEA-εC, NSGA-II and MOEA/D-AWA have been independently run for 30 times for each test instance on a 2.93 GHz Intel (R) Core (TM)2 Duo Processor computer with 2GB RAM PC via MATLAB languages. As mentioned before, *IGD*, *C-metric*, and time performance metrics are used to compare results.

With the purpose of calculating the *IGD* metric value, P^* is set as the nondominated solutions selected from the combination of all Pareto fronts obtained via three algorithms. In other words, P^* for each instance is a result of combining PFs obtained via using DMOEA-εC, NSGA-II, and MOEA/D-AWA and performing non-dominated sorting. The mean and standard deviations of *IGD* and *C-metric* values over 30 independent runs of each algorithm on small-scale, medium-scale and large-scale instances are shown in TABLE II. The Wilcoxon's rank sum test at a 5% significance level is

TABLES II. STATISTICAL RESULTS OF TWO METRICS OF DMOEA-εC, NSGA-II AND MOEA/D-AWA OVER THREE MWTA INSTANCES

		DMOEA-εC	NSGA-II	MOEA/D-AWA
Small-scale	*IGD*	0.3488(0.0058)	**0.0388(0)** §	0.7093(0.0642) †
	$C(P^*,*)$	1(0)	**0.97(0)** §	1(0) †
	$C(*,P^*)$	0.9667(0.0004)	**0.9672(0)** §	08738(0.0047) †
Medium-scale	*IGD*	4.5758(2.7306)	**2.3425(0.5326)** §	10.1201(1.04486) †
	$C(P^*,*)$	1(0)	**0.998(0)** §	1(0) †
	$C(*,P^*)$	0.8734(0.0011)	**0.9343(0.0016)** §	0.7874(0.0015) †
Large-scale	*IGD*	**4.8754(4.3181)**	7.323(9.7586) †	11.6686(10.6451) †
	$C(P^*,*)$	**0.999(0)**	1(0) †	1(0) †
	$C(*,P^*)$	**0.9573(0.0013)**	0.8864(0.0024) †	0.8746(0.0014) †

* represents the approximate *PF*s obtained via each algorithm.

TABLES III. STATISTICAL RESULTS OF THE C-METRIC AMONG DMOEA-εC , NSGA-II AND MOEA/D-AWA OVER THREE MWTA INSTANCES

	Small-scale	Medium-scale	Large-scale
C(DMOEA-εC, NSGA-II)	0.958(0.0006)	0.919(0.003)	0.9714(0.0014)
C (NSGA-II, DMOEA-εC)	1(0)	0.994(0.0002)	0.89(0.0032)
C (DMOEA-εC, MOEA/D-AWA)	0.999(0)	1(0)	0.9671(0.0013)
C (MOEA/D-AWA, DMOEA-εC)	0.9109(0.0027)	0.887(0.0037)	0.9643(0.0045)
C (NSGA-II, MOEA/D-AWA)	1(0)	1(0)	0.9994(0)
C (MOEA/D-AWA, NSGA-II)	0.868(0.00410	0.822(0.0044)	0.905(0.0037)

conducted to test the significance of differences between the mean metric values yielded by DMOEA-εC and comparison algorithms. The numbers in parentheses are the standard deviations. †, §, and ≈ indicate that the performance of the DMOEA-εC is better than, worse than, and similar to that of the comparison algorithm according to the Wilcoxon's rank sum test, respectively. The bold data in the table are the best mean metric values for each instance.

As can be seen in Table II, in terms of *IGD* metric values, NSGA-II shows a significant advantage over DMOEA-εC and MOEA/D-AWA on small-scale and medium-scale MWTA instances. However, DMOEA-εC performs significantly better over NSGA-II and MOEA/D-AWA on the large-scale MWA instance. As to *C-metric*, compared with the approximated true PF P^*, a smaller $C(P^*,*)$ value or a bigger $C(*,P^*)$ value both indicate a better performance of the comparison algorithm. The performance of NSGA-II is superior to DMOEA-εC and MOEA/D-AWA in terms of *C-metric* values on small-scale and medium-scale MWTA instances. However, DMOEA-εC shows significant superiority over NSGA-II and MOEA/D-AWA on the large-scale instance.

To give a further comparison study among DMOEA-εC, NSGA-II, and MOEA/D-AWA on solving the multi-objective MWTA problems, we calculate the *C-metric* of the final PFs obtained by the three algorithms. Table III shows the statistical results of *C-metric* of each two algorithms on three MWTA instances. Paired comparison results in Table III reveals that the overall rank of three algorithms is NSGA-II, DMOEA-εC, and MOEA/D-AWA on small-scale and medium-scale MWTA instances in terms of the *C-metric* value. The overall rank of three algorithms is DMOEA-εC, NSGA-II, and MOEA/D-AWA on the large-scale MWTA instance.

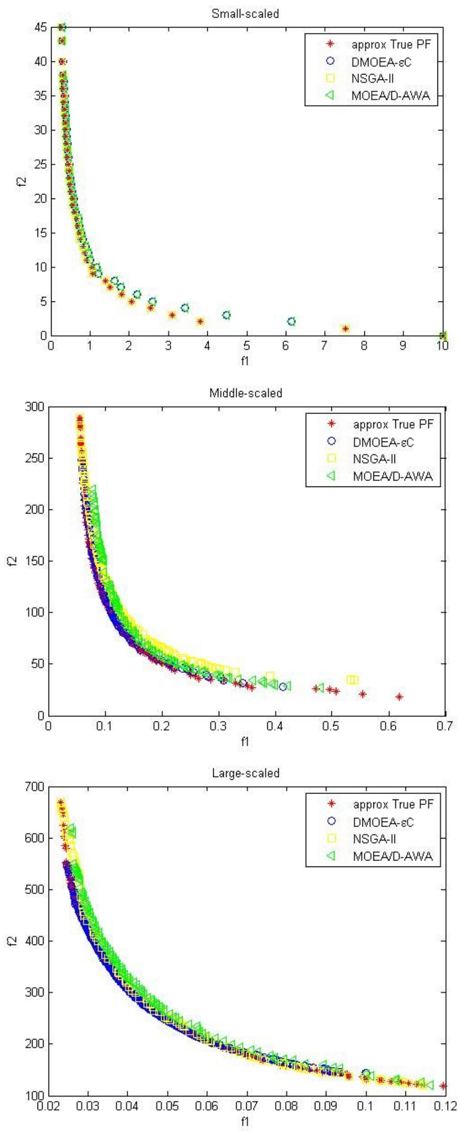

Fig. 1 The final populations in the objective space with the minimum *IGD* metric value within 30 runs obtained by DMOEA-εC, NSGA-II, and MOEA/D-AWA on three MWTA instances.

All these two performance metrics are used to evaluate performances of an algorithm from the angle of quantification, but the intuitive judgment is also essential. For a visual observation, Fig. 1 presents the distribution of the approximated true PF and final PFs in the objective space with the minimum IGD metric value within 30 runs obtained by DMOEA-εC, NSGA-II, and MOEA/D-AWA on small, medium and large-scale MWTA instances. It is visually evident that for the small-scale MWTA instance, the final populations obtained by all three algorithms can cover the whole approximated PF very well. Among them, NSGA-II performs best. While for the medium-scale and large-scale MWTA instances, the final population obtained by NSGA-II cover the whole approximated PF very well. But the final population obtained DMOEA-εC shows better convergence. It

TABLE IV. STATISTICAL RESULTS OF THE TIME PERFORMANCE OF DMOEA-εC, NSGA-II AND MOEA/D-AWA OVER THREE MWTA INSTANCES

		DMOEA-εC	NSGA-II	MOEA/D-AWA
Small-scale	Max	11.36	9.39	12.50
	Mean	10.98(0.56)	6.00(1.23)	12.03(0.22)
	Min	10.36	5.13	11.68
Medium-scale	Max	23.41	14.40	25.90
	Mean	23.10(4.38)	13.71(0.33)	23.74(5.02)
	Min	22.58	13.32	24.68
Large-scale	Max	146.36	85.50	157.10
	Mean	125.31(6.35)	76.91(5.04)	137.35(7.15)
	Min	113.58	70.00	124.3

should be noted that NSGA-II is good at obtaining nondominated solutions located at tails of the PF. DMOEA-εC and MOEA/D-AWA spend more efforts on nondominated solutions located in the middle of the PF, but failed to find solutions located at tails. This can be ascribed to the fact that both DMOEA-εC and MOEA/D-AWA need to estimate the ideal point or the nadir point and update them iteratively. If they failed to obtain good approximations of the two points, they may unable to find solutions located at tails of the PF. However, as to the middle part of the Pareto front, both DMOEA-εC and MOEA/D-AWA can find solutions with better convergence compared with that obtained via NSGA-II.

The following observations can be derived from TABLES II-III and Fig. 1:

- DMOEA-εC performs worse than NSGA-II but better than MOEA/D-AWA on small-scale and medium-scale MWTA instances in terms of both *IGD* and *C-metric* values. Besides, DMOEA-εC demonstrates best performance compared with NSGA-II and MOEA/D-AWA on the large-scale MWTA instance in terms of both *IGD* and *C-metric* values.
- NSGA-II is good at obtaining nondominated solutions located at tails of the Pareto front while DMOEA-εC and MOEA/D-AWA spend more efforts on nondominated solutions located in the middle of the Pareto front. Both DMOEA-εC and MOEA/D-AWA can find solutions with better convergence compared with that obtained via NSGA-II.

TABLE IV shows the statistical results of time performances, in which numbers in parentheses represent standard deviations. It is clear from TABLE IV that, on average, NSGA-II runs about twice as fast as DMOEA-εC and MOEA/D-AWA with the same number of function evaluations for the same scale problem. DMOEA-εC runs slightly faster than MOEA/D-AWA. This observation is because that DMOEA-εC and MOEA/D-AWA needs to normalize objective functions for each run, while NSGA-II does not.

In real combat situations, the time consumption should be limited within seconds or even within milliseconds. Apparently, the processing time on PCs as shown in TABLE III is unqualified. However, this issue can be settled via two aspects. Firstly, one can use lower level languages such as C++ to obtain better time performance. Secondly, since DMOEA-εC, NSGA-II, and MOEA/D-AWA are population-based, techniques for parallel computations on high-performance hardware platform such as the field programmable gate array (FPGA) can be adopted to reduce the processing time in order to satisfy the real-time requirement.

V. CONCLUSION AND FUTURE WORKS

In this paper, the multi-stage weapon target assignment (MWTA) problem which is the basis of DWTA problems in practice is considered and formulated as a multi-objective optimization problem. Apart from the damage effect, the ammunition consumption is also taken into account from a practical standpoint. A newly proposed MOEA, i.e., DMOEA-εC, is firstly applied to solve the MWTA problems. And we compare it with NSGA-II and MOEA/D-AWA on solving three different-scale MWTA instances. Numerical results demonstrate that NSGA-II perform best on small-scale and medium-scale instances, while DMOEA-εC shows advantages over the other two algorithms in solving the large-scale instance in terms of both *IGD* and *C-metric* values. NSGA-II is good at obtaining nondominated solutions located at tails of the Pareto front. While DMOEA-εC and MOEA/D-AWA spend more efforts on nondominated solutions located in the middle of the Pareto front and can obtain solutions with better convergence.

It could be our future work to employ more effective methods for estimating the ideal and nadir points and incorporate problem-specific knowledge to further enhance the performance of DMOEA-εC on solving MWTA problems.

REFERENCES

[1] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, "Exact and heuristic methods for the weapon target assignment problem," *Operations Research*, vol. 55, no. 6, pp. 1136-1146, 2007.

[2] S. P. Lloyd and H. S. Witsenhausen, "Weapons allocation is NP-complete," in *Proc. IEEE Summer Simulation. Conference*, Reno, NV, 1986, pp. 1054–1058.

[3] P. A. Hosein and M. Athans, "Some analytical results for the dynamic weapon-target allocation problem," MIT, Cambridge, U.K., Technical Report, LIDS-P-1944, 1990.

[4] P. A. Hosein and M. Athans, "Preferential defense strategies. Part I: The static case," MIT Lab. Inf. Decis. Syst., Cambridge, MA, Technical Report, LIPS-P-2002, 1990.

[5] P. A. Hosein and M. Athans, "Preferential defense strategies. Part II: The dynamic case," MIT Lab. Inf. Decis. Syst., Cambridge, MA, Technical Report, LIPS-P-2003, 1990.

[6] L. Wu, H. Y. Wang, F. X. Lu, and Peifa Jia, "An anytime algorithm based on modified GA for dynamic weapon-target allocation problem, " in *Proc. IEEE World Congress on Computational Intelligence,* Hong Kong, P. R. China, 2008, pp. 2020-2025.

[7] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[8] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, Evolutionary algorithms for solving multi-objective problems. Norwell, MA: Kluwer, 2002.

[9] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Tech. Rep. 103, Swiss Federal Institute of Technology, 2001..

[12] Z. Zitzler and S. Kunzli, “Indicator-based selection in multiobjective search,” in Parallel Problem Solving from Nature – PPSN VIII:Lecture Notes in Computer Science, vol. 3242, pp. 832–842, 2004.

[13] J. Bader and E. Zitzler, “Hype: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[14] Q. Zhang and H. Li, “A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[16] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances,” Tech. Rep. CES-491, The school of Computer Science and Electronic Engineering, University of Essex, 2009.

[17] A. Zhou and Q. Zhang, “Are all the subproblems equally important? Resource allocation in decomposition based multiobjective evolutionary,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2015.

[18] Y. Qi, X. Ma, and F. Liu, “MOEA/D with adaptive weight adjustment,” *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[19] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[20] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part II: handling constraints and extending to an adaptive approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[21] J Chen, J Li, B. Xin, “DMOEA-εC: Decomposition-bsed multi-objective evolutionary algorithm with the ε-constraint framework,” *IEEE Transactions on Evolutionary Computation*, conditionally accepted.

[22] K. Miettinen, Nonlinear Multiobjective Optimization. Boston: Kluwer Academic Publishers, 1999.

[23] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.

[24] B. Chen, W. Zeng, and Y. Lin, “A new local search-based multiobjective optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 50–73, 2015.

[25] D. P. Mitchell, “Spectrally optimal sampling for distribution ray tracing,” *ACM Siggraph Computer Graphics*, vol. 25, no. 4, pp. 157–164, 1991.

[26] J Li, J Chen, B Xin, “Efficiently solving multi-objective dynamic weapon-target assignment problems by NSGA-II”, in *Proc. 34th Chinese Control Conference (CCC)*, Hanzhou, China, 2015, pp. 2556-2561.

[27] J Li, J Chen, B Xin, L Dou, “Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGAII and adaptive MOEA/D: A comparison study,” in *Proc. IEEE Congress on Evolutionary Computation* (CEC), Seidai, Japan, 2015, pp. 3132–3139.

[28] G. G. Yen, Z. N. He, “Performance metric ensemble for multiobjective evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, Vol. 18, no. 1, pp. 131–144, 2014.