Solving Multi-objective Multi-stage Weapon Target Assignment Problem via Adaptive NSGA-II and Adaptive MOEA/D: A Comparison Study

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

***Abstract*—The weapon target assignment (WTA) problem is a fundamental problem arising in defense-related applications of operations research, and the multi-stage weapon target assignment (MWTA) problem is the basis of dynamic weapon target assignment (DWTA) problems which commonly exist in practice. The MWTA problem considered in this paper is formulated into a multi-objective constrained combinatorial optimization problem with two competing objectives. Apart from maximizing damage to hostile targets, this paper follows the principle of minimizing ammunition consumption under the consideration of resource constraints, feasibility constraints and fire transfer constraints. In order to tackle the two challenges, two types of multi-objective optimizers: NSGA-II (domination-based) and MOEA/D (decomposition-based) enhanced with an adaptive mechanism are adopted to achieve efficient problem solving. Then a comparison study between adaptive NSGA-II (ANSGA-II) and adaptive MOEA/D (AMOEA/D) on solving instances of three scales MWTA problems is done, and four performance metrics are used to evaluate each algorithm. Numerical results show that ANSGA-II outperforms AMOEA/D on solving multi-objective MWTA problems discussed in this paper, and the adaptive mechanism definitely enhances performances of both algorithms.**

Keywords—multi-stage weapon target assignment (MWTA); multi-objective constrained optimization problem; fire transfer constraints; adaptive mechanism; non-dominated sorting genetic algorithm with elitist strategy (NSGA-II); multi-objective evolutionary algorithm based on decomposition (MOEA/D); combinatorial optimization

I. INTRODUCTION

The weapon target assignment (WTA) problem is a fundamental problem arising in defense-related applications of military operations research [1], which deals with 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. The WTA problem is a classical constrained combinatorial optimization problem and has been proved to be NP-complete [2] which means any enumeration-based solver faces exponential computational complexity as the problem size increases. The research on WTA problems dates back to 1950s and 1960s when Manne [3], Braford [4] and Day [5] investigated the WTA modeling issues. Before 1970s, the research had been focused on some special areas, e.g., the missile-based aerial defense [6]. Matlin summarized those works in [7]. The general WTA problem was investigated systematically by Hosein et al. in 1980s [8].

Hosein et al. grouped the WTA problem into two categories: the static WTA (SWTA) problem and the dynamic WTA (DWTA) one [9-11]. 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 the outcomes of this engagement are assessed. Then the strategy for the next stage is decided based on the former assessment. DWTA is a global decision-making process which takes the whole defense effects through all stages into account and incorporates the concept of time window. Time window is a time interval from the time that 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, so each weapon-target-stage pair will be assigned a time window. The term "stage" here is regarded as the period that 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. Multi-stage weapon target assignment (MWTA) falls between SWTA and DWTA, which also takes time windows into account, but doesn't possess the dynamic process that DWTA does. 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 [12]. Following with that, a new decision making process at the next stage will be triggered. This process is the same as the previous, except that the computational complexity is decreased due to the reduction of the numbers of weapons and targets. Thus, there is a cyclic computation: **Decision Making → Damage assessment → Decision Making** in actual DWTA. In a word, the multi-stage version of WTA problems lays the foundation of dynamic ones.

According to different combat situations and missions, WTA problems can be categorized into two types: the target-based WTA problem and the asset-based WTA problem. The aim of the target-based WTA problem is to maximize the

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). Corresponding Author: Bin Xin (brucebin@bit.edu.cn).

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

expectation of the 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 of WTA problems, and the target-based model can be regarded as a special case of the asset-based one.

The objective of traditional WTA problems is focused on operational effects, such as maximizing the expected damage of targets and minimizing the expected loss of protected assets. While in an actual combat situation, a proper WTA scheme should not only satisfy certain operational requirements, but also minimize operational cost. Since the two goals are conflicting in the sense that more ammunition cost, better operational effect, the WTA problem in practice is a multi-objective optimization problem (MOP). Several researchers noticed this fact, used a weighting approach to incorporate several conflicting objectives into a single one and solved it via a single-objective optimizer. However, this approach suffers from the hard choice of proper weights for different objectives, and may produce solutions which are not really expected by commanders. Accordingly, this paper overcomes these disadvantages, models WTA as a two-objective constrained optimization problem and solves it through two prevailing multi-objective optimizers. An important advantage of the two approaches is that they are able to provide an entire front of approximate Pareto optimal solutions. Therefore human decision-makers can have a global view of candidate desirable WTA schemes.

Various algorithms have been proposed for MOPs, including exact and heuristic methods, such as branch and bound (BB) [1], evolutionary algorithms [13], swarm intelligence algorithms [14], differential evolution (DE) [15] and so forth, among which there are three types of frameworks: aggregation-based, domination-based and decomposition-based framework [16]. The aggregation-based framework aggregates multiple objectives into a single one, which may suffer from the problem of choosing proper weights as mentioned before. The domination-based framework optimizes conflicting objectives simultaneously by domination behaviors on solutions. The decomposition-based framework decomposes an MOP into a series of single-objective subproblems and optimizes them concurrently. In this paper, NSGA-II and MOEA/D are adopted and improved with an adaptive mechanism.

The aim of this paper is to modify two state-of-the-art multi-objective optimizers: NSGA-II and MOEA/D by adding an adaptive mechanism, and apply them to solving the proposed multi-objective MWTA problem. The contribution of this paper is in the following aspects:

- Most studies which model WTA problems as MOPs concentrate on SWTA problems instead of the MWTA ones studied in this paper.
- In previous research, the fire transfer constraint which is very important in real combat situations is seldom considered. This paper takes the fire transfer constraint into consideration and formulates it.
- As to the optimizers, both NSGA-II and MOEA/D are very popular algorithms in the field of MOP solving. But previous research mostly benchmarked on a series of continuous instances, some specific combinatorial optimization problems, such as multi-objective traveling salesman problems (MOTSPs) and SWTA problems. In contrast, this paper not only modifies NSGA-II and MOEA/D with an adaptive mechanism, but also adapts it to solve multi-objective MWTA problems.

The rest of this paper is organized as follows. In Section II, a mathematical formulation of the multi-objective MWTA problem is given. Section III provides algorithmic descriptions of NSGA-II, MOEA/D, and the adaptive mechanism. Section IV describes the parameter settings of instances of three scales MWTA problems and experimental results. Section V concludes the paper and gives future works.

II. Mathematical Model

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 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

The MWTA problem is formulated as a two-objective problem from the perspective of target-based models under the assumption that users allocate W weapons among T targets through S stages.

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:

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

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 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 stag 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.

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

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

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

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

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

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

Constraint (3) restricts the maximum number of ammunitions that can be used to destroy each target at each stage, and the value of m_j depends on the performance of available weapons. Constraint (4) signifies that weapon i can fire at most n_i targets at one time, which reflects the capability of weapons to fire at multiple targets simultaneously. Actually, most weapons can fire only one target at the same time, 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 always 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's 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, since DWTA 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. Besides, this update procedure will definitely increase the complexity of DWTA.

3) Fire transfer constraints

Fire transfer constraints widely exist in practice, especially when one weapon is scheduled to hit multiple targets in a given order. Usually, a weapon needs time to prepare for the next engagement (e.g., adjust its orientation and load ammunition) after completing the current attack. Fire transferring is mainly caused by the requirement that a weapon needs to shoot multiple targets at different positions and directions. 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 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 constraints, 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					w_2	$\cdots$	w_W
	(s_1,s_2)	(s_1,s_3)	(s_2,s_3)	$\cdots$	(s_{S-1},s_S)	$\cdots$	$\cdots$	$\cdots$
(t_1,t_2)	1	0	1					
(t_2,t_1)	0	1	0		$\cdots$	$\cdots$	$\cdots$	$\cdots$
(t_1,t_3)	1	0	1					
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
(t_{T-1},t_T)	1	1	0					
(t_T,t_{T-1})	1	1	1		$\cdots$	$\cdots$	$\cdots$	$\cdots$

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

$$x_{w_i t_j}(s_m) x_{w_i t_k}(s_n) \le ftf(w_i, s_m, s_n, t_j, t_k) \tag{7}$$

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, fire transfer constraints impose restrictions on different operations of the same weapon (especially those aimed at different targets).

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

$$P1: \begin{cases} \min \quad f_1 = 1/D_t(\mathrm{X}^t) \\ \min \quad f_2 = C_{ammu} \end{cases}$$
$$s.t. \quad (3),(4),(5),(6),(7)$$

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. Two advanced MOP optimizers: NSGA-II and MOEA/D enhanced with an adaptive mechanism intending to improve performances of original algorithms are adopted to solve the two-objective MWTA problem proposed in this paper.

A. NSGA-II

Kalyanmoy Deb et al. proposed the non-dominated sorting genetic algorithm with elitist strategy (NSGA-II) in 2002, which is a computationally fast and elitist MOEA based on non-dominated sorting approach [17]. 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 calculated by following a given procedure and is used to preserve the diversity of population. With the two attributes, a comparison operator is presented as:

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

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. Above strategies guide the selection process at the various stages of the algorithm toward a uniformly spread-out Pareto optimal front. NSGA-II is a simple yet efficient MOP optimizer with the computational complexity of $O(M \cdot N \cdot \log N)$ (where M is the number of objectives and N is the population size). Readers are suggested to refer to [17] for more details.

B. MOEA/D

The multi-objective evolutionary algorithm based on decomposition (MOEA/D) [18] is a recently developed MOEA in which the decomposition idea is applied instead of the preceding Pareto dominance relation. It decomposes an MOP into a number of scalar optimization problems which are aggregated by using predefined weight vectors, and optimizes them concurrently. There is no need to specify a diversity preservation scheme, since the diversity can be preserved by the predefined uniformly spread weight vectors. Besides, MOEA/D adopts the concept of neighborhood among these aggregated single-objective subproblems，which is defined based on the neighborhood relationships of weight vectors. Each subproblem is optimized by using information mainly from its neighboring subproblems. Specifically, two parents are randomly selected from the current individual's neighborhood and go through reproduction to generate an offspring for the current individual. Then the offspring is compared with the current individual and neighbors using the scalarizing function with its own weight vector. Any solutions that are inferior to the offspring are replaced by the offspring.

There are a number of aggregation approaches for decomposition, such as weighted sum approach, Tchebycheff approach, boundary intersection (BI) approach etc. [18]. This paper employs the Tchebycheff approach with the following form (taking a minimization problem: $\min(f_1(x),...,f_M(x))$ as an example):

$$\min g^{te}(x \mid \lambda, z^*) = \max_{1\le i\le M}\{\lambda_i \mid f_i(x) - z_i^* \mid\}$$

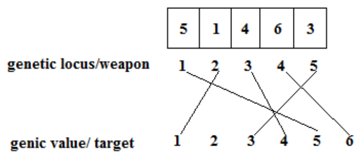
where $\vec{\lambda} = (\lambda_1,...,\lambda_M)$ is the predefined weight vector, $z^* = (z_1^*,...,z_M^*)$ is the reference point, i.e., $z_i^* = \min\{f_i(x)\}, \forall i = 1,...,M$. MOEA/D is an efficient MOP solver with the computational complexity of $O(M \cdot N \cdot niche)$ (where M is the number of objectives, N is the population size and $niche$ is the neighborhood size). Actually, when MOEA/D is adopted to solve practical problems, normalization of objectives is an essential step, which guarantees the effectiveness of algorithms when there is a big difference in objective values. Readers are suggested to refer to [18] for more details.

C. Problem-Specific Operators

1) Chromosome encoding

This paper adopts decimal encoding. The length of a chromosome is the number of weapons. Each weapon is regarded as a genetic locus, and the genic value of each genetic locus indicates the number of the target to which the weapon is assigned. Fig. 1 provides an example to explain the encoding scheme ($W = 5, T = 6, S = 1$), and X is the corresponding 0-1decision matrix.

Such an encoding method can guarantee that every solution satisfies the constraint (4) naturally. As to the other constraints, it will be checked whether each new solution generated by genetic operators satisfies constraints or not. If not, this solution should be repaired according to the random repair mechanism described below.



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

Fig. 1 An example of the encoding scheme

2) Genetic operators based on random repair mechanism

The power of GAs arises from crossover which causes a structured, yet randomized exchange of genetic materials between solutions [19]. A two-point crossover technique with random repair mechanism is adopted in this paper. Since the problem proposed in this paper is constrained, the regular two-point crossover operator may produce unfeasible solutions that violate one or more constraints. To deal with this issue, a random repair mechanism is introduced to repair n_i , m_j , N_i,, respectively, as shown below.

Random repair mechanism

Input: An infeasible solution *x*, the parameter setting of one constraint (including n_i , m_j , N_i).

Output: A feasible solution x'.

Step1. Transform *x* into the corresponding 0-1 matrix *X*;

Step2. Find the indexes of rows (and/or columns) that violate this constraint;

Step3. Calculate the number of redundant 1s in these rows (and/or columns), denote as *num*;

Step4. Randomly replace *num* 1s in these rows (and/or columns) with 0s, obtaining $X^{'}$.

Step5. Retransform the 0-1 matrix $X^{'}$ into $x^{'}$.

Mutation involves the modification of the value of each 'gene' of a solution with some probabilities [19], whose role is to keep diversity of the population and prevent the premature convergence of GA. In this paper we adopt a single-point mutation technique while preserving the feasibility of the solution. Specifically, a genetic locus is randomly selected to be mutated, and the corresponding genetic value is mutated to its possible value according to the feasibility matrix F in constraint (6). This method firstly guarantees the satisfaction of constraint (6) and (4), then randomly mutates to a value that satisfies the other three constraints.

D. Adaptive Mechanism

When using GA as search engine, crossover and mutation probabilities have an important effect on the performance of algorithms. Since Srinivas et al. [19] first proposed the adaptive GA (AGA), various kinds of adaptive mechanisms have been put forward, such as non-linear and fuzzy mechanisms, which were all proved to perform better than the standard GA (SGA). Most previous research holds the view that a good solution should not be encouraged to crossover to avoid breaking the structure of solutions. However, this adaptive mechanism may lead to premature convergence to local optima Thus, we proposed an opposite adaptive mechanism of crossover rate, which allots high crossover probability to good solutions with the idea that 'good' solutions can generate 'better' ones. The goal of this adaptive mechanism is to pass good structures of good solutions to the next generation.

Specifically, taking the minimization problem: $\min f$ as an example to illustrate the mechanism of varying crossover p_c and mutation probabilities p_m adaptively:

$$p_c = \begin{cases} k_1 \dfrac{\overline{f} - f_{\min}}{f^{'} - f_{\min}} & f^{'} > \overline{f} \\ k_3 & f^{'} \le \overline{f} \end{cases}, \quad p_m = \begin{cases} k_2 \dfrac{f - f_{\min}}{\overline{f} - f_{\min}} & f \le \overline{f} \\ k_4 & f > \overline{f} \end{cases}$$

where $\overline{f}$ and $f_{\min}$ are the average and minimal fitness value of the population, respectively; $f^{'}$ represents the smaller fitness values of the solutions to be crossed; f stands for the fitness value of the solution to be mutated; k_1 and k_2 have to be less than 1 to constrain p_c and p_m to the range [0, 1]; We set $k_1 = k_3 = 1; k_2 = k_4 = 0.5$.

Replace fixed crossover and mutation probabilities in the original NSGA-II and MOEA/D with the proposed adaptive mechanism, we will get the adaptive NSGA-II and adaptive MOEA/D (denoted as ANSGA-II and AMOEA/D for convenience), respectively.

IV. NUMERICAL EXPERIMENTS

A. Parameter Settings

Since the actual data is difficult to obtain, in this section, three types of data for different combat situations, i.e. small, medium and large-scaled cases are randomly generated. For each situation, the weapon kill probability matrix $P(s)$, ammunition consumption matrix $U(s)$, feasibility matrix $F(s)$ and the fire transfer feasibility matrix $FTF(s)$ are randomly generated within a given range related to the problem scale. The number of weapons/targets/stages in small, medium and large-scaled instances are 3/5/3, 20/12/5 and 50/50/8, respectively. The value vector of targets comes from Ref [20] and other parameters are given in TABLE I.

TABLE I. PARAMETER SETTINGS OF (A) NSGA-II AND (A)MOEA/D FOR INSTANCES OF THREE SCALES ON MWTA PROBLEMS

Instance	n_i	m_j	N_i	*pop*	*gen*	*niche*
Small	1	1	2	100	100	5
Medium	1	2	2	100	150	5
Large	1	2	2	100	200	5

In all instances: k_1=k_3=1; k_2=k_4=0.5.

B. Performance metrics

No single one performance metric can provide a comprehensive measure on the performance of an MOEA [21]. In the following experimental studies, the following three widely used performance metrics are used.

1) Ratio of non-dominated individuals (RNI): $RNI = \dfrac{|\overline{P}|}{|P|}$

$\overline{P}$ denotes the set of non-dominated individuals in population P, and $|\cdot|$ equals the number of points in a set. Clearly $RNI \in [0,1]$ checks the proportion of non-dominated individuals in population P, and is the larger the better.

2) 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 can not miss any part of the whole PF.

3) 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 metrics and is defined as the percentage of solutions in P_2 that are dominated by at least one solution in P_1.

C. Comparison Results

ANSGA-II and AMOEA/D are tested on instances of three different scales MWTA problems (small, medium and large-scaled ones), intending to show the effectiveness of the adaptive mechanism and give a comparison study. Both ANSGA-II and AMOEA/D have been independently run for 20 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 said before, *RNI*, *IGD*, *C-metric* and *time performance metric* are used to compare results.

In what follows, the small-scaled instance is firstly taken as an example to illustrate the performance of NSGA-II, ANSGA-II, MOEA/D and AMOEA/D. Then the following introduces medium-scaled and large-scaled MWTA instances to further demonstrate performances of both algorithms.

1) Computation results on small-scaled instance

Firstly, an enumeration method is employed to get a set of evenly distributed true optimal solutions, and thus obtain an evenly distributed true Pareto front (PF) for the small-scaled instance. More specifically, the original two-objective constrained combinatorial optimization problem is converted to a single-objective constrained one by transferring the second objective into another constraint using an inequality with a lower bound. To get an evenly distributed PF, the upper bound of the second objective function $C_{\max}$ is computed firstly. Then divide $[0, C_{\max}]$ into several equal intervals $[0,d_1],[d_1,d_2],...,[d_n,C_{\max}]$ and apply the enumeration method to solve the single-objective constrained optimization problem *P2* described as below to obtain the evenly distributed true PF.

$$P2: \max f_1 = D_t(X^t) = \sum_{j=1}^{T(t)} v_j (1 - \prod_{s=t}^{S} \prod_{i=1}^{W(t)} (1-p_{ij}(s))^{x_{ij}(s)})$$
$$s.t. \sum_{s=1}^{S} \sum_{j=1}^{T} \sum_{i=1}^{W} \beta_i u_{ij}(s) x_{ij}(x) \le d_k, k = 1, ...n+1$$
$$(3),(4),(5),(6),(7)$$

Next, NSGA-II, ANSGA-II, MOEA/D and AMOEA/D are applied to solve the multi-objective MWTA problem *P1* to get approximate PFs for further comparison.

TABLES II gives the statistical results of three metrics among NSGA-II, ANSGA-II, MOEA/D and AMOEA/D on small-scaled instances. The PF^*_{true} in $C(PF^*_{true},*)$ denotes the true PF obtained via the enumeration method. Numbers in parentheses represent standard deviations.

TABLES II. STATISTICAL RESULTS OF THREE METRICS AMONG NSGA-II, ANSGA-II, MOEA/D AND AMOEA/D ON SMALL-SCALED INSTANCES

		ANSGA-II	NSGA-II	AMOEA/D	MOEA/D
RNI	Max	**1**	1	1	1
	Mean	**1(0)**	1(0)	0.95(0.052)	0.95(0.053)
	Min	**1**	1	0.86	0.86
IGD	Max	**7.76E-05**	2.90	23.30	23.30
	Mean	**3.9E-6(1.7E-5)**	0.54(0.96)	18.8(2.83)	18.9(2.84)
	Min	**0**	0	11.875	11.875
$C(PF^*_{true},*)$	Max	**0.857**	1	1	1
	Mean	**0.857 (0)**	1(0)	1(0)	1(0)
	Min	**0.857**	1	1	1

* represents the approximate PFs obtained via each algorithm.

2) Computation results for medium and large-scaled instances

For medium and large-scaled instances, it is unacceptable to use the enumeration method to obtain true PFs. Thus, 80 PFs (20 PFs each algorithm) for each instance are combined together and performed non-dominated sorting to get an approximate true PF, denoted as PF^*_{approx}, for performance metrics calculations. Specifically, PF^*_{approx} for each instance is a result of combining PFs obtained via NSGA-II, ANSGA-II, MOEA/D and AMOEA/D and performing non-dominated sorting. TABLES III and TABLES IV present statistical results of three metrics among the four algorithms on medium and large-scaled instances, respectively. Similarly, numbers in parentheses represent standard deviations of each performance metric.

TABLES III. STATISTICAL RESULTS OF THREE METRICS AMONG NSGA-II, ANSGA-II, MOEA/D AND AMOEA/D ON MEDIUM-SCALED INSTANCES

		ANSGA-II	NSGA-II	AMOEA/D	MOEA/D
RNI	Max	**1**	1	0.9	0.84
	Mean	**1(0)**	1(0)	0.72(0.16)	0.65(0.13)
	Min	**1**	1	0.37	0.35
IGD	Max	**13.2**	21.5	50.3	58.12
	Mean	**4.11 (2.85)**	5.88(6.71)	35.1(8.85)	45.3 (6.72)
	Min	**1.24**	0.784	22.4	35.3
$C(PF^*_{approx},*)$	Max	0.98	0.98	1	1
	Mean	0.98(0)	0.97(0.007)	1(0)	1(0)
	Min	0.98	0.96	1	1

* represents the approximate PF obtained via each algorithm.

TABLES IV. STATISTICAL RESULTS OF THREE METRICS AMONG NSGA-II, ANSGA-II, MOEA/D AND AMOEA/D ON LARGE-SCALED INSTANCES

		ANSGA-II	NSGA-II	AMOEA/D	MOEA/D
RNI	Max	**1**	1	0.96	0.95
	Mean	**1(0)**	1(0)	0.80(0.155)	0.73(0.12)
	Min	**1**	1	0.43	0.51
IGD	Max	**17.10**	24.6911	119.3	122.7
	Mean	**4.20(4.92)**	11.80(6.36)	100.5(9.50)	102.5(16.3)
	Min	**1.30**	1.4903	81.8	73.8
$C(PF^*_{approx},*)$	Max	0.98	0.98	1	1
	Mean	0.97(0.003)	0.97(0.005)	1(0)	1(0)
	Min	0.97	0.96	1	1

* represents the approximate PF obtained via each algorithm.

To give a further comparison study between ANSGA-II and AMOEA/D on solving multi-objective MWTA problems, we calculate the *C-metric* of the final PF obtained by the two algorithms with the same adaptive mechanisms and repair techniques. TABLES V shows the statistical results of *C-metric* on instances of three scales. Denote by PF_1 and PF_2 the PF obtained by ANSGA-II and that obtained by AMOEA/D, respectively.

TABLES V. STATISTICAL RESULTS OF C-METRIC BETWEEN ANSGA-II (PF_1) AND AMOEA/D (PF_2) ON INSTANCES OF THREE SCALES

C-metric		Small Scale	Medium Scale	Large Scale
$C(PF_1,PF_2)$	Max	1	1	1
	Mean	1(0)	0.75(0.44)	1(0)
	Min	1	0.5	1
$C(PF_2,PF_1)$	Max	**0.81**	**1**	**0.98**
	Mean	**0.347 (0.192)**	**0.4 (0.502)**	**0.280 (0.337)**
	Min	**0.03**	**0**	**0**

The following observations can be derived from TABLES II-V:

- Both ANSGA-II and AMOEA/D outperform the original algorithms for the reason of incorporating the adaptive mechanism.
- For each instance, ANSGA-II can find solutions that all locate in the first non-dominated front (i.e., *RNI* values all equal to 1), while solutions found by AMOEA/D come from two or more non-dominated fronts. This is because solutions in ANSGA-II are sorted based on the non-dominated behaviors which make final solutions locate in the first non-dominated front, In contrast, AMOEA/D is a decomposition-based method which doesn't possess the mechanism to maintain non-dominated solutions during evolution.

- The standard deviations of each performance metric show that ANSGA-II performs more stable than AMOEA/D.
- As to the comparison results illustrated in TABLES V, ANSGA-II outperforms AMOEA/D in solving the two-objective constrained optimization problem, which is consistent with former analysis.

All these three performance metrics are used to evaluate performances of an algorithm from the angle of quantification, the intuitive judgment is still essential. Fig. 2 and Fig. 3 present the distribution of the true or approximate true PF and final PFs obtained in 20 runs with the best performance metrics of each algorithm for each test instance in the objective space. It is evident that ANSGA-II is better than AMOEA/D as to the quality of solutions. Specifically, ANSGA-II can find closer and more evenly distributed PF in the objective space than AMOEA/D. Fig. 4, Fig. 5 and Fig. 6 show the evolution of best objective values (f_1 and f_2) in ANSGA-II and AMOEA/D on instances of three scales, which also indicate similar results.

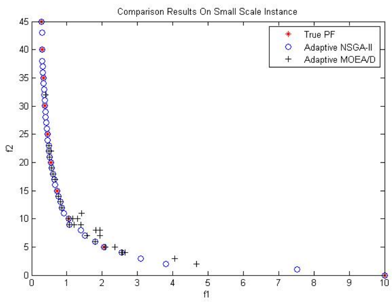


Fig. 2 Plots of the true PF and approximate PFs with best performance metrics in 20 runs of ANSGA-II and AMOEA/D on small-scaled instance.

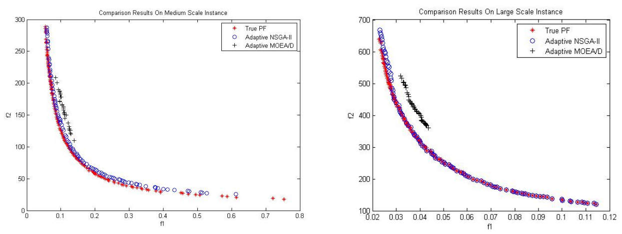


Fig. 3 Plots of the approximate true PF and approximate PFs with best performance metrics in 20 runs of ANSGA-II and AMOEA/D on medium and large-scaled instances.

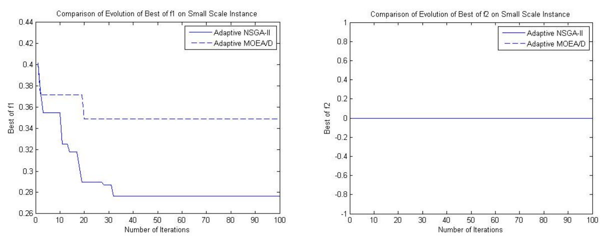


Fig. 4 The evolution of best objective values (f_1 and f_2) in ANSGA-II and AMOEA/D on small-scaled instance.
Remark: The solid line and dash line in Fig. 4 are overlapped

3) Time performance on instances of three scales

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

TABLE VI. STATISTICAL RESULTS OF TIME PERFORMANCES ON INSTANCES OF THREE SCALES (IN SECS.)

Time Performance	Small Scale		Medium Scale		Large Scale	
	ANSGA-II	AMOEA/D	ANSGA-II	AMOEA/D	ANSGA-II	AMOEA/D
Max	9.39	12.5	14.4	25.9	85.5	157
Mean (std.)	6.00 (1.23)	12.03 (0.223)	13.7 (0.334)	23.7 (5.02)	76.9 (5.04)	137 (7.15)
Min	5.125	11.68	13.3	24.6	70	124

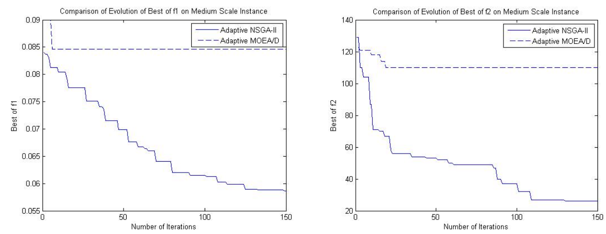


Fig. 5 The evolution of best objective values (f_1 and f_2) in ANSGA-II and AMOEA/D on medium-scaled instance.

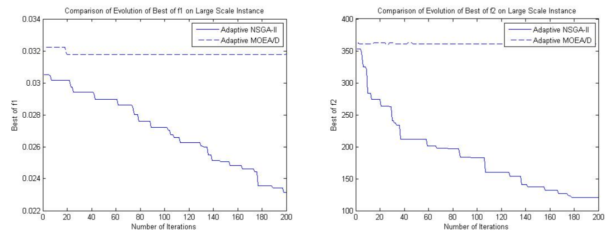


Fig. 6 The evolution of best objective values (f_1 and f_2) in ANSGA-II and AMOEA/D on large-scaled instance.

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 VI 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 NSGA-II and MOEA/D both 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.

Remark:

That is not to say MOEA/D has no advantages on solving MOPs. Some research has shown that NSGA-II suffers from the problem of "selection pressure" when dealing with many-objective problems ($M \geq 4$). While MOEA/D is more appropriate for solving problems with many objectives. Besides, a simple yet efficient way of normalizing objectives and techniques that can maintain the diversity of solutions will be helpful in applying MOEA/D to various MOPs. As to the multi-objective MWTA problem discussed in this paper, firstly, the strictness of the five constraints causes a complex feasible region. Thus, the exchange of information among neighborhoods in MOEA/D does not work effectively when dealing with this problem. While the domination behaviors in NSGA-II are less affected by the complexity of the feasible region. Secondly, the normalization of objectives is an essential step in MOEA/D when dealing with the multi-objective MWTA problem, which definitely increases the computational complexity. While NSGA-II does not need this procedure.

V. Conclusion and Future Works

In this paper, the multi-stage weapon target assignment (MWTA) problem which is the basis of DWTA problems is considered and formulated into a multi-objective optimization problem. Apart from the damage effect, the ammunition consumption is taken into account from a practical standpoint. Apart from traditional resource constraints, the fire transfer constraints are introduced, which also makes the model more practical. Then an adaptive mechanism is proposed and embedded into two types of multi-objective optimizers: domination-based and decomposition-based ones to get the adaptive NSGA-II (ANSGA-II) and adaptive MOEA/D (AMOEA/D). To give a comprehensive comparison study between ANSGA-II and AMOEA/D on solving multi-objective MWTA problems, three performance metrics of the solution quality are used to evaluate each algorithm on instances of three different scales. Besides, plots of the (approximate) true Pareto front and approximate PFs with best performance metrics in 20 runs of ANSGA-II and AMOEA/D are presented to have a visual impression. At last, we give the time performance of two algorithms on each test instance and two possible methods to settle the issue of calculation time.

To summarize, ANSGA-II obviously outperforms AMOEA/D. ANSGA-II can give either optimal or almost optimal solutions for all multi-objective MWTA instances. In contrast, AMOEA/D can only find solutions that are inferior to these found by ANSGA-II with regard to convergence and diversity. Specifically, solutions obtained by AMOEA/D not only are dominated by most solutions in the true or approximate true PF, but also spread not as uniformly as solutions obtained by ANSGA-II for each multi-objective MWTA instance. In addition, the adaptive mechanism definitely enhances performances of both algorithms.

In the paper, we only test the performance of the proposed adaptive mechanism on series of the MWTA problems. It could be our future work to check the behavior of the proposed adaptive mechanism on standard benchmark problems. Besides, an improved version of NSGA-II (called NSGA-III [22]) was proposed recently for efficiently solving many-objective problems, which can be taken into consideration in solving WTA problems in the future research.

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] A. S. Manne, "A target-assignment problem," *Operations Research*, vol. 6, no. 3, pp. 346-351, 1958.

[4] J. C. Braford, "Determination of optimal assignment of a weapon system to several targets, " Vought Aeronautics, Dallas, Texas, Technical Report, AEREITM-9, 1961.

[5] R. H. Day, "Allocating weapons to target complexes by means of nonlinear programming," *Operations Research*, vol. 14, no. 6, pp. 992-1013, 1966.

[6] H. Cai, J. Liu, Y. Chen, H. Wang, "Survey of the research on dynamic weapon-target assignment problem," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 559 – 565, 2006.

[7] S. Matlin, "A review of the literature on the missile allocation problem," *Operations Research*, vol. 18, no. 2, pp. 334-373, 1970.

[8] P. A. Hosein, J. T. Walton, M. Athans, "Dynamic weapon-target assignment problems with vulnerable C2 nodes," MIT Lab. Inf. Decis. Syst., Cambridge, Technical Report, LIDS-P-1786, 1988.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] AR Yildiz, "Comparison of evolutionary based optimization algorithms for structural design optimization," *Engineering Applications of Artifical Intelligence,* vol. 26, no. 1, pp. 327-333, 2013.

[14] AR Yildiz, KN Solanki, "Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach," *International Journal of Advanced Manufacturing Technology,* vol. 59, no. 1-4, pp. 367-376, 2012.

[15] AR Yildiz, "Comparison of evolutionary based optimization algorithms for structural design optimization," *Engineering Applications of Artifical Intelligence,* vol. 26, no. 1, pp. 327-333, 2013.

[16] V. A. Shim, K. C. Tan and K. K. Tan, "A hybrid adaptive evolutionary algorithm in the domination-based and decomposition-based frameworks of multi-objective optimization," in *Proc. IEEE World Congress on Computational Intelligence,* Brisbane, QLD, 2012, pp. 1-8.

[17] K. Deb, A. Pratap, S. Agarwal 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.

[18] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transaction on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[19] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 6, pp. 656-667, 1994.

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

[21] 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.

[22] K. Deb, 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.