Efficient Decision Makings for Dynamic Weapon-Target Assignment by Virtual Permutation and Tabu Search Heuristics

Bin Xin, *Member, IEEE*, Jie Chen, *Member, IEEE*, Juan Zhang, Lihua Dou, and Zhihong Peng

***Abstract*—The dynamic weapon-target assignment (DWTA) problem is a typical constrained combinatorial optimization problem with the objective of maximizing the total value of surviving assets threatened by hostile targets through all defense stages. A generic asset-based DWTA model is established, especially for the warfare scenario of force coordination, to formulate this problem. Four categories of constraints, involving capability constraints, strategy constraints, resource constraints (i.e., ammunition constraints), and engagement feasibility constraints, are taken into account in the DWTA model. The concept of virtual permutation (VP) is proposed to facilitate the generation of feasible decisions. A construction procedure (CP) converts VPs into feasible DWTA decisions. With constraint satisfaction guaranteed by the synergy of VPs and the CP, an elaborate local search (LS) operator, namely move-to-head operator, is constructed to avoid repeatedly generating the same decisions. The operator is integrated into two tabu search (TS) algorithms to solve DWTA problems. Comparative experiments involving a random sampling method, an LS method, a hybrid genetic algorithm, a hybrid ant-colony optimization algorithm, and our TS algorithms show that the proposed TS heuristics for DWTA outperform their competitors in most test cases and they are competent for high-quality real-time DWTA decision makings.**

***Index Terms*—Combinatorial optimization, constraint handling, Dynamic weapon-target assignment (DWTA), metaheuristics, military decision making, tabu search (TS), virtual permutation (VP).**

I. Introduction

THE weapon-target assignment (WTA) problem is a classical constrained combinatorial optimization problem arising in the field of military operations research. Its objective is to achieve certain tactical goals by assigning available weapons to hostile targets at appropriate occasions. The automatic resolution of this problem becomes indispensable due to the demand on fast decision makings in modern warfare and the advent of more hi-tech weapon platforms, like the unmanned air vehicles (UAVs). In fact, the WTA problem entailed by single-platform combat and also force coordination (see the surveys [1] and [2]) is one of the crucial issues for the automation of military command and control (C2) [3]. Besides, the WTA problem solving can also provide some inspirations for solving other decision-making problems (e.g., media allocation [4]).

WTA has been proved to be NP-complete [5]. It has two versions—static WTA (SWTA) and dynamic WTA (DWTA) [6], [7]. In SWTA, all weapons engage targets in a single stage, and all of the parameters for the problem are known. Thus, the goal of SWTA is to find the optimal assignment for a temporary defense task. In contrast, DWTA is a multistage problem and the outcome of each engagement is assessed for subsequent decision makings. The goal of DWTA is to find a global optimal assignment for the whole defense process in which the engagement occasion of weapons must be taken into account. Intuitively, DWTA can be achieved by a series of SWTAs through all stages. However, although SWTA can at best guarantee the optimality of the WTA decisions for its corresponding defense stage, the combination of all SWTA decisions may not be optimal for the whole defense process. Besides, the actual issue of time windows, which limit the engagement of weapons, is not involved in SWTA [6]. In addition to the engagement constraints, the complexity of DWTA problems is also caused by resource constraints and strategy constraints [1], [8]. The rest of this section includes a review of previous researches on WTA and an outline of this paper.

Previous researches on WTA mainly focus on SWTA [9]–[22]. In respect of SWTA models, Hosein and Athans [6] proposed an asset-based SWTA model that was also used in [9] and [10]. In the research of Karasakal, the probability to down all incoming targets is adopted as the objective function of the air defense WTA model for a naval task group [11]. Some scholars adopted target-based SWTA models that do not employ the value of threatened assets directly [12]–[21]. Instead, each target in this case is assumed to have certain value of threat and the objective is to minimize the total threat of all targets. Typically, threats are assessed according to two criteria: *capability* and *intent* [2], and the intent aspect mainly refers to the value of threatened assets. Therefore, the asset-based WTA model can interpret in a straightforward manner the inherent relationship between WTA decisions and the ultimate objective of protecting own-force assets. In fact, the probability model employed by Karasakal [11] and the target-based models are just special cases of the asset-based model proposed by Hosein and Athans [6]. Besides, the cost of weapons is also taken into account in some models, like that in the research of Kwon *et al.* [22]. A more

Manuscript received August 6, 2009; revised November 17, 2009, January 29, 2010, and April 8, 2010; accepted April 21, 2010. Date of publication June 1, 2010; date of current version October 15, 2010 This work was supported by the National Science Fund for Distinguished Young Scholars under Grant 60925011 and by the Key Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences under Open Issues Foundation Grant 20060104. This paper was recommended by Associate Editor S. H. Rubin.

The authors are with the School of Automation, Beijing Institute of Technology, 100081 Beijing, China and also with the Key Laboratory of Complex System Intelligent Control and Decision, Ministry of Education, 100081 Beijing, China (e-mail: brucebin@bit.edu.cn; chenjie@bit.edu.cn; zhjuan@bit.edu.cn; doulihua@bit.edu.cn; peng@bit.edu.cn).

Digital Object Identifier 10.1109/TSMCC.2010.2049261

complicated model that considers the function of special assets can be found in [23].

Based on the aforementioned models, varied algorithms have been proposed to solve SWTA since the middle of the last century. In early stages, SWTA algorithms were limited to traditional algorithms, such as implicit enumeration algorithms and branch-and-bound algorithms [1]. With the development of computer technology, some novel algorithms, such as neural networks [12], genetic algorithms (GAs) [9], [13], [14], tabu search (TS) [15], simulated annealing algorithms (SAs) [16], ant-colony optimization (ACO) [17], and particle swarm optimization (PSO) [18], have been developed. Some scholars also tried hybrid algorithms [10], [19], [20]. For example, Lee *et al.* designed a memetic algorithm that combines the advantages of global search (GA) and local search (LS) (greedy eugenics) to solve target-based SWTA problems [19]. Besides, Ahuja *et al.* successfully developed several branch-and-bound algorithms and a very large-scale neighborhood search algorithm to solve target-based SWTA problems [21]. Blodgett *et al.* categorized WTA problem-solvers into reactive planning and deliberative planning from the perspective of resource management [24]. The reactive planning uses low-level reasoning techniques for a simple response to a situation to give a short reaction time. Due to the virtue of real-time decision making, the reactive planning has been applied in many military systems, such as the rapid anti-ship missile-integrated defense system (RAIDS) and the reactive resource allocation at single ship level (RRASSL) system [24]. In contrast, the deliberative planning, employing sophisticated high-level reasoning techniques, involves heuristic and metaheuristic methods and often takes into account the factor of time horizon.

In contrast to SWTA, DWTA attracted more attention of researchers in recent years, though it was proposed by Hosein and Athans in 1990 [1], [7], [8], [23]–[31]. Cai *et al.* introduced some basic concepts on DWTA and provided a systematic survey on WTA problems [1]. Hosein and Athans did an early research on a two-stage asset-based DWTA problem and proposed a suboptimal algorithm for finding a good solution [7]. Hosein *et al.* also made some typical empirical experiments and provided analytical solutions to several simple cases of DWTA [23], [25]. Khosla used a hybrid GA (HGA), which incorporates an SA-type heuristic to solve a target-based DWTA problem [26]. In particular, a weighted combination of threat value and option weight is employed in the objective function of this DWTA model. Havens models DWTA by means of simulation [27]. However, it is in fact the repetition of SWTA [8]. In the research of Zacherl [28], a GA is employed to address the DWTA problem regarding UAV in the mission of destroying some time-sensitive targets, like the leader of terrorists. Note that the WTA problem in [28] is formulated as a constrained linear 0–1 programming model, which conceals the inherent relationship between WTA decisions and the objective of maximizing the damage to enemies. Wu *et al.* also used a GA to solve DWTA problems, and the effect of time windows is integrated into the generation of decisions [29]. However, the DWTA model in [29] is also a repetition of SWTA models. Sikanen employed dynamic programming to solve DWTA problems with the assumption that all engaged targets will always be destroyed [30]. Although, Sikanen tried to reduce the computation burden of the dynamic programming, its computation complexity is still exponential. Dionne *et al.* proposed a sequential DWTA algorithm for naval warfare [31]. This algorithm considers all potential decisions and also suffers from the curse of dimension explosion. Li *et al.* proposed a target-based DWTA model with the objective of minimizing the total threat of the targets that survive the final stage of air defense operation [8].

The goal of this paper is to develop an efficient algorithm to solve *asset-based* DWTA problems, involving *capability constraints*, *strategy constraints*, *resource constraints*, and *engagement feasibility constraints*. The contribution of this paper mainly includes the following three aspects.

1) A generic defense-oriented DWTA model is established, which takes into account different kinds of practical constraints.
2) A technique of combining virtual permutation (VP) based solution representation and construction procedure (CP) is proposed to achieve the satisfaction of all constraints. It is a new technique for constraint handling.
3) TS algorithms based on the constraint-handling technique mentioned earlier are designed to efficiently solve DWTA problems.

The rest of this paper is organized as follows. In Section II, the mathematical model for asset-based DWTA problems is formulated and some novel concepts about DWTA are presented. In Section III, a general "*virtual*" representation of solutions (decisions), termed as VP, is proposed to facilitate the generation of feasible solutions, and a CP transforms VPs into feasible solutions. Then, an LS operator, namely move-to-head (MtH) operator, is proposed to generate new VPs. The operator is integrated into TS algorithms to solve DWTA problems. In Section IV, the proposed DWTA algorithms are empirically tested in two simple, but typical cases at first. Then, a DWTA test-case generator is designed to compare the performance of different optimizers in solving DWTA instances of different scales. Finally, Section V concludes the paper.

II. DWTA Formulation: An Asset-Based Model

DWTA models depend on many factors, such as defense strategies, features of targets and weapons, and actual combat situations. Different defense scenarios may require different models. The scenario considered in this paper is narrated as follows. At certain time, the defender detects T offensive targets with their attack aims exposed, and K assets of the defender are threatened. There are W weapons available to intercept the targets. Before these targets break through the defense, there are, at most, S stages in which the defender's weapons can be assigned to certain targets. The value of S depends on the distance between targets and their aims, target's flight speed, weapon's regulation and launch and flight time, the delay of data analysis, decision making, etc [11]. A general engagement policy "shoot-look-shoot (SLS)" is assumed, which is a tradeoff between defense effect and defense cost [7], [11], [24]–[26]. This scenario is very common within the context of naval task groups (e.g., carrier battle group) and ground-based air defense systems (GBADS), which demand force coordination [2].

The expected total value of assets surviving through the whole defense process is adopted as the objective function for DWTA decision makings as follows:

$$J_t(\mathbf{X}^t) = \sum_{k=1}^{K(t)} v_k \prod_{j \in \mathsf{T}_k(t)} \left[1 - q_{jk} \prod_{h=t}^{S} \prod_{i \in \mathsf{W}_j^h} (1 - p_{ij}(h))^{x_{ij}(h)} \right]$$

$$\text{with} \quad t \in \{1, 2, \ldots, S\} \tag{1}$$

where t and h are the indexes of defense stage, $\mathbf{X}^t = [X_t, X_{t+1}, \ldots, X_S]$ with $X_t = [x_{ij}(t)]_{W \times T}$ is the decision variable in stage t, and $x_{ij}(t) = 1$ if weapon i is assigned to target j in stage t, $x_{ij}(t) = 0$ otherwise, $K(t)$ is the total number of remaining assets, and obviously, $K(1) = K$, $\mathsf{T}_k(t)$ is the index set of the targets that threaten asset k in stage t, W_j^h is the index set of the weapons that are assigned in stage h to intercept target j, v_k is the value of asset k, q_{jk} is the lethality probability that target j destroys asset k, and $p_{ij}(t)$ is the lethality probability that weapon i destroys target j in stage t.

Definition 1 (Global decision): The decision corresponding to the whole defense process from current stage to final stage is termed as a global decision. In the aforementioned expression, $\mathbf{X}^t = [X_t, X_{t+1}, \ldots, X_S]$ denotes the global decision in stage t.

Definition 2 (Executive decision): The first component of a global decision is termed as an executive decision. $X_t = [x_{ij}(t)]_{W \times T}$ denotes the executive decision in stage t.

Remark 1: In general, the DWTA procedure can be described by the following two steps.

Step 1: Global decision making. Choose a globally optimal or satisfactory decision scheme from the perspective of all stages.

Step 2: Implementation. Carry out the executive decision w.r.t. the global decision made in Step 1.

After an executive decision is implemented, its defense effect will be evaluated by observation of target states, which indicates the end of a defense stage. Then, the two aforementioned DWTA steps will proceed until all targets are destroyed or the ammunition for weapons is used out.

The constraints involved in WTA primarily include capability constraints, strategy constraints, resource constraints, and engagement feasibility constraints

$$\sum_{j=1}^{T} x_{ij}(t) \le n_i \quad \forall t \in \{1, 2, \ldots, S\} \quad \forall i \in \{1, 2, \ldots, W\} \tag{2}$$

$$\sum_{i=1}^{W} x_{ij}(t) \le m_j \quad \forall t \in \{1, 2, \ldots, S\} \quad \forall j \in \{1, 2, \ldots, T\} \tag{3}$$

$$\sum_{t=1}^{S} \sum_{j=1}^{T} x_{ij}(t) \le N_i \qquad \forall i \in \{1, 2, \ldots, W\} \tag{4}$$

$$x_{ij}(t) \le f_{ij}(t) \quad \forall t \in \{1, 2, \ldots, S\} \quad \forall i \in \{1, 2, \ldots, W\} \quad \forall j \in \{1, 2, \ldots, T\}. \tag{5}$$

The constraint set (2) reflects the capability of weapons in firing at multiple targets at the same time. Most of actual weapons can shoot only one target at a time. Besides, a special weapon that can simultaneously engage multiple targets can be viewed as multiple separate weapons. In view of these facts, we set $n_i = 1$ for $\forall i \in \{1, 2, \ldots, W\}$. The constraint set (3) limits the weapon cost for each target in each stage. The setting of m_j $(j = 1, 2, \ldots, T)$ usually depends on the combat performance of available weapons. In our research, we suppose that $m_j = 1$ for $\forall j \in \{1, 2, \ldots, T\}$. This is a reasonable setting for missile-based defense systems and the "SLS" engagement policy [7], [11]. For artillery-based defense systems, the value of m_j $(j = 1, 2, \ldots, T)$ may be greatly increased under the same demand on defense strength. Therefore, the constraints in (3) can be considered as strategy constraints. The constraint set (4) reflects, in essence, the amount of ammunition equipped for weapons. $N_i (i = 1, 2, \ldots, W)$ is the maximal number of times that weapon i can be used due to the limit of its equipped ammunition. In the constraint set (5), $f_{ij}(t)$ is the indication of actual engagement feasibility for weapon i assigned to target j in stage $t. f_{ij}(t) = 0$ if weapon i cannot shoot target j in stage t with any potential reason; $f_{ij}(t) = 1$ otherwise. The time window of targets and weapons is the primary factor that affects engagement feasibility [1], [8]. Some scholars also use the term "cue" or "deadline" when referring to the issue of time windows [24], [29]. The constraint set (5) is very important to actual DWTA problems, since it contains the influence of time windows on the engagement feasibility of weapons. Besides, it also increases the complexity of DWTA problems and the difficulty of generating feasible solutions. In this case, it is hard to design a desirable operator that can generate new solutions and guarantee their feasibility at the same time.

To sum up, the optimization model for the DWTA problem can be formulated as follows:

$$\text{maximize } J_t(\mathbf{X}^t), \quad \text{s.t.}(2), (3), (4), (5), \quad t = 1, 2, \ldots, S. \tag{6}$$

There are at most S stages (chances) for the defender to make DWTA decisions if the prediction, embedded in DWTA decision makings, on the tracks of targets and their speeds is accurate enough. From the viewpoint of problem solving, there are no essential differences between the decision makings at different stages. Generally speaking, however, the DWTA decision making in later stages is easier due to the reduction of targets and available weapons.

III. DWTA Algorithms

A. *VP for Representation of DWTA Solutions*

As a first step to guarantee the feasibility of generated solutions, the concepts of available assignment pair (AAP) are proposed for the satisfaction of the constraints in (5).

Definition 3 (AAP): An assignment pair denoted by $i - j - t$, indicating that weapon i is assigned to engage target j in stage t, is called an AAP iff $f_{ij}(t) = 1$.

Remark 2: Only AAPs are considered in the assignment process so that all constraints in (5) will not be violated. In fact, all

TABLE I
PSEUDOCODE FOR CP

```
Procedure Construction ($P_e$)  // $P_e$ is a permutation of all AAPs.
  Initialize the saturation states of the constraints in (2), (3) and (4).
  Let $\mathbf{X} = \mathbf{O}_{WxTxS}$  // a zero matrix
  For $k$ = 1 to size($S_{AAP}$) // $S_{AAP}$ is the set of all AAPs.
      If the assignment of the $k$th pair in the list $L$ denoted by $i_k$ - $j_k$ - $t_k$ does
            not cause any violation of constraints
      // $L$ is the list formed by all AAPs in the order corresponding to $P_e$.
          Let $x(i_k, j_k, t_k)$ = 1 and add the AAP into the list of AAAPs.
          Update the saturation states of constraints related to the assigned
            AAP.
          If all constraints in (2), (3) or (4) are saturated
              Break//The procedure is over since any additional assignment
                          will cause the violation of certain constraint.
          End If
      Else  Add the AAP into the list of UAAPs.
      End If
  End For
  Return X
End Procedure
```

AAPs can be determined by analyzing the time windows of each weapon-target pairing. For example, the prediction method employed by Karasakal [11] or the cue-generation algorithm employed by Blodgett *et al.* [24] can be used as a feasible analysis approach.

Definition 4 (Assigned AAP): For a global decision, any AAP contained in the decision is called an AAAP. Obviously, all AAAPs constitute a DWTA solution.

Definition 5 (Unassigned AAP): For a global decision, any AAP that is not contained in the decision is called a UAAP. Note that an AAP may be an AAAP for one decision (solution), but a UAAP for another.

Definition 6 (Virtual permutation): The permutation of all AAPs is termed as VP. Assume, for instance, that there are three AAPs in total, denoted by AAP1 ($i_1 - j_1 - t_1$), AAP2 ($i_1 - j_2 - t_2$), and AAP3 ($i_2 - j_1 - t_2$), respectively. For example, (AAP2, AAP1, AAP3) and (AAP3, AAP1, AAP2) are two VPs.

Remark 3: A VP can be regarded as an *indirect* representation of certain DWTA solution. However, it needs a further treatment to be transformed into a real DWTA solution. In the following section, a CP is proposed to complete this task.

B. Construction Procedure

Since the satisfaction of constraints in (5) has been ensured, the goal of the CP is to generate feasible DWTA solutions, which satisfy all constraints in (2)–(4). To achieve this, the concept of saturated constraint is introduced first.

Definition 7 (Saturated constraint): A constraint $c(\mathbf{X}) \leq 0$ is said to be saturated iff the decision $\mathbf{X}$ results in $c(\mathbf{X}) = 0$.

At the beginning of the CP, all constraints in (2)–(4) are unsaturated. As more and more AAPs are assigned during the construction process, some constraints may become saturated. The pseudocode for the CP is presented in Table I.

The aforementioned procedure will add each AAP into a DWTA solution according to the order of AAPs in the corresponding VP if the operation does not violate the constraints in (2)–(4). The combination of VP and CP lays a foundation for utilizing various heuristics, such as TS and GA, to solve the DWTA problem formulated in Section II without constraint violation. Each permutation can be used to produce only one feasible decision because the CP is deterministic. All feasible decisions can be generated by certain permutations, but a feasible decision may not correspond to a unique VP. In the following text on the design of different DWTA solvers, we will elaborate some approaches to avoid generating the same decision repeatedly. It should be noted that the increase in the use of any effective weapon without the violation of constraints will further improve the objective value, and thus, lead to a better decision. Therefore, the decision scheme in which all APPs are assigned will be optimal if it does not cause any violation of constraints. In general, however, the assignment of all APPs is infeasible due to the limitation of the constraints in (2)–(4). Although the aforementioned CP cannot ensure the optimality of generated decisions, it greatly reduces the search scope of the DWTA problem. This is because the generated decision with constraint saturation is the best feasible one among all feasible decisions generated by the same permutation with or without constraint saturation, and those inferior "unsaturated" feasible decisions will not be produced.

C. Basic Operation of Generating New Solutions

Since the aforementioned CP is a deterministic process, the order of AAPs in a VP is the decisive factor determining the formation of a DWTA solution. Thus, it is necessary, for iterative algorithms, to change the order of AAPs so as to generate new solutions from old ones. As mentioned earlier, a DWTA solution may correspond to multiple VPs. Therefore, the AAP order has to be adjusted in a deliberative manner to avoid repeatedly generating the same solution. For this purpose, it is essential to analyze the characteristics of aforementioned AAAPs and UAAPs at first. AAAPs contained in the same solution have no conflict against each other w.r.t. objective function and constraints. On one hand, this is because they correspond to different decision components of the solution. On the other hand, the aforementioned CP ensures that AAAPs are compatible w.r.t. constraints. Therefore, a solution will not be changed by only rearranging the order of AAAPs in the corresponding VP. This is the same case for the means of changing the order of UAAPs. Consequently, the only feasible way of generating a new solution is to exchange the positions of UAAPs and their AAAP competitors in the VP. Note that an AAAP is called the competitor of a UAAP if the AAAP has conflict against the UAAP w.r.t. at least one constraint. A basic operation that can guarantee the birth of a new solution is presented as follows.

MtH operation: Select m UAAPs ($1 \leq m \leq \text{size}(S_{\text{UAAP}})$) from a VP and move them, one by one, to the head of the VP. Note that S_{UAAP} is the set of all UAAPs in a VP.

It is clear that the UAAP rearranged at the head of the new VP will become an AAAP in the solution corresponding to the generated VP. Therefore, the MtH operation followed by CP can generate a new solution different from the original one. This operation will be utilized by the DWTA algorithms presented in the following section.

TABLE II
PSEUDOCODE FOR LS

```
Procedure Local_search ($P_e$, $\mathbf{X}$) // $P_e$ is a permutation of all AAPs.
  //Record the current best decision during the local search and its
      corresponding VP: $\mathbf{X}^* = \mathbf{X}$, $P_e^* = P_e$
  Repeat
    Implement an MtH operation on $P_e$ to get a new VP $P_e'$.
    Apply the construction procedure on $P_e'$ to get a new decision $\mathbf{X}'$.
    If $J(\mathbf{X}') > J(\mathbf{X}^*)$ Then $\mathbf{X}^* = \mathbf{X}'$, $P_e^* = P_e'$, and update the tabu list End If
  Until $\forall\ P_e' \in N(P_e)$: $J(\mathbf{X}') \leq J(\mathbf{X}^*)$ // $N(P_e)$ is the neighborhood of $P_e$,
      that is, the set of VPs which can be generated from $P_e$ by MtH
      operations.
  Return $P_e^*$ and $\mathbf{X}^*$
End Procedure
```

D. Tabu Search

The TS algorithm originated by Glover [32] is a well-known intelligent computation tool for solving combinatorial optimization problems. It is an extension of local neighborhood search by virtue of "memory" (i.e., so-called tabu list) to avoid both frequent cycling of operations and revisiting local optima. The memory function of tabu list is one of the most important mechanisms in TS to escape from local optima. The design of the TS for DWTA involves the following four aspects.

1) *Initialization:* In order to ensure the efficiency of TS in solving DWTA problems, a *construction* heuristic is employed to provide initial solutions, which is somewhat similar to that in [24]. First, the heuristic ranks all AAPs according to their "values," which are defined as the product of the value of aimed assets, the lethality probability that targets kill their aims, and the lethality probability that weapons kills engaged targets. Then, all AAPs will be arranged in a VP in the descending order of their ranks. In other words, AAPs with larger values will be preferred as the components of an initial solution. The construction heuristic in essence is similar to a *greedy* algorithm.

2) *Neighborhood and LS:* The MtH operation presented earlier will be employed here to implement LS. In particular, we choose the parameter $m = 1$ for the operation. In each operation, a nontabu AAP will be selected and moved to the head of the original VP to form a new VP, which will be further converted into a DWTA solution by CP. For any VP, all VPs that can be generated from it by the MtH operation ($m = 1$) constitute its neighborhood. An LS will be terminated if all neighbors of the starting VP are visited. The pseudocode for the LS is presented in Table II.

3) *Tabu list and tabu tenure:* After each MtH operation during LS, the generated UAAPs, which are AAAPs in the original solution, will be put into a tabu list if the resulting solution brings the improvement of objective value relative to that of the original solution. The UAAPs in the tabu list will not be considered as the candidate for MtH operations. They will be released after a tenure of t_e times LS. The tenure parameter t_e is randomly selected from a predefined range with its lower and upper bounds denoted by $t_{\min}$ and $t_{\max}$, respectively.

The improvement of objective value can be ascribed to the addition of a new AAAP. While the new AAAP is imported, some old AAAPs are excluded and changed into UAAPs in the new solution, indicating that these excluded AAAPs comparatively have less contribution to objective value. However, the excluded AAAPs may be reused immediately in subsequent operations to produce solutions previously visited, which results in an undesirable short-term cycling. This is the consideration based on which the tabu list is designed.

4) *Diversification strategy (DS):* The tradeoff between diversification and intensification is a common and crucial issue in the design of metaheuristics [33], [34]. The LS mentioned earlier can be viewed as an implementation of intensification. Here, two diversification strategies are proposed as follows in order to achieve a better tradeoff.

Diversification strategy 1 (DS1): If no better solution can be found in the neighborhood of a solution, an MtH operation with the value of its parameter m selected from $\{2, 3, \ldots, \mathrm{size}(S_{\mathrm{UAAP}})\}$will be executed on the best VP found so far, and followed by CP to provide a starting solution for subsequent LS. Note that the selection of UAAPs in this case is not confined by the tabu list, which is to say, any UAAP can be chosen for the MtH operations during diversification. The probability for $m = i$ ($i \in \{2, 3, \ldots, \mathrm{size}(S_{\mathrm{UAAP}})\}$) is set as follows:

$$\mathrm{prob}(m = i) = \frac{2(r + 1 - i)}{r(r - 1)}, \quad \text{with} \quad r = \mathrm{size}(S_{\mathrm{UAAP}}).$$

Obviously, smaller values are preferred with the aforementioned probability distribution function for the setting of *m*, which tends to benefit the inheritance of good building blocks in the current best solution.

This DS can *de facto* be categorized as a random restart strategy based on a diversified version of the global best solution [34].

Diversification strategy 2 (DS2): If no better solution can be found in the neighborhood of a solution, all AAPs will be randomly permutated to generate a new starting VP for subsequent LS. In contrast to the first DS, this strategy is of a pure random restart type.

Remark 4: As a common ingredient of TS, aspiration criteria are often utilized to allow the tabu status to be overridden in cases where the forbidden UAAPs exhibit desirable properties [34]. In fact, this measure has been incorporated into the DS in which the tabu function will be disregarded. The diversification brings more opportunities for those excluded AAPs to compete against their competitors, which are inferior to them. No doubt, it is one of the most important mechanisms for TS to be endowed with the ability of global optimization.

To sum up, the flow of the proposed TS algorithm is stated as follows.

Step 1 (Initialization): 1) Use the "greedy" construction heuristic to obtain the initial virtual permutation VP_0 and implement the CP to generate the corresponding DWTA decision $\mathbf{X}_0$. 2) Evaluate the objective value of the initial decision and record it as the current best one: $\mathbf{X}_{\mathrm{best}} = \mathbf{X}_0$, $\mathrm{VP}_{\mathrm{best}} = \mathrm{VP}_0$, and $J_{\mathrm{best}} = J(\mathbf{X}_0)$. 3) Set the starting VP and its corresponding

TABLE III
COMPUTATIONAL COMPLEXITY OF THE TS ALGORITHM TS-DS1 FOR DWTA

Main Operations		Time Complexity	Space Complexity
Initialization (Construction heuristic)	Ranking	$O(L \cdot \log L)$	$O(size(S_{AAAP}))+O(size(S_{UAAP}))=O(L)$
	CP	$O(L)$	$O(L)+O(W \cdot S+T \cdot S+W)=O(L+W \cdot S+T \cdot S+W)$
	Function Evaluation	$O(W \cdot T \cdot S)$	$O(1)$
Main Loop	LS: <MtH (m=1)+CP>	$O(L)$	$O(L+W \cdot S+T \cdot S+W)$
	DS1: <MtH (1<m≤L)+CP>	$O(L)+O(m)=O(L)$	$O(L+W \cdot S+T \cdot S+W)$
	Function Evaluation	$O(W \cdot T \cdot S)$	$O(1)$
Total		$O(N_{\mathrm{fe}} \cdot W \cdot T \cdot S+L \cdot \log L)$	$O(L+W \cdot S+T \cdot S+W)$

Note that L denotes the total number of AAPs, and, obviously, $L \leq W \cdot T \cdot S$. LS and CP are the abbreviations of local search and construction procedure, respectively. N_{fe} represents the number of function evaluations.

decision and objective value: $\mathrm{VP}_s = \mathrm{VP}_0$, $\mathbf{X}_s = \mathbf{X}_0$, and $J_s = J(\mathbf{X}_0)$.

Step 2 (Main Loop): If termination criteria are satisfied, output the current best decision and its objective value and stop the algorithm. Execute the procedure **Local_search**(VP_s ,$\mathbf{X}_s$). If no decisions are better than the starting solution during the LS, adopt the diversification strategy, DS1 or DS2, to acquire a new starting virtual permutation VP_s and its corresponding decision $\mathbf{X}_s$, and then, rerun Step 2; otherwise, denote by $\mathbf{X}_l$ and VP_l the discovered locally optimal decision and its VP, respectively. Let $\mathbf{X}_{\mathrm{best}} = \mathbf{X}_l$, $\mathrm{VP}_{\mathrm{best}} = \mathrm{VP}_l$, and $J_{\mathrm{best}} = J(\mathbf{X}_l)$ if $\mathbf{X}_l$ is better than $\mathbf{X}_{\mathrm{best}}$. Choose the local optimum as a new starting solution: $\mathbf{X}_s = \mathbf{X}_l$, $\mathrm{VP}_s = \mathrm{VP}_l$, and $J_s = J(\mathbf{X}_l)$. Rerun Step 2.

Remark 5: For convenience of differentiation, we term the TS variants with the first and second diversification strategies as TS-DS1 and TS-DS2, respectively.

E. Computational Complexity Analysis

Considering the main operations involved in the proposed TS algorithm (TS-DS1) for DWTA, we summarized the worst-case computational complexity of the algorithm in Table III. The total computational complexity can be divided into two parts—the complexity contributed by initialization and that caused by the main loop of the algorithm. The initialization includes the ranking of all AAPs, the CP, and function evaluation for the initial solution. It should be noted that the time complexity of a desirable algorithm for an L-sized ranking (sorting) problem is $O(L \cdot \log L)$ [35]. Besides, the ranking algorithm will take up at most L memory units to achieve its function [35]. In order to ensure the feasibility of a solution, the CP will check, in the worst case, the saturation states of all constraints regarding all AAPs, which will take at most L basic operations. Meanwhile, the CP will use L, $W \times S$, $T \times S$, and W memory units, respectively, to record the saturation states of four kinds of constraints. Considering that each target can threaten at most one asset since we have assumed that their aims are exposed, we will only implement the computation regarding each asset and its threats. Besides, it is clear that the number of targets is not less than that of the threatened assets ($T \geq K$). Therefore, the time complexity of the objective function evaluation [see (1)] will depend on the numbers of the targets, weapons, and stages, regardless of that of the assets (K). From (1), the worst-case time complexity of function evaluations can be easily derived as $O(W \cdot T \cdot S)$.

The main loop of the algorithm also includes three parts—LS, diversification, and function evaluation. The computational complexity of LS is mainly caused by its CP, since the MtH operator involved only implements a simple move operation (see Section III-C). Due to the same reason, the computational complexity of diversification is comparative with that of LS.

To sum up, we can conclude that the total time complexity of the algorithm TS-DS1 is mainly caused by function evaluation. This is because the ranking operation is implemented only once at initialization and the time complexity of ranking and the other operations is negligible in contrast to that of function evaluation. Besides, the space complexity of TS-DS1 is mainly caused by the CP.

With the setting of the maximal number of function evaluations $N_{\mathrm{fe}} = 5W \cdot T \cdot S$ employed in Section IV, the worst-case time complexity of TS-DS1 can be further simplified as $O(W^2 \cdot T^2 \cdot S^2)$. This is because the term $O(L \cdot \log L)$is comparatively negligible, due to $L \leq W \cdot T \cdot S$.

The TS variant TS-DS2 has the same worst-case space complexity, but a slightly heavier runtime complexity. The diversification strategy DS2 relies on a random permutation of all AAPs, whose time complexity is akin to that of a ranking algorithm. The theoretical results of time complexity analysis agree very well with the observations, as shown in Table VIII, in our experiments on scalability performance.

F. HGA and Hybrid ACO for DWTA

Both GAs and ACOs are recognized as efficient optimizers for solving complicated optimization problems, which is validated by large numbers of applications of the two optimizers [9], [13], [14], [17], [36]–[40]. Since it has been widely validated that LS can improve the performance of GAs [41]–[43], we adopt here an HGA combining the respective advantages of GA and LS in global exploration and local exploitation. The HGA is also termed as memetic algorithm or genetic LS in the literature [41]–[43]. For ACO, we also incorporate LS techniques into its optimization procedure. In fact, LS techniques are usually included in state-of-the-art ACO algorithms [44]. In order to use HGA and hybrid ACO (HACO) to solve DWTA problems, the basic operations in the two population-based metaheuristics (e.g., the crossover and mutation operators in HGA) have to be elaborately designed. The operations embedded in HGA and

HACO are adapted to DWTA problem solving in the following way.

1) *Solution representation (for HGA)—VP followed by CP:* This is the same strategy employed in the aforementioned TS heuristics.
2) *Crossover (for HGA)—Two-parent crossover for one child:* Two individuals in the current population will be randomly selected as parents. The same AAAPs shared by the two parents will be kept in the offspring and moved to the head of the offspring's VP. This strategy follows the basic idea in evolutionary computation that "good" genes should be inherited.
3) *Mutation (for HGA)—Two-exchange mutation:* Randomly select an offspring generated by crossover. Select a UAAP and one of its relevant AAAPs from the offspring to be mutated, and exchange their positions to generate a new VP. Note that one AAP is said to be relevant to another AAP iff they share at least one identical element (weapon, target, or stage), which means that they have potential conflicts against each other w.r.t. certain constraints. Bearing much similarity with the MtH operator earlier, the exchange of a UAAP and its relevant AAAPs will produce a new DWTA solution. In fact, the MtH operator can also be used for mutation operation. However, the two-exchange mutation can give birth to more different offspring than the MtH operator, since only one AAAP can be determinately replaced by selected UAAP in the MtH operation.
4) *Selection (for HGA)—(μ+ λ) ES (evolution strategy) selection, where μ is population size and λ is the number of offspring created:* The process will eliminate redundant individuals and retains the best μ individuals. This selection scheme has been proved efficient in achieving the exploration–exploitation tradeoff in many researches and widely used [19], [41], [45].
5) *LS (for both HGA and HACO):* We apply the LS used in our TS heuristics to GA and ACO. Refer to Section III-D and Table II for details on the LS procedure. In addition, it should be noted that the tabu list is not used when the LS is applied to GA and ACO.
6) *Restart-based diversification (for HGA):* If the average fitness (objective value) of the population is not changed for more than 30 generations [41], the whole population will be randomly reinitialized and evolve from new starting points. This strategy represents an escape mechanism from suboptimal regions of the search space.

To sum up, the operation of HGA can be concisely depicted as follows:

$$P(t+1) = \begin{cases} \mathrm{LS}\{\underbrace{P(t)}_{\substack{\text{Randomly}\\ \text{generated}}}\}, \text{ if HGA is restarted} \\ S\{\overbrace{\underbrace{\mathrm{LS} * M * C * \mathrm{LS}}_{\text{Composite Operation}}\{P(t)\}}^{\text{Generate } \lambda \text{ Offspring}} \cup \overbrace{\mathrm{LS}\{P(t)\}}^{\mu \text{ Parents}}\}, \text{otherwise.} \end{cases}$$

where $P(t)$ is the population at the *t*th generation, S, M, C and LS represent the selection, mutation, crossover, and LS operations, respectively. Note that the crossover and mutation operations will be implemented λ times at each evolutionary generation to generate λ offspring.

For fair comparisons with proposed TS algorithms, the first individual in the initial population of both HGA and HACO is provided by the "greedy" construction heuristic employed by the TS algorithms (see Section III-D). The remaining initial individuals in the form of VPs will be randomly generated and kept identical for HGA and HACO.

The operation of the HACO can be briefly described by the following pseudocode [46]:

```
Procedure HACO
  Set parameters, initialize pheromone trails
  Repeat
    ConstructAntsSolutions
    ApplyLocalSearch
    UpdatePheromoneTrails
  Until termination criteria are satisfied
End Procedure
```

7) *Solution construction and pheromone update (for HACO) [46], [47]:* In ACO, ants (i.e., solutions in the form of VP) are generated by constructive heuristics. The components of each VP will be generated one by one according to the following pheromone-based selection probability:

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k(t)} [\tau_{il}(t)]^{\alpha}[\eta_{il}(t)]^{\beta}}$$

$$j \in N_i^k(t), \qquad k = 1, 2, \ldots, P_S$$

where $P_{ij}^{k}(t)$ is the probability of choosing AAP_j for the *i*th position of the VP for ant *k* at generation *t*, $\tau_{ij}(t)$ is the corresponding pheromone trail intensity; initially, all trail intensities are set at the same value $\tau_0 = P_S J^*$, where J^* is the objective value of the first ant provided by the aforementioned "greedy" construction heuristic, which shares much similarity with the setting suggested in the literature [46]. $\eta_{ij}(t)$ is the corresponding heuristic value (say "visibility" in the literature [46], [47]); we set $\eta_{ij}(t) = \mathrm{value}(\mathrm{AAP}_j)$(see Section III-D for the definition of AAP value); a larger AAP value tends to result in a larger selection probability. α is trail importance factor; we set $\alpha = 1$according to the suggestion in the literature [46], β is visibility importance factor; we set $\beta = 2$ [46], P_S is the number of ants (population size), and $N_i^k(t)$ is the set of all APPs allowed for the *i*th VP position of ant *k* at generation *t*.

The updating method of pheromone trail intensity is shown as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{P_s} \Delta\tau_{ij}^{k}(t)$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} J^k, & \text{if } \mathrm{VP}_i^k = \mathrm{AAP}_j \\ 0, & \text{otherwise} \end{cases} .$$

where ρ is pheromone evaporation rate $(0 < \rho < 1)$; this factor is crucial to the tradeoff between intensification and diversification for ACO, a large ρ will benefit diversification (exploration) and weaken intensification (exploitation), and *vice versa*; we set $\rho = 0.5$ according to the suggestion in the literature [46], $\Delta\tau_{ij}(t)$is the incremental trail intensity of the pair (*i*, *j*) at generation *t*, $\Delta\tau_{ij}^{k}(t)$ is the contribution of ant *k* to $\Delta\tau_{ij}(t)$, J^k is the objective value of ant *k*, and VP_i^k is the *i*th VP component of ant *k*.

It is evident that ants with larger objective values have more contribution to pheromone increase, which potentially draws more attention to promising regions in the search space.

IV. Computational Results

In this section, a random sampling (RS) method, an LS based on MtH operations and random restart strategy (LS-MtH&RR), the proposed TS algorithms including TS-DS1 and TS-DS2, and HGA and HACO will be experimentally compared. Note that RS is a statistical method without the use of information accumulated during its sampling process. Typically, each sampling (i.e., the generation of each VP) is achieved by a pure random permutation of all AAPs like the restart approach employed in TS-DS2. In the family of iterative search methods, RS is the easiest one, but it is insensitive to local optima due to the independence among samplings. It is shown in our previous research that RS will become an efficient optimizer when the global optimization of problems relies heavily on exploration [48]. LS-MtH&RR is essentially the LS method presented in Table II with the MtH operation ($m = 1$) employed. LS-MtH&RR adopts the same construction heuristic, as that in TS-DS1, to obtain its initial solution. When an LS is over, the search process will be restarted by the random permutation approach employed in DS2. In essence, LS-MtH&RR can be regarded as a multistart LS method.

In order to make a general performance analysis of these algorithms, a DWTA test-case generator will be constructed later to produce DWTA instances of different scales for a scalability test of DWTA algorithms. The tested algorithms will be terminated if the maximal number of function evaluations (N_{fe}) is reached. All algorithms were performed on a PC with Intel(R) 2.8 GHz CPU and 4.0 GB internal memory.

For HGA, the tested values for its parameters μ and λ include $\mu =$ 10, 20, 50, 100, and 200 and λ = 1, 5, 10, 20, 50, 100, 200, and 400. We used six DWTA instances of different scales provided by the test-case generator presented in the following section to make an experimental analysis of these settings. According to our preliminary test results, the combination of $\mu = 20$ and λ = 20 gives the best performance in almost all cases. Therefore, we used this setting for HGA throughout the comparative experiments later. For HACO, we also tested similar settings, including $P_s =$ 10, 20, 50, 100, and 200, in the same way. It was observed that smaller population sizes especially $P_s = 20$ lead to better results. Accordingly, we chose $P_s = 20$ for the population sizing of HACO.

A. DWTA Test-Case Generator

As a preparation for testing the performance of different algorithms in solving DWTA problems of different scales, we developed a test-case generator at first. Given the input parameters W, T, K, and S, the generator will automatically produce the DWTA parameters $\mathbf{V} = [v_k]_{1\times K}$, $\mathbf{Q} = [q_{jk}]_{T\times K}$, $\mathbf{P} = [p_{ij}(t)]_{W\times T\times S}$, $\mathbf{F} = [f_{ij}(t)]_{W\times T\times S}$, and $\mathbf{N} = [N_i]_{1\times W}$. The details are presented as follows.

1) $\mathbf{V} = [v_k]_{1\times K}$: $v_k = 10 + 90 * \mathrm{rand}$, $k = 1, 2, \ldots, K$:

The values of assets are randomly generated in the interval (10,100). Each asset is threatened by at least one hostile target, which means $T \geq K$. In our research, we set $T = K$. For convenience without loss of generality, we assume that the aim of the *k*th target is the *k*th asset ($k = 1, 2, \ldots, K$).

2) $\mathbf{P} = [p_{ij}(t)]_{W\times T\times S}$, $\mathbf{Q} = [q_{jk}]_{T\times K}$:

$$p_{ij}(t) = p_L + (p_H - p_L)\mathrm{rand}_1, \text{for } i = 1, 2, \ldots, W;$$
$$j = 1, 2, \ldots, T; t = 1, 2, \ldots, S$$
$$q_{j,k(j)} = q_L + (q_H - q_L)\mathrm{rand}_2, \text{for } j = 1, 2, \ldots, T$$

where p_L and p_H are the lower and upper bounds of the lethality probability that weapons kill engaged targets, respectively, q_L and q_H are the lower and upper bounds of the lethality probability that targets kill their aims, respectively, $k(j)$ denotes the serial number of the asset threatened by the *j*th target, rand_1 and rand_2 are random numbers generated from the interval (0,1).

3) $\mathbf{F} = [f_{ij}(t)]_{W\times T\times S}$:

$$f_{ij}(t) = [\mathrm{sign}(\mathrm{rand}_3 - p_f) + 1]/2, \text{for } i = 1, 2, \ldots, W;$$
$$j = 1, 2, \ldots, T; t = 1, 2, \ldots, S$$

where the function $\mathrm{sign}(\cdot)$ is equal to 1 if its argument is positive, -1 otherwise, rand_3 is a random number generated from the interval (0,1), and p_f is the probability that $f_{ij}(t)$ is equal to 0 and we set $p_f = 0.5$ in the following test.

4) $\mathbf{N} = [N_i]_{1\times W}$:

The parameters N_i ($i = 1, 2, \ldots, W$) are randomly selected from the integer set $\{1, 2, \ldots, S\}$.

B. Parameter Settings for TS-DS1 and TS-DS2

Considering that each LS will visit at most $\mathrm{size}(S_{\mathrm{UAAP}})$ solutions, we relate the setting of the two tenure parameters $t_{\min}$ and $t_{\max}$ to the problem-dependent parameter $\mathrm{size}(S_{\mathrm{UAAP}})$. The following settings for the two parameters $[t_{\min}, t_{\max}]$ are included for a comparative numerical analysis.

$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 0.2]$ (No. 1), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 0.3]$ (No. 2),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 0.5]$ (No. 3), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 0.7]$ (No. 4),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 1.0]$ (No. 5), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 1.5]$ (No. 6),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.1, 2.0]$ (No. 7), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.3, 0.5]$ (No. 8),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.3, 0.7]$ (No. 9), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.3, 1.0]$ (No. 10),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.3, 1.5]$ (No. 11), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.3, 2.0]$ (No. 12),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.5, 0.7]$ (No. 13), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.5, 1.0]$ (No. 14),
$\mathrm{size}(S_{\mathrm{UAAP}}) * [0.5, 1.5]$ (No. 15), $\mathrm{size}(S_{\mathrm{UAAP}}) * [0.5, 2.0]$ (No. 16),

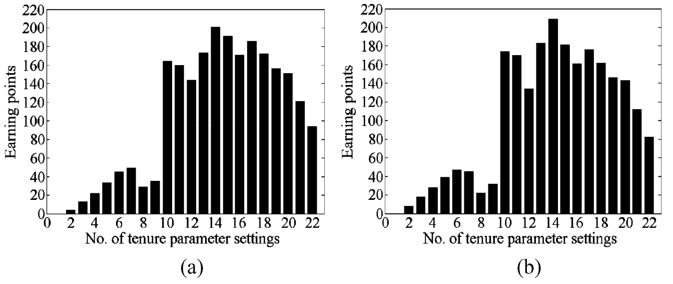


Fig. 1. Statistical results on the relative performance of TS algorithms with different tenure parameter settings. (a) TS-DS1. (b) TS-DS2.

$\text{size}(S_{\text{UAAP}}) * [0.7, 1.0]$ (No. 17), $\text{size}(S_{\text{UAAP}}) * [0.7, 1.5]$ (No. 18),

$\text{size}(S_{\text{UAAP}}) * [0.7, 2.0]$ (No. 19), $\text{size}(S_{\text{UAAP}}) * [1.0, 1.5]$ (No. 20),

$\text{size}(S_{\text{UAAP}}) * [1.0, 2.0]$ (No. 21), $\text{size}(S_{\text{UAAP}}) * [1.5, 2.0]$ (No. 22).

According to the analysis results on computational complexity presented in Table III, we relate the setting of the maximal number of function evaluations (N_{fe}) to the problem-dependent parameter $W \cdot T \cdot S$, which determines the size of DWTA problems to a large extent. In order to identify a desirable setting of tenure parameters at first, we set N_{fe} to be a large enough number $50W \cdot T \cdot S$ to ensure stable results. We used ten different DWTA instances, randomly generated by the test-case generator, to check which combination of the aforementioned settings can produce the best results. To save space, we will not present the details on these instances. Regarding each instance, we ran the TS algorithms with each setting 50 times and used the *Wilcoxon* signed-rank test [49] with 95% confidence level to carry out paired comparisons. In each paired comparison, the setting producing better results than another one with 95% confidence level will earn one point. The total points earned by each setting over ten DWTA instances are presented in Fig. 1. The results shown in Fig. 1 indicate that, for both TS-DS1 and TS-DS2, the tenure parameter setting labeled as No. 14, i.e., $[t_{\min}, t_{\max}] = \text{size}(S_{\text{UAAP}}) * [0.5, 1.0]$, is a better choice.

For the sake of curtailing running time while retaining decision-making quality, we also analyzed the time with which stable results were achieved during the aforementioned test for the determination of tenure parameters. In the aforementioned test, we recorded the accumulated number of function evaluations when TS algorithms found their best solutions in each case. It should be noted that what we concern here is the recorded number with regard to the best tenure parameters. Denote by N_r the recorded number. Then, a statistical analysis of the index $N_r/(W \cdot T \cdot S)$ will benefit appropriate settings for N_{fe}. The statistical results of this index are shown in the form of box plots in Fig. 2. It is easy to see from Fig. 2 that $N_{\text{fe}} = 5W \cdot T \cdot S$ is a good choice for the setting of maximal number of function evaluations. With this setting, we can observe the stable results, without large extra time cost, for all of the ten DWTA instances. Based on the aforementioned observations, we adopted the parameter settings $[t_{\min}, t_{\max}] = \text{size}(S_{\text{UAAP}}) * [0.5, 1.0]$ and $N_{\text{fe}} = 5W \cdot T \cdot S$ for subsequent experiments on performance comparisons.

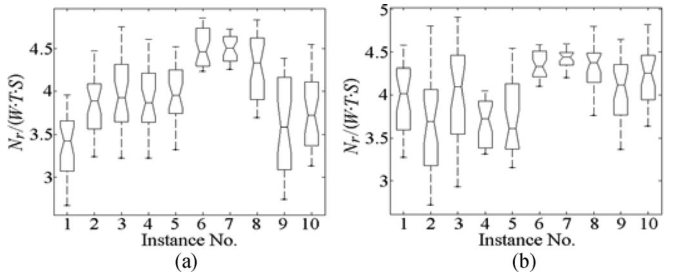


Fig. 2. Statistical results on the index N_r/(WTS) for TS algorithms with best tenure parameters. (a) TS-DS1. (b) TS-DS2.

C. Simple DWTA Tests

All DWTA algorithms involved will first go through the performance comparisons by two simple, but typical DWTA instances originated, respectively, from naval single-platform combat and ground-based air defense systems (GBADS). In order to evaluate the performance of tested algorithms for the whole defense process, the Monte Carlo method (MCM) was executed to statistically simulate the result of stochastic events including the damage of targets and assets. Assume that the probability that an event occurs is p and the MCM generates a pseudorandom number rand $(0 < \text{rand} < 1)$. The event is confirmed to occur iff $\text{rand} < p$. The simulation of the whole decision-making process follows the DWTA procedure mentioned in Section II. Each algorithm runs 50 times for the decision making on each DWTA instance.

1) Single-Platform Combat—A Naval Warfare Scenario: In the naval single-platform combat scenario, a warship without any assistance has to protect itself against the hostile targets, e.g., air-to-surface missiles (ASMs). There is only one asset (i.e., the warship itself) threatened by the enemies, which means $K = 1$. In this sense, the scenario can be viewed as a special case of the general DWTA problem formulated by (1)–(6). Note that the asset value has no essential influence on DWTA decision makings, since there is no need for the defender to make decisions on which asset to protect as a priority. Thus, we assume, for simplicity, that the value of the warship is one. The following are a brief description of a scenario similar to that used in the literature [24] and the indexes of weapons and targets.

> Three hostile ASMs are on their way to attack the warship. The warship has five available weapons, including two guns, two surface-to-air missile (SAM) launchers, and one close-in weapon system (CIWS) to defend itself.
>
> Weapons: *w*1—Gun, *w*2—Gun, *w*3—SAM, *w*4—SAM, and *w*5—CIWS; Targets: *t*1—ASM, *t*2—ASM, and *t*3—ASM.

The parameters for the DWTA instance are presented in Table IV. For comparison, the TS heuristic proposed by Blodgett *et al.* [24] for naval single-platform combat was also employed in this test. The statistical results are presented in Table V. The involved performance indexes include the following:

1) proportion of the times in which the warship is confirmed, according to simulation results, to survive the whole defense process (P_{ts});

TABLE IV
DWTA PARAMETERS FOR THE NAVAL SINGLE-PLATFORM COMBAT SCENARIO

W=5	T=3	S=3	K=1	v_1=1							
Q=$[q_j]_{1\times T}$	0.5	0.75	0.9	**N**= $[N_i]_{1\times W}$	1000	1000	2	2	3000		
P: $[p_{ij}(1)]_{W\times T}$	0.01	0	0	**P**: $[p_{ij}(2)]_{W\times T}$	0.03	0	0	**P**: $[p_{ij}(3)]_{W\times T}$	0.08	0.05	0.01
	0.01	0	0		0.02	0	0		0.06	0.03	0.03
	0.70	0.60	0		0.70	0.60	0.40		0.70	0.60	0.40
	0.80	0.70	0		0.80	0.70	0.60		0.80	0.70	0.60
	0	0	0		0	0	0		0.006	0.001	0.001
F: $[f_{ij}(1)]_{W\times T}$	1	0	0	**F**: $[f_{ij}(2)]_{W\times T}$	1	0	0	**F**: $[f_{ij}(3)]_{W\times T}$	1	1	1
	1	0	0		1	0	0		1	1	1
	1	1	0		1	1	1		1	1	1
	1	1	0		1	1	1		1	1	1
	0	0	0		0	0	0		1	1	1

TABLE V
STATISTICAL RESULTS OF DWTA TEST FOR THE NAVAL SINGLE-PLATFORM COMBAT SCENARIO

Index	RS	LS-MtH&RR	TS [24]	HGA	HACO	TS-DS1	TS-DS2
P_{ts}	0.36	0.46	0.46	0.54	0.48	**0.70**	0.66
$\tilde{P}_s$	0.3653 ± 0.036	0.4513 ± 0.036	0.4542 ± 0.031	0.4865 ± 0.022	0.4619 ± 0.043	**0.5109 ± 0.001 [6]**	0.5024 ± 0.015 [5]
Comput. time (sec)	0.013 ± 0.000	0.009 ± 0.000	0.015 ± 0.000	0.14 ± 0.01	0.07 ± 0.01	**0.007 ± 0.000 [6]**	0.009 ± 0.000 [4]

2) average expected probability, with respect to the first global decision, that the warship finally survives the whole defense process ($\tilde{P}_s$);
3) computation time (total decision-making time-barring simulation time).

Regarding the last two indexes, we employed the *Wilcoxon* signed-rank test (95% confidence level), like that employed in Section IV-B for parameter setting, to make paired comparisons. The earning points of TS-DS1 and TS-DS2 against their competitors are bracketed in Table V. As shown in Table V, TS-DS1 outperforms all of its competitors in the aspect of both decision-making quality and computation time. The earning point of TS-DS1 indicates that the advantage of TS-DS1 over any of its six competitors is statistically significant. TS-DS2 also has a desirable performance, and it is only inferior to TS-DS1. The advantage of TS-DS2 over LS-MtH&RR validates the effectiveness of the tabu mechanism, which is the unique difference between the two algorithms.

HGA performs better, with regard to decision-making quality, than RS, HACO, LS-MtH&RR, and the TS proposed in [24]. However, its time cost compared with the other algorithms is too large, which is mainly due to its crossover operations spending much time on the identification of common "genes" in parents. In contrast to HGA, HACO is less time-consuming, but its decision-making quality is worse.

2) Force Coordination—A Ground-Based Air Defense Scenario: In this DWTA test, each algorithm also ran through all stages, and the average total value of finally surviving assets (ATVFSA) in 50 runs was evaluated and compared. An illustration for the combat scenario is shown in Fig. 3. The assets, targets, and weapons are labeled by "a," "t," and "w" plus their serial numbers, respectively. The dashed lines with arrows in Fig. 3 indicate the attacking aims of targets. The parameters of the DWTA instance are presented in Table VI. The statistical results are presented in Table VII in the form of ATVFSA, the average expected total value of finally surviving assets (AETVFSA) analyzed at the first stage, and the computation time (total decision-making time-barring simulation time), respectively, plus their corresponding standard deviations. The earning points of TS algorithms based on the *Wilcoxon* signed-rank test (95% confidence level) are also included in Table VII. Note that AETVFSA is in fact the mean of finally discovered best objective values corresponding to the global decision at the first stage.

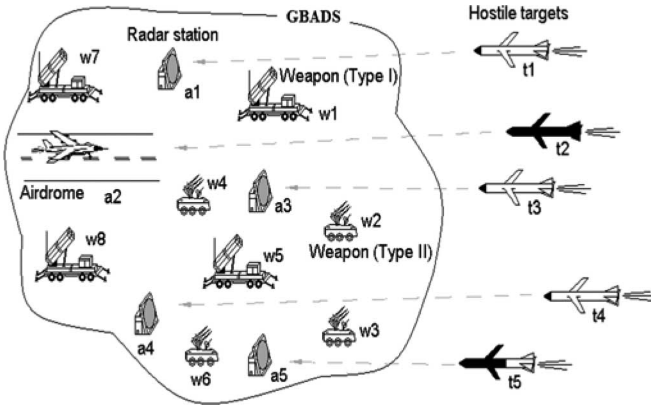


Fig. 3. Combat scenario for GBADS.

The main conclusions drawn from Table VII are very similar to those in the first DWTA test on naval single-platform combat. TS-DS1 in this test is also the best DWTA algorithm. TS-DS1 produced better DWTA decisions in a relatively shorter time. The advantage of two TS methods over RS, HACO, and HGA is prominent, since the time costs of the two TS variants with better or comparative decision-making quality are only about half of RS, and even much less than those of HGA and HACO. Again, TS-DS2 performs better than LS-MtH&RR, indicating the merit of tabu mechanism.

In addition, the results indicated by AETVFSA agree very well with those by ATVFSA, which rely on MCM simulations

TABLE VI
DWTA PARAMETERS FOR THE GROUND-BASED AIR DEFENSE SCENARIO

W	8							0.9	0	0	0	0
T	5							0	0.7	0	0	0
K	5					$\mathbf{Q} = [q_{jk}]_{T\times K}$		0	0	0.9	0	0
S	3							0	0	0	0.9	0
$\mathbf{V} = [v_k]_{1\times K}$	20	100	20	20	20			0	0	0	0	0.8
	0.8	0.6	0.8	0.8	0.7			1	1	1	0	0
	0.7	0.6	0.7	0.7	0.5			1	1	1	1	1
$\mathbf{P}$:	0.7	0.6	0.7	0.7	0.5			0	0	1	1	1
$[p_{ij}(1)]_{W\times T}$	0.7	0.6	0.7	0.7	0.5	$\mathbf{F}$:		0	0	0	0	0
$= [p_{ij}(2)]_{W\times T}$	0.8	0.6	0.8	0.8	0.7	$[f_{ij}(1)]_{W\times T}$		0	1	1	0	1
$= [p_{ij}(3)]_{W\times T}$	0.7	0.6	0.7	0.7	0.5			0	0	0	0	0
	0.8	0.6	0.8	0.8	0.7			0	0	0	0	0
	0.8	0.6	0.8	0.8	0.7			0	0	0	0	0
	1	1	0	1	0			0	1	0	0	0
	1	1	0	1	0			0	0	0	0	0
	0	0	0	1	0			0	0	0	0	0
$\mathbf{F}$:	0	1	1	0	0	$\mathbf{F}$:		1	0	0	1	0
$[f_{ij}(2)]_{W\times T}$	1	0	0	1	0	$[f_{ij}(3)]_{W\times T}$		0	0	0	0	0
	0	0	0	1	1			0	0	0	0	0
	1	1	0	0	0			0	0	0	1	0
	0	0	0	0	0			1	1	0	1	0
$\mathbf{N} = [N_i]_{1\times W}$	4	2	2	2	4	2	4	4	-	-	-	-

TABLE VII
STATISTICAL RESULTS OF DWTA TEST FOR THE GROUND-BASED AIR DEFENSE SCENARIO

Index	RS	LS-MtH&RR	HGA	HACO	TS-DS1	TS-DS2
ATVFSA	172.0 ± 2.0	174.4 ± 1.8	**179.6 ± 0.4**	176.8 ± 1.1	**179.6 ± 0.4 [4]**	177.6 ± 1.1 [2]
AETVFSA(1st stage)	163.5 ± 3.4	165.2 ± 2.6	170.8 ± 0.3	168.7 ± 0.2	**171.7 ± 0.0 [5]**	169.3 ± 0.4 [2]
Comput. time (sec)	0.024 ± 0.002	0.013 ± 0.001	0.25 ± 0.03	0.13 ± 0.01	**0.011 ± 0.001[5]**	0.014 ± 0.001[3]

TABLE VIII
STATISTICAL RESULTS ON SCALABILITY TEST OF DWTA ALGORITHMS

No.	Total asset value	Index	RS	LS-MtH&RR	HGA	HACO	TS-DS1	TS-DS2
1	662.15	AETVFSA	527.38 ± 3.35	638.28 ± 3.15	647.41 ± 4.74	645.38 ± 1.27	**651.13 ± 0.65 [5]**	642.15 ± 2.82 [2]
		Comput. time (sec)	0.65 ± 0.04	0.30 ± 0.02	4.17 ± 0.06	2.92 ± 0.08	**0.25 ± 0.01 [5]**	0.32 ± 0.02 [3]
2	2791.24	AETVFSA	2188.24 ± 9.53	2705.83 ± 15.06	2735.62 ± 10.86	2739.14 ± 9.73	**2750.61 ± 8.00 [5]**	2731.43 ± 7.96 [2]
		Comput. time (sec)	4.61 ± 0.19	2.15 ± 0.08	27.28 ± 0.07	14.74 ± 0.08	**1.33 ± 0.05 [5]**	2.21 ± 0.07 [3]
3	2791.24	AETVFSA	2459.83 ± 7.81	2688.52 ±22.64	2766.27 ± 9.12	2763.35 ± 6.88	**2769.62 ± 2.57 [5]**	2758.36.± 3.02 [2]
		Comput. time (sec)	41.34 ± 1.35	14.15 ± 0.76	183.15 ± 1.86	101.26 ± 1.22	**9.22 ± 0.31 [5]**	15.75 ± 0.50 [3]
4	5517.60	AETVFSA	4838.16 ± 11.52	5373.41 ± 27.13	5413.25 ± 17.36	5403.92 ± 15.24	**5452.73 ± 10.31 [5]**	5407.24 ± 10.25 [2]
		Comput. time (sec)	39.67 ± 1.32	12.54 ± 0.82	169.54 ± 1.82	93.71 ± 1.13	**8.51 ± 0.30 [5]**	14.12 ± 0.51 [3]
5	5517.60	AETVFSA	5211. 61 ± 10.32	5397.36 ± 21.28	5477.19 ± 9.37	5431.60 ± 7.61	**5486.38 ± 2.06 [5]**	5419.03 ± 2.13 [2]
		Comput. time (sec)	158.52 ± 3.26	48.72 ± 1.69	563.01 ± 2.85	375.38 ± 2.13	**37.25 ± 1.22 [5]**	51.37 ± 1.67 [3]
6	10944.51	AETVFSA	9281.77 ± 25.31	10744.53 ± 39.63	10834.62 ± 21.06	10842.22 ± 19.85	**10852.18 ± 15.11 [5]**	10815.29 ± 17.35 [2]
		Comput. time (sec)	610.01 ± 6.18	172.64 ± 4.02	981.73 ± 4.32	762.91 ± 4.15	**122.61 ± 3.02 [5]**	178.92 ± 3.85 [3]
7	10944.51	AETVFSA	9663.26 ± 20.63	10772.06 ± 37.61	10872.17 ± 17.33	10885.51 ± 15.49	**10902.74 ± 8.15 [5]**	10851.39 ± 9.81 [2]
		Comput. time (sec)	3152.18 ± 9.22	708.47 ± 6.23	4606.02 ± 9.14	3674.27 ± 8.53	**501.63 ± 5.30 [5]**	717.62 ± 5.81 [3]

to some extent. However, the values of the first stage AETVFSA in all cases are obviously smaller than those of ATVFSA. This is mainly because the stage-by-stage damage of targets decreases the defense pressure, and the DWTA problems for later stages turn easier.

D. Scalability Test Base on the Test-Case Generator

In this test, we used the DWTA test-case generator established in Section IV-A to compare the scalability performance of all DWTA algorithms involved. The number of stages (S) was fixed at four. The setting of the other input parameters W, T, and K includes the following cases:

$$W10T10K10 \text{ (No. 1)},\ W50T50K50 \text{ (No. 2)},\ W100T50K50 \text{ (No. 3)},\ W50T100K100 \text{ (No. 4)},\ W100T100K100 \text{ (No. 5)},\ W100T200K200 \text{ (No. 6)} \text{ and } W200T200K200 \text{ (No. 7)}.$$

For the two simple DWTA instances in previous sections, we utilized the MCM-based simulation to evaluate the global decision-making quality of different algorithms throughout all defense stages. This simulation-based multistage evaluation is beneficial to observe accumulated performance differences of these algorithms in making DWTA decisions at multiple consecutive stages. However, from the viewpoint of problem solving, there are no essential differences between the decision makings at different stages. In each stage, the decision-maker behaves

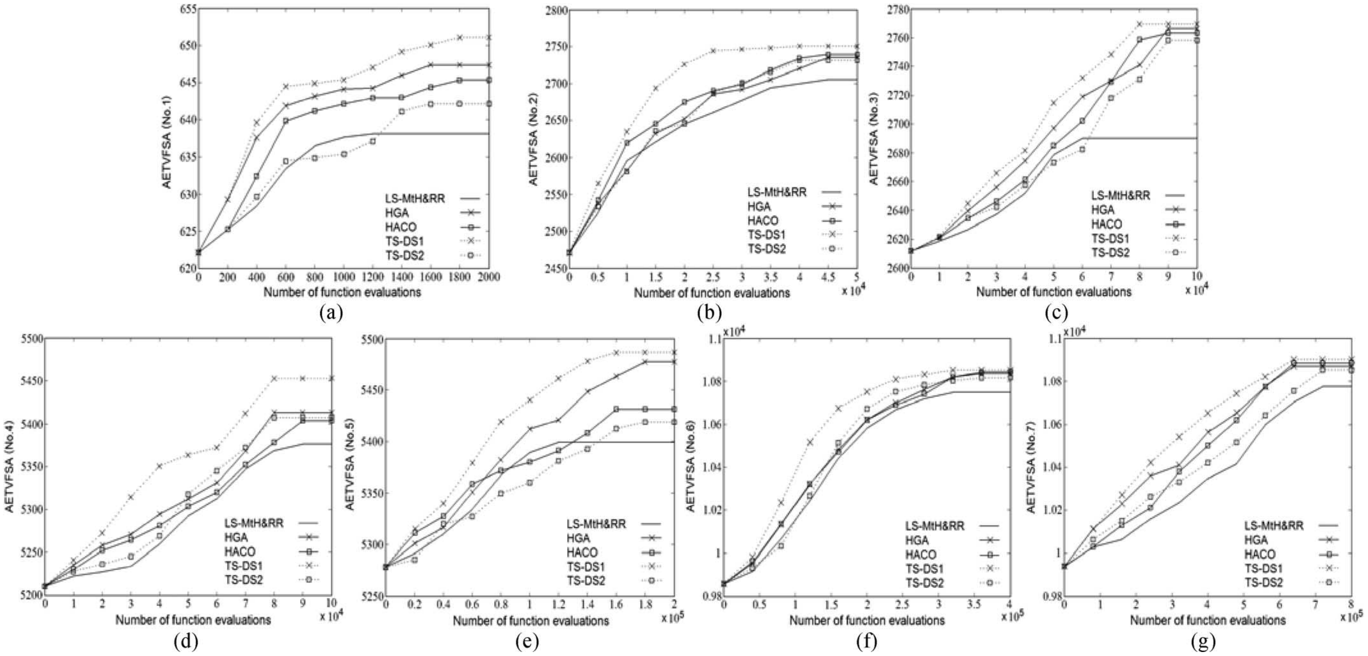


Fig. 4. Convergence curves of five DWTA algorithms.

identically, thinking about a "global" problem, but carrying out a "local" decision (i.e., executive decision). Therefore, for the purpose of revealing the performance differences of different algorithms, it is sufficient to only consider the global DWTA decision making in the first stage. The tested algorithms were used to make DWTA decisions for the aforementioned test cases (No. 1–No. 7) provided by the test-case generator. Experimental results are presented in Table VIII, including the statistical results on AETVFSA and computation time, as well as the earning points of TS algorithms based on the *Wilcoxon* signed-rank test (95% confidence level). The convergence curves for TS-DS1, TS-DS2, LS-MtH&RR, HGA, and HACO are shown in Fig. 4.

According to the statistical results in Table VIII, TS-DS1 outperforms RS, LS-MtH&RR, TS-DS2, HGA, and HACO in the aspect of decision-making quality as well as computation time. By comparing the results produced by RS and the starting values of convergence curves in each case, we can make sure that the greedy construction heuristic employed by LS-MtH&RR, HGA, HACO, and two TS variants, did provide them "good" starting points. TS-DS1 has obvious advantage over TS-DS2 especially in solving DWTA problems of larger scales, which demonstrates the efficiency of the DS based on random restart from current best solutions. In contrast to TS-DS2, HGA and HACO produced better solutions, but both of them have a large time cost. The convergence curves indicate that TS-DS1 also has obvious superiority over LS-MtH&RR, HGA, HACO, and TS-DS2 in the aspect of convergence speed. Besides, TS-DS2 outperforms LS-MtH&RR in solving the seven instances of different scales, which further proves the effectiveness of tabu mechanism.

When solving the instance No. 7 with 200 targets and 200 weapons, even the time cost of the best algorithm (TS-DS1), as much as about 8 min, is usually unacceptable for practical real-time decision making. However, TS-DS1 and TS-DS2 can solve the DWTA instances from No. 1 to No. 4 within a few seconds. Note that there are 100 weapons, 50 targets, and 4 stages in the instance No. 3, which corresponds to a representative DWTA scenario of large scale in practice. In the instances No. 5 and No. 6, TS-DS1 and TS-DS2 took a few minutes to make a final decision. The time cost may be unbearable in practice, but it can be reduced by the use of more advanced computing platforms. On the whole, the TS methods proposed in this paper, especially the variant TS-DS1, are competent for DWTA decision makings.

E. Discussion

For search-based DWTA algorithms, such as TS heuristics, GAs, and ACOs, the tradeoff between intensification and diversification is a key point in determining their global optimization performance. The DWTA algorithms involved in this paper adopt different paradigms to achieve the exploitation of accumulated information during search (intensification). For example, the TS methods employ LS for intensification. In addition to LS, HGA and HACO also utilize crossover operation and pheromone exchange, respectively. It is expected that these exploitative operations will benefit the identification and preservation of good building blocks for high-quality solutions. On the implementation of intensification, one significant issue is how to avoid revisiting local optima and discover worse solutions. Obviously, LS-MtH&RR, HGA, and HACO do not include any mechanism to address this problem. Thus, it is rational to anticipate the relative reduction of search efficiency about these algorithms. No doubt that one of the most important factors contributing to the success of TS-DS1 is the utilization of tabu mechanism. Another important issue about intensification is the time complexity of intensification operations. As indicated by the experimental results, the operations of HGA and HACO are very time-consuming, though HGA and HACO have some superiority over TS-DS2 in many cases. Novel operators with convenient implementation and low time complexity are worth further research.

On diversification strategies, the superiority of TS-DS1 over TS-DS2 indicates that the strategy of random restart from current best solutions employed by TS-DS1 is more efficient. The random permutation strategy employed by TS-DS2, which is of pure random restart type, completely discards accumulated information in the process of diversification. In contrast, the random restart in TS-DS1 favors to some extent the inheritance of potentially useful information contained in current best solutions. This is the main difference between the diversification strategies, respectively, employed by TS-DS1 and TS-DS2. Therefore, the DS with a better tradeoff between the exploration of unexplored problem space and the exploitation of accumulated information is another important factor for the success of TS-DS1.

Last but not least, the exploitation of problem characteristics should be integrated into the design of search-based DWTA algorithms. Problem-specific knowledge is a valuable aid to guide search process and reduce problem difficulty. In this paper, we used some knowledge on the DWTA problem in the process of generating initial solutions. It will be a promising attempt to further incorporate the knowledge into the whole search process.

V. Conclusion

TS-based DWTA algorithms were proposed to solve a generic asset-based DWTA problem incorporating capability constraints, strategy constraints, resource constraints, and engagement feasibility constraints. The combination of a VP-based representation and a CP ensures the satisfaction of all constraints. This technique of constraint handling without any specific repair operators or penalty functions enhances the efficiency of generating feasible solutions. A greedy construction heuristic is utilized to provide good initial solutions for the TS methods. Two diversification strategies are employed and experimental results show that the strategy based on random restart from current best solutions can lead to better DWTA decisions. The proposed algorithm TS-DS1 outperforms an LS method (LS-MtH&RR), an HGA, a hybrid ant-colony algorithm, and another TS variant (TS-DS2) in DWTA test cases of different scales. It can produce high-quality DWTA decisions with desirable computation costs in solving large-scale DWTA instances, like those including 100 weapons, 100 targets, and 4 defense stages.

Acknowledgment

The authors would like to thank the Editor-in-Chief, Associate Editor, and the anonymous referees for their constructive suggestions and comments on the earlier versions of this paper.

References

[1] H. Cai, J. Liu, Y. Chen, and H. Wang, "Survey of the research on dynamic weapon-target assignment problem," *J. Syst. Eng. Electron.*, vol. 17, no. 3, pp. 559–565, 2006.

[2] J. N. Roux and J. H. van Vuuren, "Threat evaluation and weapon assignment decision support: A review of the state-of-the-art," *ORiON: J. ORSSA*, vol. 23, no. 2, pp. 151–187, 2007.

[3] M. Athans, "Command and control (C2) theory: A challenge to control science," *IEEE Trans. Autom. Contr.*, vol. AC-32, no. 4, pp. 286–293, Apr. 1987.

[4] E. Çetin and S. T. Esen, "A weapon-target assignment approach to media allocation," *Appl. Math. Comput.*, vol. 175, pp. 1266–1275, 2006.

[5] S. P. Lloyd and H. S. Witsenhausen, "Weapon allocation is NP-complete," in *Proc. IEEE Summer Simul. Conf.*, Reno, NV, 1986, pp. 1054–1058.

[6] P. A. Hosein and M. Athans, "Preferential defense strategies. Part I: The static case," MIT Laboratory for Information and Decision Systems with partial support, Cambridge, MA, Tech. Rep. LIPS-P-2002, 1990.

[7] P. A. Hosein and M. Athans, "Preferential defense strategies. Part II: The dynamic case," MIT Laboratory for Information and Decision Systems with partial support, Cambridge, MA, Tech. Rep. LIPS-P-2003, 1990.

[8] J. Li, R. Cong, and J. Xiong, "Dynamic WTA optimization model of air defense operation of warships' formation," *J. Syst. Eng. Elec.*, vol. 7, no. 1, pp. 126–131, 2006.

[9] A. Malhotra and R. K. Jain, "Genetic algorithm for optimal weapon allocation in multilayer defence scenario," *Defence Sci. J.*, vol. 51, no. 3, pp. 285–293, 2001.

[10] S. Bisht, "Hybrid genetic-simulated annealing algorithm for optimal weapon allocation in multilayer defence scenario," *Defence Sci. J.*, vol. 54, no. 3, pp. 395–405, 2004.

[11] O. Karasakal, "Air defense missile-target allocation models for a naval task group," *Comput. Oper. Res.*, vol. 35, pp. 1759–1770, 2008.

[12] E. Wacholder, "A neural network-based optimization algorithm for the static weapon-target assignment problem," *ORSA J. Comput.*, vol. 1, no. 4, pp. 232–246, 1989.

[13] K. E. Grant, "Optimal resource allocation using genetic algorithms," *Naval Review*, Naval Research Laboratory, Washington, DC, pp. 174–175, 1993.

[14] H. Lu, H. Zhang, X. Zhang, and R. Han, "An improved genetic algorithm for target assignment optimization of naval fleet air defense," in *Proc. 6th World Cong. Intell. Contr. Autom.*, Dalian, China, 2006, pp. 3401–3405.

[15] A. C. Cullenbine, "A taboo search approach to the weapon assignment model," Master thesis, Department of Operational Sciences, Air Force Institute of Technology, Hobson Way, WPAFB, OH, 2000.

[16] H. R. Li and Y. Miao, "WTA with the maximum kill probability based on simulated annealing algorithms," in *Proc. Conf. Spec. Committee C2 Comput. Electron. Technol. Acad. Committee China,* Ship Engineering Society, 2000, pp. 436–440.

[17] Z. J. Lee, C. Y. Lee, and S. F. Su, "An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem," *Appl. Soft Comput.*, vol. 2, pp. 39–47, 2002.

[18] X. Zeng, Y. Zhu, L. Nan, K. Hu, B. Niu, and X. He, "Solving weapon-target assignment problem using discrete particle swarm optimization," in *Proc 6th World Congr. Intell. Contr. Autom.*. Dalian, China, 2006, pp. 3562–3565.

[19] Z. J. Lee, S. F. Su, and C. Y. Lee, "Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 1, pp. 113–121, Feb. 2003.

[20] T. Fu, Y. Liu, and J. Chen, "Improved genetic & ant colony optimization for regional air defense WTA problem," in *Proc 1st Inter. Conf. Innov. Comput., Info., Contr.*. Dalian, China, 2006, pp. 226–229.

[21] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, "Exact and heuristic algorithms for the weapon-target assignment problem," *Oper. Res.*, vol. 55, no. 6, pp. 1136–1146, 2007.

[22] O. Kwon, K. Lee, D. Kang, and S. Park, "A branch-and-price algorithm for a targeting problem," *Naval Res. Log.*, vol. 54, pp. 732–741, 2007.

[23] P. A. Hosein, J. T. Walton, and M. Athans, "Dynamic weapon-target assignment problems with vulnerable C2 nodes," MIT Laboratory for Information and Decision Systems with partial support, Cambridge, MA, Tech. Rep. LIDS-P-1786, 1988.

[24] D. Blodgett, M. Gendreau, F. Guertin, and J. Y. Potvin, "A tabu search heuristic for resource management in naval warfare," *J. Heur.*, vol. 9, pp. 145–169, 2003.

[25] P. A. Hosein and M. Athans, "Some analytical results for the dynamic weapon-target allocation problem," MIT Laboratory for Information and Decision Systems with partial support, Cambridge, MA, Tech. Rep. LIDS-P-1944, 1990.

[26] D. Khosla, "Hybrid genetic approach for the dynamic weapon-target allocation problem," in *Proc. SPIE*, vol. 4396, pp. 244–259, 2001.

[27] M. E. Havens, "Dynamic allocation of fires and sensors," Master thesis, Naval Postgraduate School, Monterey, CA, OMB No. 0704–0188, 2002.

[28] B. J. Zacherl, "Weapon-target pairing: Revising an air tasking order in real-time," Master thesis, Naval postgraduate School, Monterey, CA, 2006.

[29] L. Wu, C. Xing, F. Lu, and P. Jia, "An anytime algorithm applied to dynamic weapon-target allocation problem with decreasing weapons and targets," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, China, 2008, pp. 3755–3759.

[30] T. Sikanen (2008). Solving weapon target assignment problem with dynamic programming [Online]. Tech. Rep. 55670 A TFY. N Available: http://www.sal.tkk.fi/Opinnot/Mat-2.108/pdf-files/esik08b.pdf.

[31] D. Dionne, E. Pogossian, A. Grigoryan, J. Couture, and E. Shahbazian, "An optimal sequential optimization approach in application to dynamic weapon allocation in naval warfare," in *Proc. 11th Inter. Conf. Info. Fusion*, Colgone, Germany, 2008, pp. 1–6.

[32] F. Glover, "Tabu search: Part I," *ORSA J. Comput.*, vol. 1, pp. 190–206, 1989.

[33] M. Gendreau and J. Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Oper. Res.*, vol. 140, pp. 189–213, 2005.

[34] T. James, C. Rego, and F. Glover, "Multistart tabu search and diversification strategies for the quadratic assignment problem," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 579–596, May 2009.

[35] D. E. Knuth, *The Art of Computer Programming*, 2nd ed., Vol. 3, *Sorting and Searching, Reading*. MA: Addison-Wesley, 1998.

[36] Z. X. Guo, W. K. Wong, S. Y. S. Leung, J. T. Fan, and S. F. Chan, "A genetic-algorithm-based optimization model for solving the flexible assembly line balancing problem with work sharing and workstation revisiting," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 218–228, Mar. 2008.

[37] N. H. W. Eklund, "Using genetic algorithms to estimate confidence intervals for missing spatial data," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 519–523, Jul. 2006.

[38] M. A. Bayir, I. H. Toroslu, and A. Cosar, "Genetic algorithm for the multiple-query optimization problem," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 147–153, Jan. 2007.

[39] W. N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 29–43, Jan. 2009.

[40] L. Wang and C. Singh, "Reliability-constrained optimum placement of reclosers and distributed generators in distribution networks using an ant colony system algorithm," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 6, pp. 757–764, Nov. 2008.

[41] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 337–352, Nov. 2000.

[42] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–487, Oct. 2005.

[43] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of PMSM drives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 28–41, Feb. 2007.

[44] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

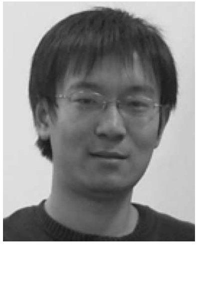
[45] T. Back, "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *Proc. 1st IEEE Conf. Evol. Comput.*, Orlando, FL, 1994, pp. 57–62.

[46] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Reading, MA: MIT Press, 2004.

[47] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1999.

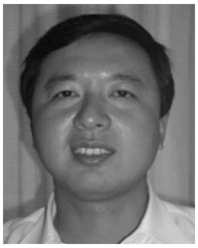
[48] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang, "Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 680–691, May 2009.

[49] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, pp. 617–644, 2009.



Bin Xin (S'09–M'10) received the B.S. degree in information engineering from Beijing Institute of Technology, Beijing, China, in 2004, where he is currently working toward the Ph.D. degree in the School of Automation.

His current research interests include search and optimization, decision makings, evolutionary computation, operations research, combinatorial optimization, and evolutionary games.



Jie Chen (M'09) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Beijing Institute of Technology, Beijing, China, in 1986, 1993, and 2000, respectively.

From 1989 to 1990, he was a Visiting Scholar at California State University. From 1996 to 1997, he was a Research Fellow in the School of Electronic and Electrical Engineering, University of Birmingham, Birmingham, U.K. He is currently a Professor of control science and engineering at Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology. His research interests include complicated system multiobjective optimization and decision, intelligent control, constrained nonlinear control, and optimization methods.

Juan Zhang received the B.S. and Ph.D. degrees in control theory and control engineering from Beijing Institute of Technology, Beijing, China, in 1997 and 2002, respectively.

She is currently an Associate Professor of control science and engineering, Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology. Her research interests include operations research, intelligent decision makings, distributed simulation, and image processing.

Lihua Dou received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Beijing Institute of Technology, Beijing, China, in 1979, 1987, and 2001, respectively.

She is currently a Professor of control science and engineering at Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology. Her research interests include multiobjective optimization and decision, pattern recognition, and image processing.

Zhihong Peng received the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000.

She is currently an Associate Professor at Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology, Beijing, China, where she was a Postdoctoral Research Associate from December 2000 to February 2003. Her research interests include genetic algorithms, intelligent information processing, bioinformatics, and intelligent control.