Hindawi
Complexity
Volume 2019, Article ID 7589760, 18 pages
https://doi.org/10.1155/2019/7589760

Research Article

Weapon Selection and Planning Problems Using MOEA/D with Distance-Based Divided Neighborhoods

Jian Xiong,[1] **Rui Wang**,[2] **and Jiang Jiang**[2]

[1]*School of Business Administration, Southwestern University of Finance and Economics, Chengdu 610074, China*
[2]*College of Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China*

Correspondence should be addressed to Jian Xiong; xiongjian2017@swufe.edu.cn

Received 25 May 2019; Revised 16 October 2019; Accepted 25 October 2019; Published 22 November 2019

Academic Editor: Alex Alexandridis

Real-world multiobjective optimization problems are characterized by multiple types of decision variables. In this paper, we address weapon selection and planning problems (WSPPs), which include decision variables of weapon-type selection and weapon amount determination. Large solution space and discontinuous, nonconvex Pareto front increase the difficulty of problem solving. This paper solves the addressed problem by means of a multiobjective evolutionary algorithm based on decomposition (MOEA/D). Two mechanisms are designed for the complex combinatorial characteristic of WSPPs. The first is that the neighborhood of each individual is divided as selection and replacement neighborhoods. The second is that the neighborhood size is changing during the evolution by introducing a distance parameter to constrain the search scope of each subproblem. The proposed algorithm is termed as MOEA/D with distance-based divided neighborhoods (MOEA/D-DDNs) which can overcome possible drawbacks of original MOEA/D with weighted sum approach for complex combinatorial problems. Benchmark instances are generated to verify the proposed approach. Experimental results suggest the effectiveness of the proposed algorithm.

1. Introduction

Portfolio optimization problems and project planning problems are two classic topics in the areas of operational research and management science. Many optimization problems in industries, economics, and production can be modeled as portfolio problems or planning and scheduling problems. For instance, in defense planning areas, most countries have paid much attention to capability-based planning (CBP) from the beginning of this century [1]. CBP usually involves the development of new types of weapons for future capability fulfilment. Limited by budgetary, decision makers need to balance among different objectives such as cost, efficiency, and risk [2–4].

During the process of CBP, decision makers need to decide on both "what to buy" and "when to buy," which correspond to the concepts of portfolio and project planning. However, in the existing literature, portfolio problems and project planning problems are two relatively independent branches. The former focuses on the item selection among available set, while the latter concerns activity arrangement with a time horizon. Simultaneous consideration of these two types of optimization processes is rather scant in the literature. In a previous research, the basic optimization process of weapon development in CBP is modeled as a weapon selection and planning problem (WSPP) [5].

In this paper, we refine the WSPP model and concern the design of effective multiobjective evolutionary algorithms to solve the problem. The main contributions of this paper are twofold:

(1) This paper refines the multiobjective optimization model of WSPPs. One main trait of the WSPP is that each weapon equipment has its own service period. This is a main difference between WSPPs and other planning and scheduling problems. The previous work did not consider this attribute [5]. Weapon equipment has effectiveness only in their service

periods. The calculation of solution effectiveness needs to consider the beginning point and the end point of each unit of weapon equipment. Therefore, the solution evaluation becomes more expensive. Thus, incorporating the attribute of the service period, fundamentally, changes the structure of the problem and increases the problem complexity. In the proposed model, simultaneous consideration of solution effectiveness and risk provides decision makers with important trade-offs. To illustrate the addressed problem, we generate three sets of test problems including 15 instances with different scales.

(2) The decision space of the WSPP is discrete, and the Pareto front is discontinuous and nonconvex. Thus, multiobjective evolutionary algorithms (MOEAs) are preferred to solve the problem. Although classic MOEAs are reported and tested on many benchmarks, the performances of MOEAs for solving complex combinatorial problems are still open questions. We solve the problem in the framework of the multiobjective evolutionary algorithm based on decomposition (MOEA/D). In the MOEA/D, the weighted sum approach is simple and effective for convex problems, but the performance is limited when Pareto fronts are nonconvex and/or disconnected. Furthermore, components of original MOEA/D, such as neighborhood definition, selection, and replacement, might be problematic for combinatorial problems. To address these issues, a mechanism of distance-based divided neighborhood (DDN) is designed and incorporated into the MOEA/D. The proposed algorithm is termed as MOEA/D-DDN. Experimental results show that the DDN mechanism is effective and MOEA/D-DDN performs well for solving WSPPs.

The rest of the paper is organized as follows. In Section 2, we briefly review the literature on portfolio problems and capability planning problems. Problem formulation and proposed MOEA/D-DDN are described in Sections 3 and 4, respectively. Section 5 presents the genetic operators for WSPPs. Experimental results are reported in Section 6. Finally, some conclusions are drawn in Section 7.

2. Review of the Related Literature

Since the addressed WSPP is an amalgamation of portfolio and planning problems, we briefly review related works on these two areas in this section.

2.1. Portfolio Optimization with Evolutionary Computation. Portfolio problems inherently have multiobjective traits. Due to nonconvex constraints in practical cases, metaheuristic algorithms such as evolutionary computation (EC) are preferred to solve portfolio problems. Arnone et al. [6] was among the first attempts of applying evolutionary algorithms to solve the problem. The authors suggested the possibility of applying the genetic approach and implemented some of the ideas. Maringer and Kellerer [7] optimized cardinality constrained portfolios by designing an algorithm combining simulated annealing and evolutionary strategies. Cura [8] presented a particle swarm optimization approach to solve the portfolio optimization problem and compared the results with those of other metaheuristic approaches. Although the above works applied different algorithms to portfolio problems, one characteristic shared is that the problem was converted as the single objective optimization problem by a weighted sum approach.

As the development of MOEAs, last two decades have witnessed the growth of taking advantage of MOEAs for solving portfolio problems. Doerner et al. [9] studied project portfolio problems with objectives of benefit and remaining resources. The authors presented a two-phase procedure for the solution approach, where in the first phase, Pareto ant colony optimization was used to determine the solution space of all efficient portfolios. Branke et al. [10] solved portfolio problems with nonconvex constraints by integrating an active set algorithm into a MOEA, termed as envelope-based MOEA. Pai and Michel [11] first employed a k-means cluster analysis to eliminate the cardinality constraint and then designed an algorithm with evolution strategy to solve a subclass of portfolio optimization problems. Gutjahr et al. [12] and Kremmel et al. [13] addressed project portfolio problems. In Gutjahr et al. [12], economic and staff competence benefits are simultaneously optimized. The overall problem is decomposed into a master problem and a slave problem. The authors, respectively, implemented NSGA-II [14] and Pareto ant colony optimization to solve the subproblems. Kremmel et al. [13] focused on software project portfolio problems, and a multiobjective optimization approach based on the Prototype Optimization with Evolved iMprovement Steps (mPOEMS) was used to solve the problem. In Anagnostopoulos and Mamanis's study [15], performances of different MOEAs for portfolio problems were compared. Experimental results suggest that performances of the Strength Pareto Evolutionary Algorithm 1 (SPEA1) and the Pareto Envelope-based Selection Algorithm (PESA) are comparable and better than that of NSGA-II. In a later research of the same authors [16], performances of five MOEAs for cardinality constrained portfolio optimization problems were investigated. For recent comprehensive surveys of solving portfolio problems by MOEAs, readers are referred to Metaxiotis and Liagkouras [17] and Ponsich et al. [18].

Compared to other areas such as financial and project management, there are seldom applications of portfolio optimization in the defense area. Greiner et al. [19] studied the screening of weapon system development projects and their realistic application in the air force. In that research, the authors presented a decision support methodology integrating an analytic hierarchy process (AHP) and a 0-1 integer portfolio optimization model. Yang et al. [20] employed heuristic algorithms to deal with portfolio selection problems for military investment assets. Kangaspunta et al. [4] studied a weapon system portfolio problem

```
$\delta_n^i$: original budget share index at each time unit
$A_n$: actual available budget each year
$S_n$: actual spent capital in each year
$B_n^i$: available budget at each time unit
$\widehat{B}_n^i$: spent budget at each time unit
$\overline{B}_n^i$: accumulated budget, $\widehat{B}_n^1 = 0$
for $n = 1; n \leq N; n++$ do
    $S_n = 0$
    for $i = 1; i \leq 12; i++$ do
        $B_n^i = \overline{B}_n^i + S_n \times \delta_n^i$
        if $i > 1$ then
            $B_n^i = \overline{B}_n^i$
        end if
        $\widehat{B}_n^i = 0$
        $U$ is the total number of unselected and unplanned weapon units;
        index = 1;
        listing the unselected weapon units with ascending order by $p_w^{i_w}$;
        while $\widehat{B}_n^i \leq B_n^i$ and index $\leq U$ do
            $p_{w^*}^{i_{w^*}^*}$ is the indexth lowest value;
            index = index + 1;
            if $\widehat{B}_n^i + c_{w^*} \leq B_n^i$ then
                bool $constraintviolation$ = false
                $t_1 = (n-1) \times 12 + i$
                $t_2 = t_1 + d_{w^*}$
                for $t = t_1; t \leq t_2; t++$ do
                    if $a_{w^*\_t}^d = \overline{a}_{w^*}^d$ then
                        $constraintviolation$ = true
                        break;
                    end if
                end for
                if $constraintviolation$ = false then
                    weapon unit with $p_{w^*}^{i_{w^*}^*}$ is selected and planned;
                    $\widehat{B}_n^i = \widehat{B}_n^i + c_{w^*}$
                end if
            else
                $\overline{B}_n^i = B_n^i - \widehat{B}_n^i$
                break;
            end if
        end while
    end for
end for
```

Algorithm 1: Decoding procedure.

in which a weapon portfolio is evaluated by an additive value function.

2.2. Planning and Scheduling in Defense Area. In defense areas, decisions on weapon acquisition given a time horizon can be referred to long-term planning problems. Abbass et al. [21] studied resource planning under time constraint problems in which the main task is optimizing the mix of vehicles to fulfil future tasks. The authors developed an evolutionary multiobjective approach to obtain efficient solutions. Bui et al. [22] considered the military capability planning problem as a combination of resource-constrained project scheduling and resource investment problems. The planning process was modeled as a multiobjective problem, and an evolutionary algorithm was designed to address the optimization task. In the planning of weapon systems, the effectiveness of a planning solution is usually affected by weapons hold by the counterpart. Golany et al. [23] focused on the problem of developing effective countermeasures given limited resources. Usually, weapon acquisition needs to simultaneously consider selection and planning processes to obtain overall optimal solutions. However, in project planning and scheduling areas, the literature simultaneously addressing the combination of these processes is rather scant, except for Ghasemzadeh et al. [24], Sun and Ma [25], Gutjahr et al. [12], and Carazo et al. [26]. In these works, decision makers are only required to determine whether a project will be selected or not. The problem addressed in this paper differentiates from models of the existing work by considering the amount of different weapon types at each decision point. Since weapon developments are subject to resource constraint and the effectiveness of a solution should be measured by considering all selected weapons, the problem needs to be solved at the overall level. In the weapon selection process, simultaneous consideration of the selected type and corresponding amount expands the decision space and dramatically increases the complexity of the problem. Thus, it is worthwhile to design more effective algorithms to solve WSPPs.

3. Weapon Selection and Planning Problems

3.1. Problem Description. A WSPP can be simply described as the process that decision makers select weapons and plan their developments at each decision time point. The planning horizon is divided into M units and is indexed by $m\,(m = 1, \ldots, M)$. Each unit indicates one month [12]. There are W types of weapons, and each weapon type w has the following attributes:

(i) $a_w^{\max}$: the maximum available amount of each weapon type

(ii) c_w: the cost for one unit of each weapon type

(iii) v_w: the value for one unit of each weapon type

(iv) h_w: the risk for one unit of each weapon type

(v) d_w: the period of the weapon w from the beginning of the weapon development to the deployment of the weapon

(vi) s_w: the service period of each weapon type

Similar to revenue in project portfolio [13], initial values corresponding to each unit of weapon are pregiven. However, in our research, v_w can be interpreted as the damage rate per unit time inflicted by the weapon w on counterparts [23]. Without loss of generality, we assume that initial weapon unit value and risk are normalized so that $v_w \in (0, 1]$ and $h_w \in (0, 1]$. We assume that military departments pay all costs to the contractors at the beginning of the developments. At each time point, decision makers need to decide the selected amount for each weapon type, denoted as a_w^m. Then, the total selected amount for each weapon type can be calculated as $a_w = \sum_{m=1}^{M} a_w^m$, and the selected weapon units for each type is indexed by w^* from 1 to a_w. For a weapon type w, the development starting time of a selected weapon unit w^* is denoted as $t_w^{w^*}$; for instance, t_1^2 indicates the development starting time of the unit 2 of the weapon type 1.

At each time point, there are two entities need to be addressed for each weapon type:

(i) u_w^m: the amount of weapon w under development at time point m

(ii) o_w^m: the amount of weapon w in operation at time point m

Note that the values of u_w^m and o_w^m are determined by $a_w^{m_1}$ $(m_1 = 1, 2, \ldots, m)$. For example, we consider a planning with a horizon $M = 100$. Let $w = 1$ and $d_1 = 12$ and $s_1 = 60$. We suppose that the total selected amount of this weapon type is $a_1 = 20$. The weapon units are developed at the first and the fifth time points $a_1^1 = 5$ and $a_1^5 = 15$. Then, the values of u_1^m and o_1^m can be obtained as follows:

$$
\begin{aligned}
u_1^m &= \begin{cases} 5, & m \le 5, \\ 20, & 5 < m \le 12, \\ 15, & 12 < m \le 17, \\ 0, & \text{else}, \end{cases} \\
o_1^m &= \begin{cases} 5, & 12 < m \le 17, \\ 20, & 17 < m \le 72, \\ 15, & 72 < m \le 77, \\ 0, & \text{else}. \end{cases}
\end{aligned}
\tag{1}
$$

The WSPP subjects to the following constraints.

3.1.1. Maximum Budget for Each Year. Budget for weapon development provided by the government for each year is restricted and denoted as B_n. The capital is available at the beginning of each year. The actual spent capital in each year is denoted as S_n. Similar to Carazo et al. [26], we assume that the capital not spent in previous years can be accumulated and transferred to the next year. The actual available capital for each year is A_n. Then, there are $A_1 = B_1$ and $A_n = B_n + (A_{n-1} - S_{n-1})$.

3.1.2. Minimum Amount in Operation. For a solution of weapon planning, some types of weapons cannot form combat effectiveness unless they have a considerably large scale. Specifically, we use $\underline{o}_w$ to denote the minimum amount of weapon w in operation to form combat effectiveness at each time point. If $o_w^m < \underline{o}_w$, then the effectiveness value of weapon w at time point m is 0. This constraint is similar to a *buy-in threshold* [10, 27] constraint in portfolio management problems.

3.1.3. Maximum Amount under Development. The maximum amount of a weapon type under development is bounded because of limitation of manufacturing capability of contractors. We use $\overline{u}_w$ to indicate the maximum amount of weapon units under development, and there is $u_w^m \le \overline{u}_w$.

3.2. Objective Calculation. In this paper, we focus on tradeoff between solution effectiveness and risk.

3.2.1. Calculation of Effectiveness. For a solution of the WSPP, effectiveness needs to be evaluated at the overall level during the whole planning horizon. Usually, defense agencies intend to maximize damage rates of developed weapons [23]. Similar to weapon portfolios, the overall effectiveness can be expressed with an additive value function [4]. However, a simple additive value does not consider effectiveness balance during the whole planning process. In practice, decision makers may prefer a weapon plan which can maintain a high and stable effectiveness during the whole planning horizon. Thus, in this paper, we measure the effectiveness of a weapon plan by the average effectiveness per time unit as well as the deviation of effectiveness during the whole planning.

Furthermore, we assume that the value v_w is decreasing in time unit m. This assumption is reasonable in practice because that the effectiveness of a weapon will decrease due to performance degrade or occurrence of countermeasures. We use the following formulation to capture this characteristic:

$$v_w^m = v_w\left(\alpha + (1-\alpha)\frac{M-m+1}{M}\right), \tag{2}$$

where $\alpha \in [0,1]$ is a coefficient to indicate the minimum unit effectiveness during the planning horizon. We set $\alpha = 0.5$ in this research for the sake of calculation.

Then, at each time unit m, the effectiveness $\mathrm{EF}(m)$ is taken as the additive effectiveness value of all weapon units in operation:

$$\mathrm{EF}(m) = \sum_{w=1}^{W} x_w^m o_w^m v_w^m, \tag{3}$$

where $x_w^m = 1$ if $o_w^m \geq \underline{o}_w$; otherwise, $x_w^m = 0$. The overall effectiveness of a WSPP solution, denoted as *EF*, can be obtained by a mean-variance model:

$$\mathrm{EF} = \mathrm{mean}(\mathrm{EF}(m)) - \beta \mathrm{StD}(\mathrm{EF}(m)), \tag{4}$$

where $\mathrm{mean}(\mathrm{EF}(m))$ is the mean value over the whole planning horizon and $\mathrm{StD}(\mathrm{EF}(m))$ is the standard deviation. Similar to Yamashita et al. [28], parameter β is a variance factor that determines the degree of effectiveness unbalance aversion of the decision maker. Note that EF is to be maximized.

3.2.2. Calculation of Risk. The risk corresponding to each weapon development is considered to be a nonincreasing function of development time. Specifically, if a weapon unit is planned to be developed at a later time point, the risk it faces will be lower. This is because technology uncertainties of weapon R&D will be decreased with the development of new technologies in the future. Similarly, the overall risk of a solution can be modeled by an additive value function with the time factor. Here, we use the averaged risk of each selected weapon unit to indicate a solution's risk value:

$$\mathrm{RK} = \frac{\sum_{w=1}^{W}\sum_{w^*=1}^{a_w} h_w \exp\left(-t_w^{w^*}/M\right)}{\sum_{w=1}^{W} a_w}, \tag{5}$$

where $t_w^{w^*}$ is the starting development time of w^*th unit of weapon w.

It is known that a multiobjective problem (MOP) requires the confliction among different objectives. In our problem, the objective confliction mainly derives from the planning of selected weapon units. For example, if a new weapon will be developed, planning the R&D activity in the future can decrease the risk of failure because of possible technology breakthroughs. However, the effectiveness of the weapon will be lower in the future, compared to the weapon operation at the current period.

3.3. Multiobjective Optimization Model for WSPPs. For WSPPs, EF is to be maximized and RK is to be minimized. Without loss of generality, we convert the problem to a *minimization* format by optimizing negative values of EF. The model of multiobjective optimization problem of a WSPP can be summarized as follows.

3.3.1. Decision Variables. For an overall solution, decision variables are a_w^m, $w = 1, 2, \ldots, W$; $m = 1, 2, \ldots, M$. Note that a_w^m is a nonnegative integer value.

3.3.2. Objectives.

$$\begin{aligned} &(1)\min f_1 = -\mathrm{EF}, \\ &(2)\min f_2 = \mathrm{RK}. \end{aligned} \tag{6}$$

Objectives f_1 and f_2, respectively, represent maximization of *effectiveness* and minimization of *risk* for a weapon selection and planning solution.

3.3.3. Constraints.

$$\begin{aligned} &(1) S_n \leq A_n, \quad \forall n, \\ &(2) u_w^m \leq \overline{u}_w. \end{aligned} \tag{7}$$

Constraint (1) is a budget constraint imposed on each year. Constraint (2) reflects the maximum capability of developing each type of weapon.

4. MOEA/D with Distance-Based Divided Neighborhoods

In this research, we solve the problem by multiobjective genetic algorithm based on decomposition (MOEA/D) [29]. MOEA/D decomposes a MOP into a set of subproblems whose objective function can be a weighted linear or nonlinear aggregated function. The optimal solutions of all subproblems constitute the Pareto optimal set of the original MOP. In MOEA/D, all the scalar objective optimization subproblems are simultaneously optimized in a single run. A neighborhood relationship among all the subproblems is defined based on the distances among their weights. It should be noted that various algorithm components of MOEA/D have been investigated for performance improvement [30–32]. There are several approaches to construct a scalar optimization function [32]. For simplicity, this paper uses the weighted sum approach.

In this section, we propose a distance-based divided neighborhoods (DDNs) strategy for MOEA/D.

4.1. Motivations. Firstly, we would like to elaborate our motivation by making the following comments:

(1) A basic assumption behind MOEA/D is that adjacent subproblems should obtain similar optimal solutions. Although this assumption is reasonable for most function optimization problems, there are some exceptions, especially for some combinatorial

problems [33, 34]. If the Pareto front is disconnected or not distributed evenly, parents selected with a constant neighbor size from the neighborhood of a subproblem may have a lower chance to generate promising offsprings. This is because individuals too different from each other are generally less likely than similar individuals to produce fit offspring through mating [35]. The motivation behind the use of nonconstant neighbor size is illustrated by Figure 1.

(2) In the original MOEA/D, parents are randomly selected from the neighborhood of a subproblem. However, parents with different domination ranks vary a good deal with the capability of generating new offspring, contributing to both convergency and diversity. This is considerably precise during the early stage of evolving. Figure 2 illustrates possible limitation of the original selection mechanism. We are aware of that this situation depends on the problem characteristics. However, for most combinatorial problems, the final Pareto optimal solutions are rather sparse. For a population with a relative large size, this situation is not scarce during the evolution process.

(3) Some variants of MOEA/D address the strategy of population replacement. An alternative is that a new solution could replace any other ones in the whole population [34]. In Zhou and Zhang's study [36], a new solution replaces the one that can be improved most among all the subproblems. In Wang et al.'s study [37], neighboring solutions of a suitable subproblem are replaced by a new solution with a better scalar function value. Nevertheless, in two situations, an obtained nondominated solution could be replaced by an inferior solution. The first is that the Pareto front is nonconvex, and the second is that each subproblem has different convergency speeds. Furthermore, when the Pareto front is nonconvex, the weighted sum approach could not work well. This shortcoming could be overcome if the search direction of a subproblem is restricted within a relative narrow scope. The motivation of using such a restriction is illustrated by Figure 3.

4.2. Distance-Based Divided Neighborhood Strategy. Previous research studies suggest that the use of different neighborhoods, i.e., mating neighborhood and replacement neighborhood, can achieve better performance for MOEA/D [37, 38]. Along this avenue, in the proposed distance-based divided neighborhood (DDN) strategy, the original neighborhood (o-neighborhood) of a subproblem is divided into selection neighborhood (s-neighborhood) and replacement neighborhood (r-neighborhood). Furthermore, the size of the original neighborhood is restricted and changed according to a pregiven search scope. The DDN strategy is illustrated in Figure 4.

For a solution x_i corresponding to the subproblem i, the basic DDN strategy works as follows:

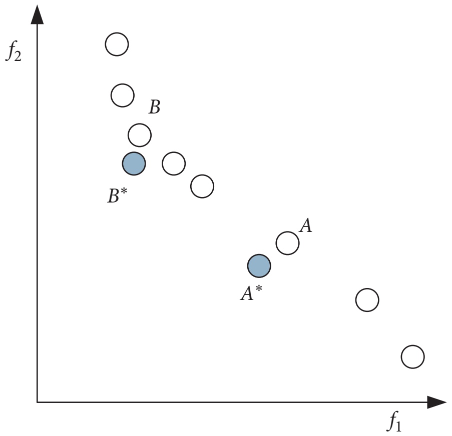


Figure 1: Illustration of *motivation 1*: suppose solutions A and B are two nondominated solutions. The neighborhood size is 5, and the neighborhood consists of 5 closest solutions in objective space. A^* and B^* are two potential dominating solutions. It is very likely that A has a lower chance to generate A^* by randomly selecting two parents in its neighborhood since A's neighbors are far from each other. In this figure, the minimization problem is considered.

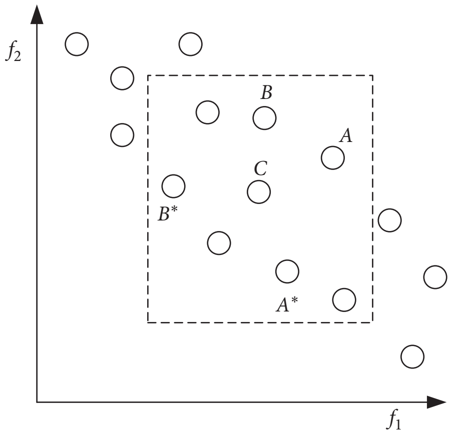


Figure 2: Illustration of *motivation 2*: given a solution C, whose neighbors are in the dash rectangle. It is clear that it has a higher chance to generate a better child solution if A^* and B^* are selected as parents, compared to A and B.

(1) Determines s-neighborhood and r-neighborhood, respectively

(2) *Mating*: select two solutions from the s-neighborhood of x_i and generate a new solution x_i^* by using genetic operators

(3) *Replacement*: update the r-neighborhood of x_i by replacing at most n solutions with worse scalar function values

Before defining s-neighborhood and r-neighborhood, we introduce a parameter, d, to indicate the search scope of a subproblem. We first determine a reference point x_r. For a solution x_i, a line crossing x_r and x_i can be constructed. Note that, for two objectives problems addressed in this research, the line can be alternatively determined by x_i and its associated weight vector. Then, d_{ij} is defined as the distance from

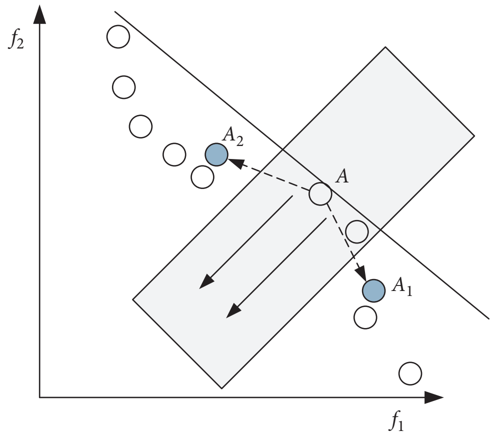


Figure 3: Illustration of *motivation 3*: given a solution A, a child solution might be A_1 or A_2. If weighted sum approach is used, it is possible that A will be replaced by A_1 or A_2. However, A is a nondominated solution, while A_1 and A_2 are both dominated. If we restrict the replacement in a certain scope, denoted by the gray area, A can be maintained in the population.

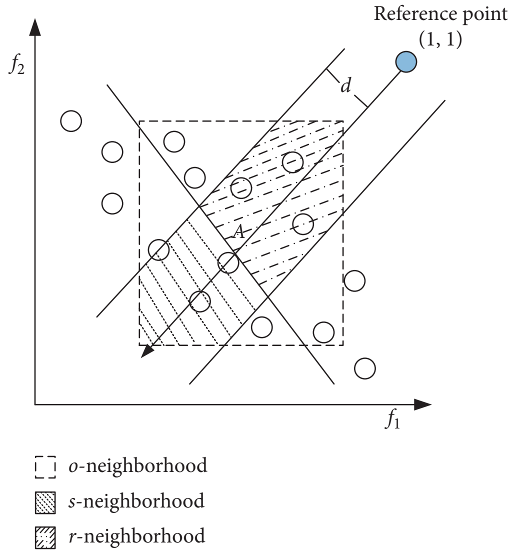


Figure 4: Illustration of the proposed DDN strategy.

another solution x_j to the line. In the calculation of distances d_{ij}, objective values are normalized for each objective vector by utilizing the obtained maximum and minimum values of each objective. Solutions in s-neighborhood and r-neighborhood are, respectively, defined as follows.

Definition 1. Given a solution x_i, a neighbor $x_j \in o-$neighborhood is called a selection neighbor (i.e., $x_j \in s-$neighborhood) if

(1) $d_{ij} \leq d$
(2) $f^{ws}(x_j \mid \lambda^i) \leq f^{ws}(x_i \mid \lambda^i)$

Definition 2. Given a solution x_i, a neighbor $x_j \in o-$neighborhood is called a replacement neighbor (i.e., $x_j \in r-$neighborhood) if

(1) $d_{ij} \leq d$
(2) $f^{ws}(x_j \mid \lambda^i) > f^{ws}(x_i \mid \lambda^i)$.

One can notice from the above two definitions that the original neighborhood is restricted by d and divided according to the scalar function.

Here, we make some remarks on the proposed DDN strategy:

(1) Given a subproblem, the DDN strategy divides its neighborhood into s-neighborhood and r-neighborhood. With the subproblem's weight vector, individuals in s-neighborhood have better values of scalar function than those in r-neighborhood. Parent individuals selected from s-neighborhood will have a good chance to produce better child individuals. Replacing individuals only in r-neighborhood can push the whole population towards the true Pareto front, thus increasing the convergency capability.

(2) The individual numbers in the neighborhood change according to the search scope of each subproblem, controlled by the parameter d. The benefits of such a restriction are twofold: firstly, mating can be restricted among similar parent individuals. Secondly, for subproblems with lower convergency speed, potential good individuals can be maintained in the population since each subproblem is only responsible for a specific narrow search direction. Note that, for an individual in the approximated non-domination set during the search process, when d is approaching 0, generation of child individuals can be considered as a local search process.

(3) The use of parameter d bears some similarity to the strategy of constrained decomposition in MOEA/D (MOEA/D-CD) [39]. In Wang et al.' study [39], a constraint is imposed to each subproblem to restrict the improvement region. Then, for a subproblem, the partial order of two solutions is redefined and the replacement is implemented according to the new dominance relation. However, in MOEA/D-CD, the neighborhood size of each subproblem remains fixed. While, in the proposed DDN strategy, the neighborhood size is controlled by the parameter d. Moreover, selection and replacement happen to different regions of the restricted neighborhood.

(4) In the proposed DDN strategy, s-neighborhood and r-neighborhood are defined according to scalar function values corresponding to the subproblem's weight vector. The s-neighborhood of each solution at least includes itself. In such a case, two parents are identical and only mutation operator works. However, for combinatorial problems, it is almost impossible that all subproblems have the same convergence speed. In other words, during the evolution, it is impossible that all individuals in the population are evenly distributed and located in the same on-dominating front. Thus, the s-neighborhood may contain more than one individual and its size is dependent on the parameter d

as well as the population distribution during different stages of evolution. In the experimental analysis, the trend of the s-neighborhood's size during evolution will be reported.

(5) To overcome the limitation of standard MOEA/D when solving problems with a complex Pareto front, some research studies employ the strategy of adaptive weight vector adjustment [40, 41]. By adjusting weight vectors, the diversity performance of the algorithm can be improved. The proposed DDN strategy and the weight vector adjustment strategy share the same motivation, i.e., tackling complex multiobjective optimization problems. However, DDN uses a different mechanism. DDN strategy focuses on improving the convergency ability of the algorithm by taking into account the property of irregular Pareto front. With a DDN strategy, the convergency ability of the algorithm will be improved by restricting mating parents when individuals locate in the discontinuous regions of the Pareto front. While for the weight vector adjustment strategy, an algorithm obtains better uniformity of solutions by redistributing the weights of subproblems [40].

(6) We are aware that there are several other kinds of scalar function in MOEA/D, such as Tchebycheff approach and penalty-based boundary intersection (PBI) approach. For complex MOPs, Tchebycheff and PBI approach might achieve better performances compared to the weighted sum approach with original framework of MOEA/D. However, the benefit comes with a price. For instance, one has to identify the reference points with the Tchebycheff approach and set the value of the penalty factor with the PBI approach. Determining the optimal parameters is not a trivial task for complex problems. The proposed MOEA/D-DDN provides an alternative for solving complex MOPs by MOEA/D with the simplest scalar function, i.e., weighted sum approach.

4.3. MOEA/D with DDN. The framework of the proposed MOEA/D-DDN is shown as Algorithm 2. The original MOP is decomposed into H scalar objective subproblems.

The algorithm specification of MOEA/D is as follows.

4.3.1. New Solution Generation. There are many ways to generate a trial solution in MOEA/D. In the MOEA/D-DDN, a trial solution is generated by using genetic operators on two randomly selected solutions in s-neighborhood. The genetic operators for the addressed problem will be elaborated in the following section.

4.3.2. Population Replacement. MOEA/D and its variants replace the population according to comparison of scalar objective values of newly generated solution and others in the neighborhood. We use the same replacement mechanism as in Li and Zhang [42]; i.e., at most n_r, solutions in the neighborhood can be replaced. In the MOEA/D-DDN, replacement is restricted in the r-neighborhood of each subproblem.

4.3.3. Summary of MOEA/D-DDN. Based on the above discussion, the inputs of MOEA/D-DDN can be summarized as follows:

(i) H: the number of subproblems.

(ii) D: the initial size of neighborhood.

(iii) $C^i \subset \{1, \ldots, H\}$, the index set of the original neighboring subproblems of subproblem i, where $C^i = D$.

(iv) d: the distance of search scope.

(v) n_r: the maximal number of solutions replaced by each child solution.

(vi) $(y_1^*, \ldots, y_k^*)$: the set of reference points in objective space, which can be determined by a problem-specific method. Note that $y_k^* = (y_k^{*-\min}, y_k^{*-\max})$.

(vii) G: the number of generation as a stopping criterion.

(viii) At each generation, MOEA/D-DDN with weighted sum approach maintains.

(ix) A population of H individuals $x_i, \ldots, x_H$.

(x) $\mathrm{FV}_1, \ldots, \mathrm{FV}_H$, where FV_i is the fitness value of x_i.

(xi) An external population (EP), which is used to store nondominated solutions found during the search.

One can see from the above discussion that MOEA/D-DDN has only one additional parameter, i.e., search scope distance d. The performances of the algorithm with different values of d will be investigated in Experimental Analysis. Since the proposed MOEA/D-DDN does not change the basic framework of MOEA/D, the computational complexity of MOEA/D-DDN remains the same as original MOEA/D.

5. Genetic Representation and Operators for WSPPs

5.1. Chromosome Representation. In portfolio problems, a hybrid binary/real-valued encoding is recommended as genetic representation by Streichert et al. [43] and also employed in other works [10, 16]. With this encoding, the real-valued vector is used to indicate the share of the budget on different assets or projects, while the binary vector is used to indicate whether the asset is selected or not. For WSPPs, a chromosome representation should include the information of weapon type, amount, and planning time. However, the hybrid representation for portfolio problems does not consider the factor of time horizon. For the problem addressed in this research, we design a chromosome representation consisting of two parts: *budget distribution* and *priority matrix*, respectively, shown as in Figures 5 and 6.

Both *budget distribution* and *priority matrix* are real-valued. The vector $((\delta_1^1, \delta_1^2, \ldots, \delta_1^{12}), \ldots, (\delta_N^1, \delta_N^2, \ldots, \delta_N^{12}))$

```
Algorithm initialization.
while not terminate do
    for $i = 1$ to $H$ do
        Generate a trial solution for subproblem $i$ using DDN strategy.
        Update the population by the new generated solution using DDN strategy.
        Update the external population.
    end for
end while
```

ALGORITHM 2: MOEA/D-DDN framework.

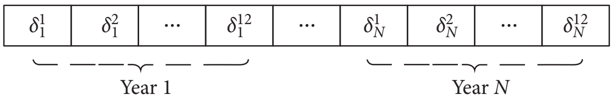


FIGURE 5: Chromosome representation of budget distribution.

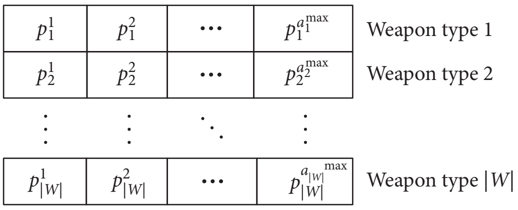


FIGURE 6: Chromosome representation of priority matrix.

indicates budget shares at each time unit for each year. Elements in the priority matrix, $p_w^{i_w}$, where $i_w = 1, \ldots, a_w^{\max}$; $w = 1, \ldots, |W|$, are encoded by real values between interval $[0, 1]$. A lower value indicates a higher priority. In the initialization, the values of $p_w^{i_w}$ are randomly generated in the interval $[0, 1]$, and all values are nonredundant. For example, given a single type of weapon with 6 units, a chromosome can be generated and represented as $(0.554, 0.034, 0.336, 0.498, 0.174, 0.689)$. Then, the second unit of weapon will be selected first if the budget is available.

5.2. Decoding Procedure. A chromosome representation should be decoded to a selection and planning solution to evaluate objective values. At each time unit, the original available budget is determined by the share value δ_n^i, where $i = 1, 2, \ldots, 12$ and $n = 1, 2, \ldots, N$. Note that actual available budget can be different due to the accumulation of previous not-spent budget. The selected weapon type and amount are determined according to priorities. The procedure of decoding is presented as Algorithm 1.

Note that the decoding procedure can ensure the satisfaction of constraints (1) and (2) in the multiobjective optimization problem shown in equation (7). We use a simple example to illustrate the decoding procedure. We consider a problem with 12 planning units (1 year) and a single type of weapon with 6 units. The cost for each weapon unit is $c_1 = 6$, and the available budget is $B_1 = 25$. The chromosome of budget distribution is $(0.4, 0.6, 0, 0, 0, 0, 0\ 0, 0, 0, 0)$, and the priority matrix is $(0.554, 0.034, 0.336, 0.498, 0.174, 0.689)$. Then, there are $B_1^1 = 10$ and $B_1^2 = 15$. At the first time point, due to the constraint of available budget, only one unit of weapon is selected. Specifically, the second unit of weapon is selected. Then, we have $\widehat{B}_1^1 = 6$ and $\overline{B}_1^1 = 10 - 6 = 4$. At the second time point, the available budget is $B_1^2 = \overline{B}_1^1 + B_1^2 = 4 + 15 = 19$. Then, 3 units of weapon can be selected, i.e., weapon unit 5, 3, and 4. At the second time point, $\widehat{B}_1^2 = 18$ and $\overline{B}_1^2 = 1$. Then, we can obtain value of decision variables $a_1^1 = 1$ and $a_1^2 = 3$.

5.3. Crossover and Mutation. Crossover and mutation are two important operators in genetic algorithms. We use BLX-α crossover as in Streichert et al. [27] since this crossover performed best for portfolio problems. The values of δ_w^n of the two parents are denoted as $\text{parent}_1(\delta_w^n)$ and $\text{parent}_2(\delta_w^n)$, respectively. The BLX-$\alpha$ crossover randomly reinitializes the values of δ_w^n from a extended range $(\max\{0, \text{parent}_{\min}(\delta_w^n) - I\alpha\}, \text{parent}_{\max}(\delta_w^n) + I\alpha)$, where $\text{parent}_{\min}(\delta_w^n) = \min(\text{parent}_1(\delta_w^n), \text{parent}_2(\delta_w^n))$, $\text{parent}_{\max}(\delta_w^n) = \max(\text{parent}_1(\delta_w^n), \text{parent}_2(\delta_w^n))$, and $I = \text{parent}_{\max}(\delta_w^n) - \text{parent}_{\min}(\delta_w^n)$. We set $\alpha = 0.5$ as in Streichert et al. [27]. A similar crossover operator is employed for the priority matrix.

The mutation operator is executed as follows: for the budget distributing list, a mutation point is randomly determined for each year. Then, a random real value between 0 and 1 is generated and the value at the mutation point is replaced. A weapon type is randomly selected and denoted as $\overline{w}$. The elements in the priority matrix corresponding to the selected type, $p_{\overline{w}}^{i_{\overline{w}}}$, $i_{\overline{w}} = 1, 2, \ldots, a_{i_{\overline{w}}}^{\max}$, are replaced by values randomly draw from the distributions $N(p_{\overline{w}}^{i_{\overline{w}}}, 0.1)$. If a new generated value is negative, then we set the value as $1e - 8$. Note that this mutation operator is also similar to that in Streichert et al. [27] in which decision variables are mutated by adding a random Gaussian number with a specific deviation.

6. Experimental Analysis

6.1. Test Instance Generation. Since there is no existing literature which addresses exact WSPPs, no benchmark instances can be utilized to test the proposed algorithm. To concretely illustrate the studied problem and investigate the performance of MOEA/D-DDN, different instances were randomly generated. The test bed consists of three sets of instances with different sizes. Table 1 shows parameters of instance generation. Each set includes 5 instances, and the corresponding values were uniformly drawn from the given

Table 1: Parameters of test instance generation.

	W	N	B_n	c_w	v_w	h_w	d_m	s_w	$\underline{o}_w$	$\overline{u}_w$
Set$_1$	30	20	[180, 220]	[10, 25]	[0.50, 0.90]	[0.10, 0.30]	[15, 35]	[50, 150]	[4, 10]	[1, 5]
Set$_2$	40	20	[220, 260]	[10, 25]	[0.50, 0.90]	[0.10, 0.30]	[15, 35]	[50, 150]	[4, 10]	[2, 7]
Set$_3$	50	20	[260, 300]	[10, 25]	[0.50, 0.90]	[0.10, 0.30]	[15, 35]	[50, 150]	[4, 10]	[3, 9]

interval. The main difference among test sets is the size of weapon types. Note that the values in Table 1 are artificial for the reason of confidentiality. However, the identification of interval values also refers to some real cases of weapon selection and planning. Specifically, all ranges of parameters refer to the data from management department of Chinese Army during 2005–2015. Although the test instances are artificial, the results are generalisable to real situations of weapon selection and planning in the future.

6.2. Parameter Settings. The algorithm was implemented with C#. The population size was set as 100, and the initial neighborhood size was set as 20. The rates for crossover and mutation were fine tuned and identified as 0.95 and 0.9, respectively. The maximal number of solutions replaced by each child solution was 2. The algorithm was terminated after 100 generations. In the experiments, all algorithms were run 20 independent times for each instance. For performance comparison, we conducted statistical significance tests for results for a significance level *IÁ* = 0.05.

6.3. Performance Metrics. Similar to most combinatorial problems, the true Pareto front is unknown. Thus, we use hypervolume and set coverage as performance metrics as in Ke et al. [34]:

(1) Hypervolume indicator (I_H) [44]: let f^* be a point in the objective space which is dominated by any Pareto optimal objective vectors. Then, the value of I_H for an obtained approximation to the Pareto front is the volume of the region which is dominated by the approximation and dominates f^*. The higher the hypervolume value, the better the approximation. Since in WSPPs, two objectives have different dimensions, values are normalized for hypervolume calculation. In the normalization, the maximum and the minimum values for objective f_1 are identified as 0 and −25, and the maximum and the minimum values for f_2 are 0.4 and 0. After normalization, the reference point was set as (1, 1) in the calculation of the hypervolume value. Note that if reference points take the range of the problem into consideration, it is not necessary to normalize the objective values of nondominated solutions [45].

(2) Set coverage (C metric) [44]: let A and B be two approximations to the Pareto front. $C(A, B)$ is defined as the percentage of the solutions in B that are dominated by at least one solution in A. For instance, $C(A, B) = 100\%$ indicates that all solutions in B are dominated by some solutions in A. Note that $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. A higher value of $C(A, B)$ and a lower value of its counterpart $C(B, A)$ means A has a better convergence performance. In the calculation of C metric, duplicate solutions were removed from the obtained nondominated set.

6.4. Sensitivity Analysis. In the proposed MOEA/D-DDN, there is an additional parameter-search scope distance d. We first investigated the performances of different d values. Statistical analyses on hypervolume of obtained Pareto solutions suggest that there is no significant difference for most instances.

Note that hypervolume was normalized with a relative wide range. This may affect the analysis of hypervolume measure. We further investigated the set coverage. We obtained the final nondominated solutions with 20 runs for each value of parameter d. A score $S(d)$ is calculated as follows:

$$S(d_i) = \sum_{j \in \text{all values of } d, j \neq i} C(d_j, d_i). \quad (8)$$

Table 2 reports the score $S(d)$ for all tested values of parameter d on 15 instances. A lower value of $S(d)$ indicates a lower percentage of obtained dominated solutions. In the table, if results with a specific value of d achieve the lowest score, the score value is highlighted with *bold* face. The results in the table also suggest that optimal value of parameter d varies with different instances. For our benchmark, the parameter of $d = 0.4$ performed best for 7 in 15 instances, while the numbers for $d = 0.05, 0.20,$ and 0.60 are 4, 3, and 1, respectively. Then, if the parameter d needs to be identified for all instances, the value $d = 0.40$ is preferred.

6.5. Effect of Algorithm Components. The proposed MOEA/D-DDN is characterized by two mechanisms: (1) dividing neighborhood as selection neighborhood and replacement neighborhood and (2) using parameter d to control the neighborhood size. In this subsection, we investigated the effects of the above two mechanisms. The algorithm only with divided neighborhoods is denoted as MOEA/D-DivN; i.e., the parameter d is relaxed. However, the algorithm including parameter d based on original MOEA/D is denoted as MOEA/D-DisN. We chose the parameter $d = 0.40$ for MOEA/D-DisN and MOEA/D-DDN. Table 3 reports the obtained results of hypervolume for the three algorithm versions, as well as original MOEA/D. For each instance, the best value, the average, and the standard deviation are reported. For the value obtained by MOEA/D-DDN, if it is significant better than one of the other three algorithms, the average is presented in *bold* face. From Table 3 we can see that MOEA/-DDN performed best for all

Table 2: Obtained $S(d)$ with different values of d.

Instance	d			
	0.05	0.20	0.40	0.60
1	**0.5455**	0.8421	2.5909	1.2333
2	2.1764	**0.6296**	1.1219	0.9565
3	1.7436	1.6875	**0.9999**	1.1333
4	2.5806	0.6250	**0.5527**	1.5666
5	1.775	1.3530	**0.6551**	2.1290
6	**1.0518**	1.1428	1.3800	1.3721
7	2.3794	1.1000	**1.0322**	1.6364
8	1.7932	**0.7143**	1.4687	1.7813
9	2.0416	1.4642	**1.0000**	1.2258
10	1.9412	1.4474	**1.0213**	1.0605
11	**1.0256**	1.6190	1.5117	1.7391
12	**0.9375**	1.9487	1.2955	0.9762
12	1.3750	**1.0139**	1.6734	1.0690
14	1.3877	2.0385	**0.9033**	1.1796
15	0.9242	1.1714	2.1000	**0.9143**

Table 3: Performance (hypervolumn) comparison of MOEA/D-DDN, MOEA/D-DivN, MOEA/D-DisN, and MOEA/D.

Instance	MOEA/D-DDN			MOEA/D-DivN			MOEA/D-DisN			MOEA/D		
	Best	Average	Std.	Best	Average	Std.	Best	Average	Std.	Best	Average	Std.
1	0.3609	**0.3397**	0.0171	0.3522	0.3178	0.0216	0.3429	0.3117	0.0149	0.349	0.3095	0.0246
2	0.4033	**0.3541**	0.0198	0.3774	0.3331	0.0201	0.3666	0.3159	0.0256	0.3661	0.3169	0.0239
3	0.3609	**0.3379**	0.0179	0.3402	0.3074	0.0214	0.3642	0.3092	0.0241	0.3674	0.3236	0.0250
4	0.3632	**0.2901**	0.0256	0.2886	0.2620	0.0164	0.3017	0.2539	0.0204	0.3105	0.2585	0.0276
5	0.4188	**0.3528**	0.0267	0.3653	0.3304	0.0185	0.3689	0.3151	0.0285	0.3616	0.3250	0.0193
6	0.3283	**0.2906**	0.0265	0.3348	0.2706	0.0292	0.3457	0.2654	0.0312	0.3325	0.2817	0.0294
7	0.4449	**0.4101**	0.0223	0.4111	0.3601	0.0301	0.4065	0.3405	0.0423	0.3861	0.3422	0.0292
8	0.4088	**0.3572**	0.0258	0.3612	0.3199	0.0289	0.3646	0.3118	0.0319	0.3533	0.3086	0.0276
9	0.4221	**0.3746**	0.0245	0.3828	0.3320	0.0325	0.4064	0.3348	0.0305	0.3996	0.3331	0.0306
10	0.4533	**0.3986**	0.0275	0.4432	0.3460	0.0283	0.3998	0.3460	0.0283	0.3936	0.3521	0.0280
11	0.4681	**0.4180**	0.0217	0.4219	0.3590	0.0317	0.4395	0.3697	0.0377	0.4391	0.3746	0.0264
12	0.4622	**0.4089**	0.0234	0.4314	0.3677	0.0347	0.4406	0.3705	0.0344	0.4216	0.3525	0.0257
13	0.5151	**0.4712**	0.0247	0.4920	0.4299	0.0352	0.4535	0.4154	0.0278	0.4795	0.4270	0.0314
14	0.4617	**0.3909**	0.0372	0.4135	0.3512	0.0384	0.4224	0.3418	0.0481	0.4276	0.3418	0.0440
15	0.4156	**0.3585**	0.0380	0.3714	0.3242	0.0310	0.3778	0.3147	0.0329	0.3793	0.3189	0.0287

instances. However, the performances of MOEA/D-DivN, MOEA/D-DisN, and MOEA/D are comparable. It suggests that the combination of two algorithm components of MOEA/D-DDN is effective for the addressed problem.

6.6. Algorithm Dynamics. Compared to original MOEA/D, the main difference of MOEA/D-DDN lies in the change of divided neighborhoods. The sizes of s-neighborhood and r-neighborhood vary during the evolution process. In this subsection, we investigated the dynamics of neighborhood sizes for individuals. The data of the first run for instance 10 was used for analysis. We first present two different patterns of neighborhood size variation. The 30th and the 96th individuals in the population were identified. The neighborhood sizes are shown as Figures 7 and 8, respectively.

Typically, a lower value of s-neighborhood might indicate a better convergence performance for an individual, and vice versa. One can see from Figure 7 that, for the 30th individual, the trend of s-neighborhood size is decreasing with evolution. At the beginning of the evolution process, the s-neighborhood size was 13, while the value remained 1 when the algorithm ended. This indicates that the 30th individual had a poor performance of nondominated ranking in the initial population. However, this performance spirally improved during the evolution. A change of s-neighborhood size from a higher value to a lower value represents the improvement of nondominated ranking performance. Figure 8 shows a different pattern. The 96th individual had a better nondominated rank in the initial population, while the performance was decreasing along with the evolution. Figure 9 shows the population at the 1st, 50th, and 100th generation, denoted as dot, triangle, and star, respectively. The positions of the 30th and 96th individuals are, respectively, marked by hexagon and circle. The 96th individual was positioned at the first nondominated rank in the initial population. However, it was dominated in the 50th and 100th generations. However, the situation of the 30th individual is opposite.

Here, a question why good solution in the initial population did not contribute to the convergence performance

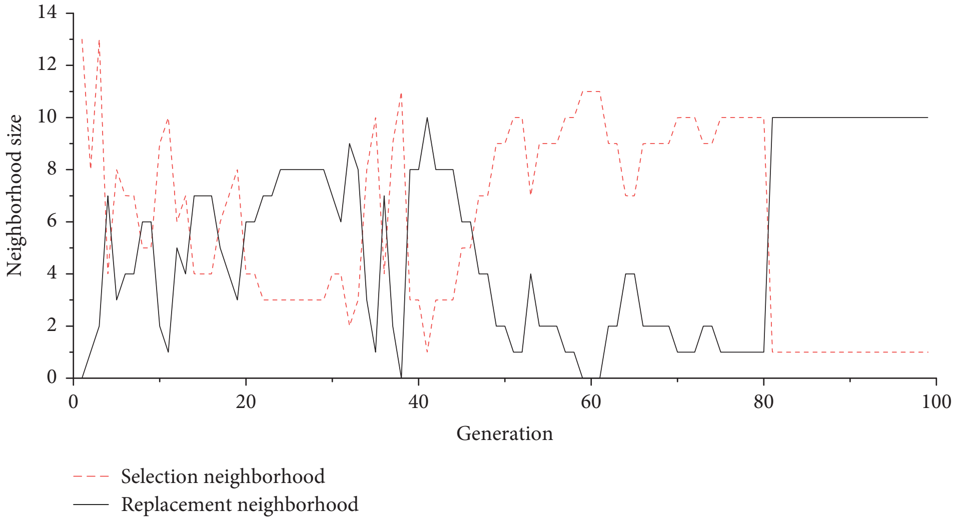


Figure 7: Variation of neighborhood sizes of the 30th individual.

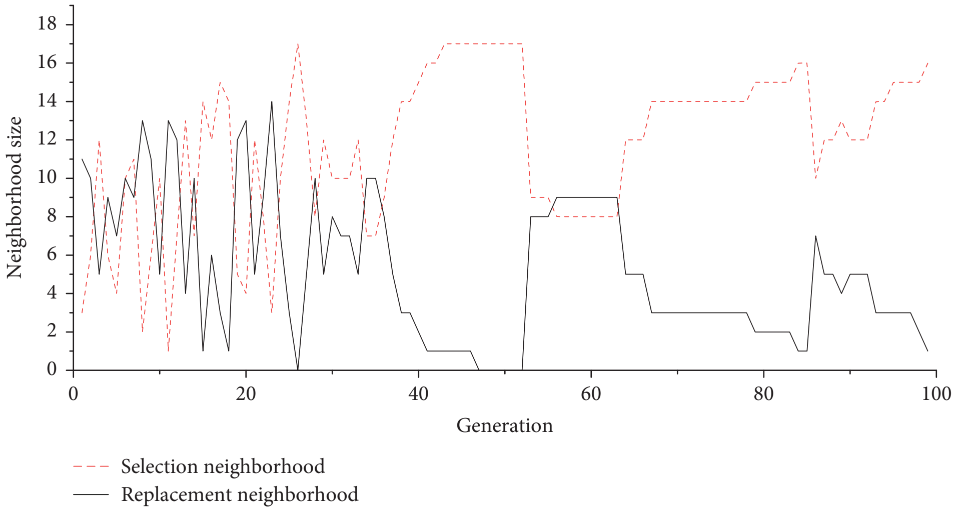


Figure 8: Variation of neighborhood sizes of the 96th individual.

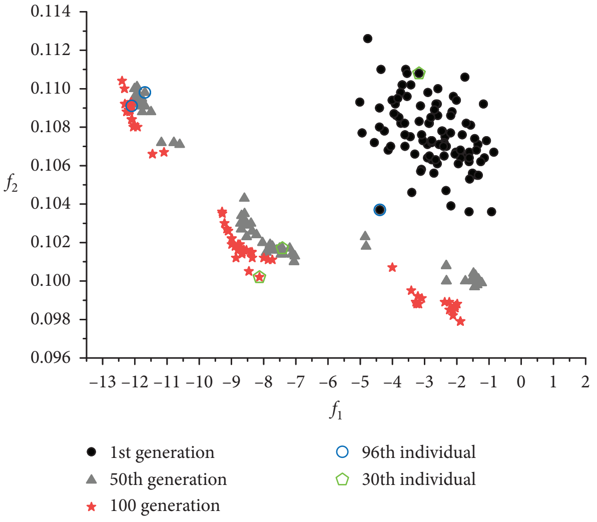


Figure 9: Population at the 1st, 50th, and 100th generation.

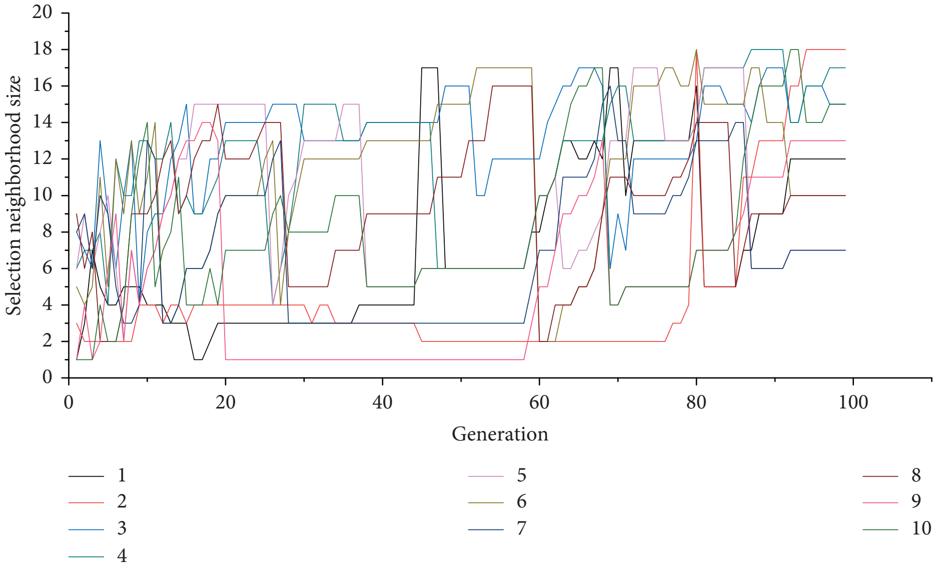


Figure 10: Variation of selection neighhorhood sizes for individuals 1 to 10.

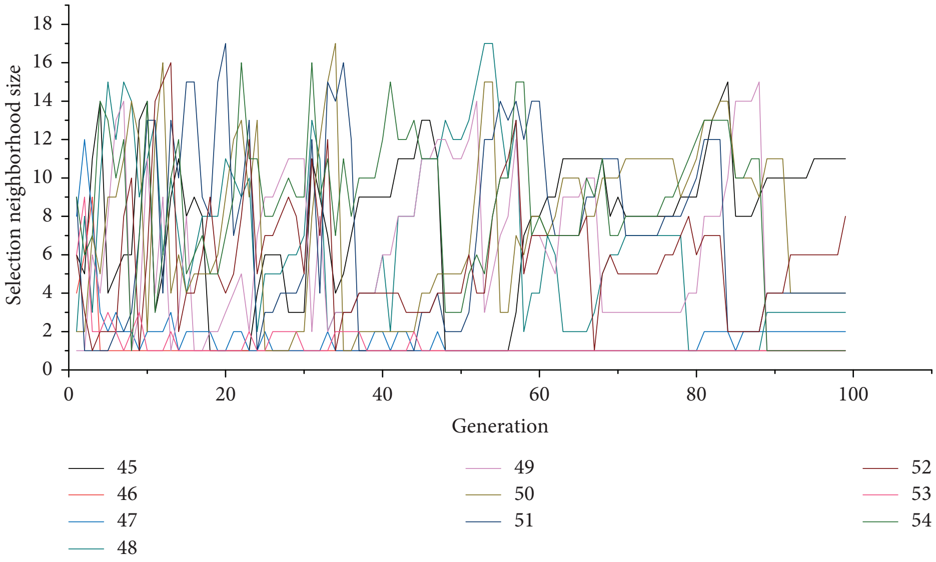


Figure 11: Variation of selection neighhorhood sizes for individuals 45 to 54.

arises. Recall that similar to original MOEA/D, MOEA/D-DDN evenly generated weight vectors for each individual, and then the population was initialized randomly. The weight vector remains unchanged during the whole evolution process. Possibly, the weight vector assigned to an individual is not the most appropriate one for the individual itself. For instance, in the initial population, the 96th individual almost lied in the middle of the population in the objective space. However, the assigned weight vector decides that the 96th individual searches along the aside direction. In other words, this type of individuals have longer paths to search in the objective space. Given evenly allocated computation resource, the performance might be worse.

Then, we assume that different parts of individuals in the population contribute differently to the convergence performance. We selected 3 sets of individuals in the population. Set 1 includes the first 10 individuals, set 2 includes individuals from 45 to 54, and set 3 consists of the last 10 individuals. The s-neighborhood sizes for each individual set during the evolution are, respectively, reported in Figures 10–12. A common feature shared by 3 sets is that the change frequency of s-neighborhood size decreased with evolution. This indicates the convergence of the population. By looking at initial populations, one can see that the spreads of s-neighborhood size for each set are similar. For sets 1 and 3, s-neighborhood sizes increased at the end of the algorithm for most individuals. However, the same values decreased for most individuals in set 2. Given an evenly spread of the population, a lower value of s-neighborhood size usually indicates a better value of nondomination rank for an individual. Then, it might be safe to state that, given evenly assigned weight vector and randomly generated initial

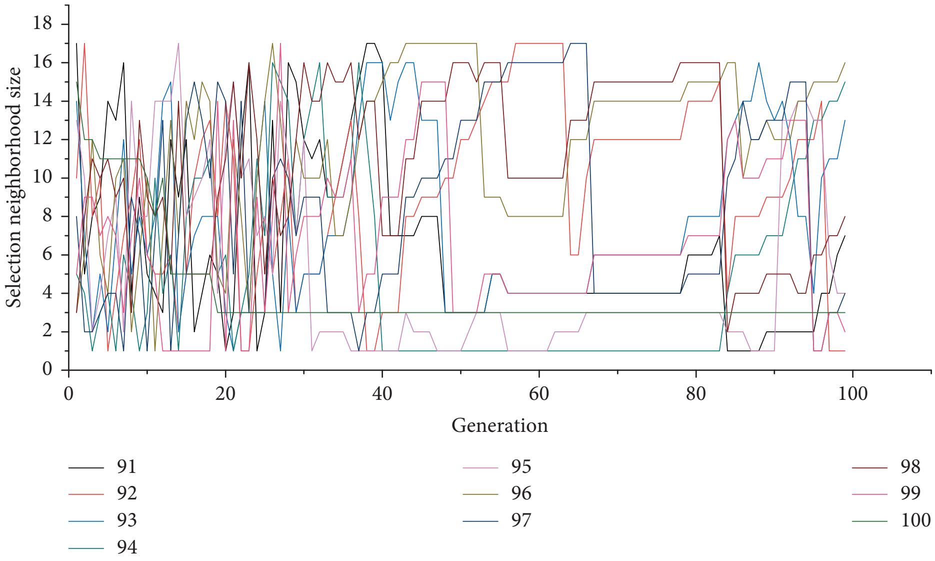
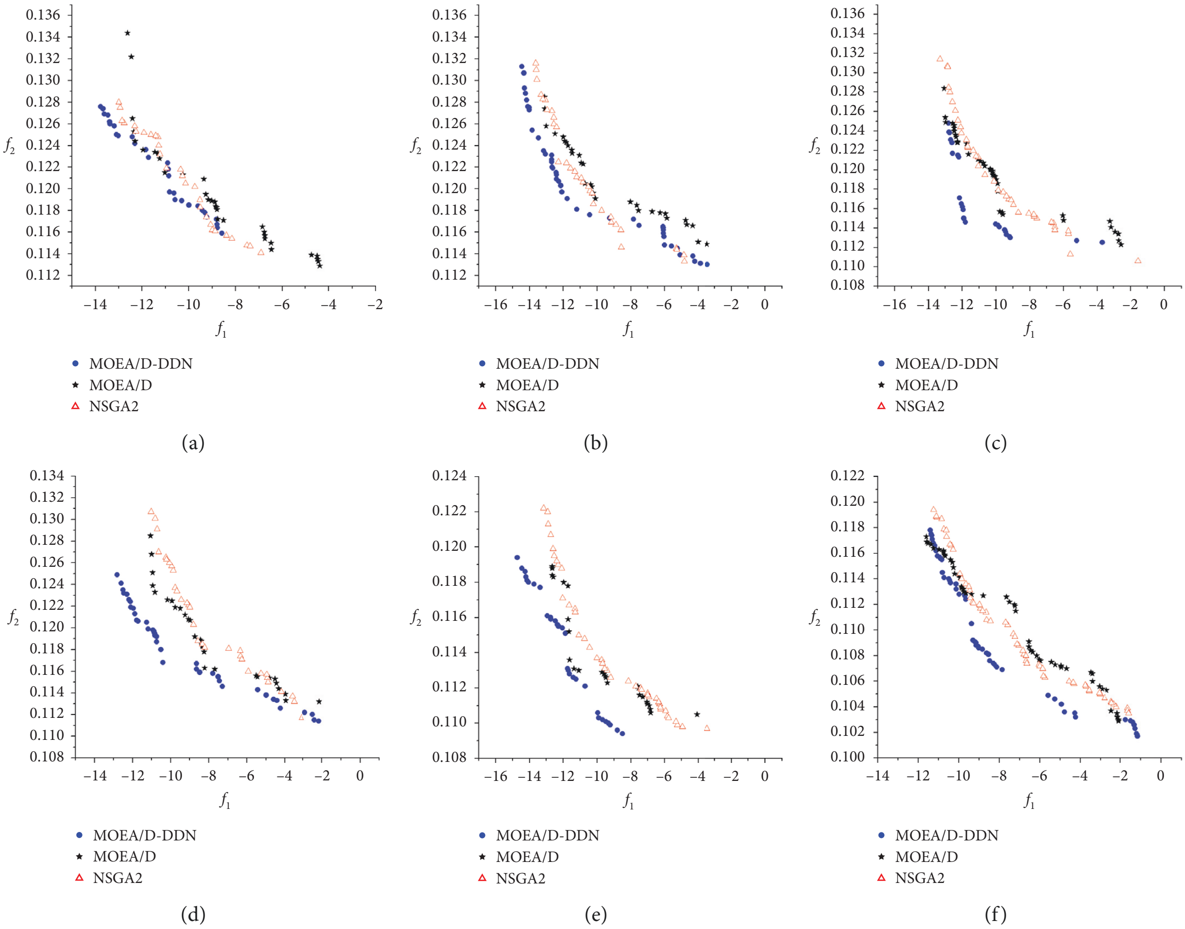


Figure 12: Variation of selection neighhorhood sizes for individuals 91 to 100.



(a)

(b)

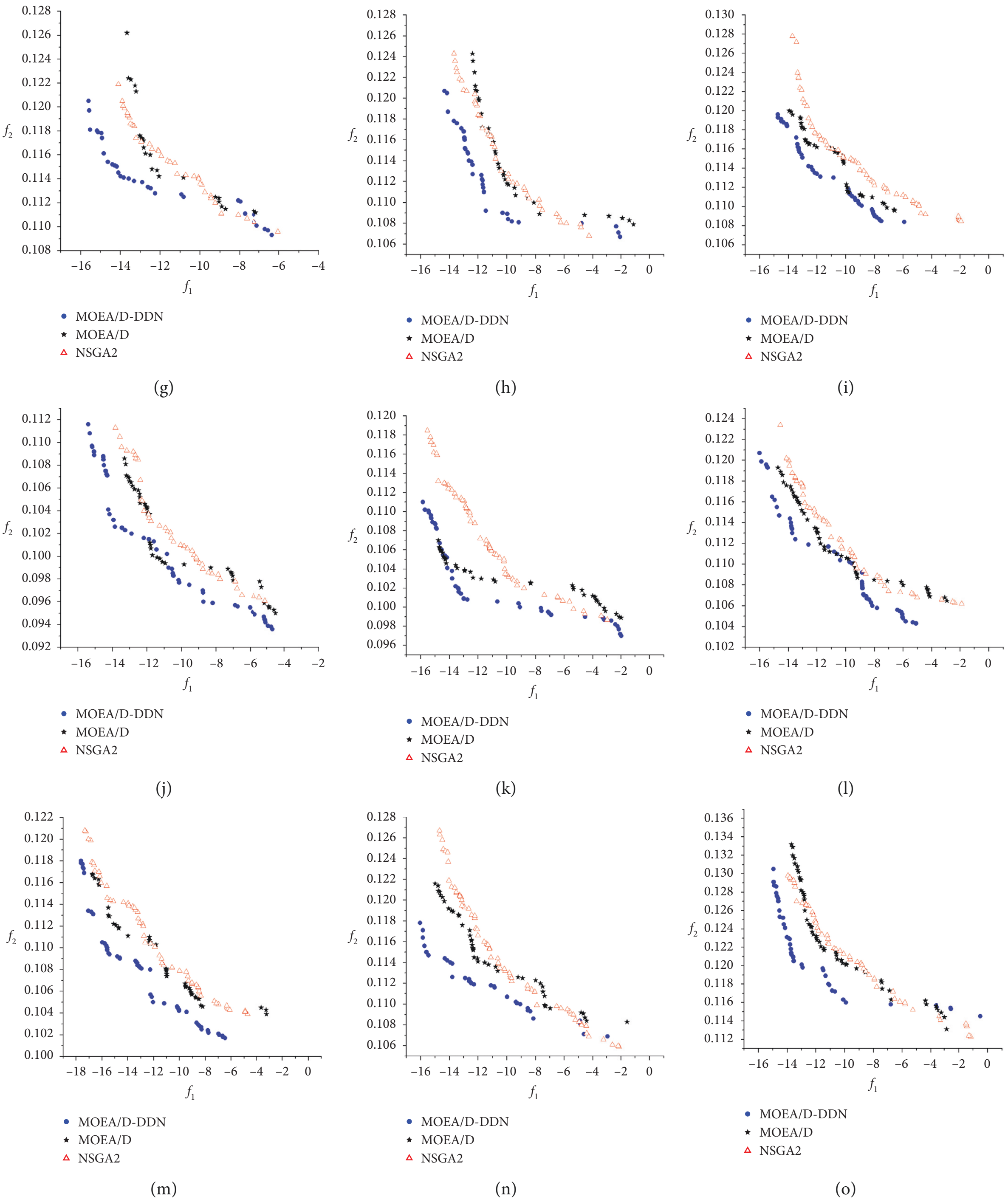
(c)

(d)

(e)

(f)

Figure 13: Continued.



(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o)

Figure 13: Obtained approximation of Pareto optimal solutions by MOEA/D-DDN, MOEA/D, and NSGA-II: (a) instance 1; (b) instance 2; (c) instance 3; (d) instance 4; (e) instance 5; (f) instance 6; (g) instance 7; (h) instance 8; (i) instance 9; (j) instance 10; (k) instance 11; (l) instance 12; (m) instance 13; (n) instance 14; (o) instance 15.

population, it is more difficult to put individuals located on end parts of the approximated Pareto front towards the true Pareto front. Some works have addressed similar issues. Li et al. [46] proposed a stable matching model for selection process of MOEA/D. This technique can be employed, and the performance of MOEA/D-DDN might be improved further for more complicated problems. However, this paper focus on the effect of distance-based divided neighborhoods for combinatorial problems. Then, further improvement of MOEA/D-DDN is not considered currently but left in future researches.

6.7. Further Performance Comparison.

MOEA/D and NSGA-II are two of the classic algorithms for solving multiobjective problems. The existing literature suggests that MOEA/D outperforms or performs similarly to NSGA-II for continuous and combinatorial multiobjective problems [29]. Recently, combinations of decomposition mechanism and NSGA-II components were successfully used to solve combinatorial problems [47]. Latest research studies also suggest the effectiveness of NSGA-II for solving similar scheduling problems [48–50]. Thus, in this paper, we choose NSGA-II as the baseline algorithm for comparison. In this subsection, we compared the performance of MOEA/D-DDN and NSGA-II as well as original MOEA/D. All parameters were set as the same with MOEA/D-DDN. Figure 13 reports the obtained approximation of Pareto optimal solutions by MOEA/D-DDN, MOEA/D, and NSGA-II for all instances. It is clear that MOEA/D-DDN achieved the best results in terms of convergence for all instances. However, it should be noted that, for some instances, NSGA-II performed better in terms of maximum spread, for example, for instances 1, 3, 14, and 15. The performances of MOEA/D and NSGA-II are comparable for both convergence and maximum spread. Then, we can conclude that the proposed MOEA/D-DDN can solve the addressed WSPPs better, compared with original MOEA/D and NSGA-II.

7. Conclusion

In capability planning area, decision makers face the problem of deciding the type and the amount of weapons. Furthermore, decision makers have to plan when to develop each unit of weapons. This type of problem is termed as weapon selection and planning problems (WSPPs). This paper refines the multiobjective model of WSPPs. The problem is combinatorial, and multiple types of decision variables increase the problem's complexity. MOEA/D is employed as the basic optimizer for solving the problem. However, the ability of original MOEA/D with weighted sum is limited due to the discontinuous, nonconvex, and irregular Pareto front. To overcome the possible drawbacks or MOEA/D, two mechanisms are designed: (1) dividing the original neighborhood of each individual as selection and replacement neighborhoods and (2) controlling the neighborhood sizes by constraining the search scope of each subproblem. The proposed algorithm is termed as MOEA/D-DDN. The neighborhood is divided by means of scalar function value. An additional parameter d is introduced to control the search scope of each subproblem.

To illustrate the addressed problem and investigate the performance of MOEA/D-DDN, benchmark instances with different scales were generated. Sensitivity analysis of d suggests that the optimal value of d needs to be identified for different problems. This might be a common challenge arisen to evolutionary algorithms. This issue is worthwhile to be addressed in future researches. The effectiveness of two designed mechanisms has been investigated. The performance of MOEA/D-DDN for solving WSPPs was compared with original MOEA/D and NSGA-II. The results suggest that MOEA/D-DDN performed better than other algorithms.

Currently, various MOEAs have been developed and applied to solve problems in different areas. However, most benchmarks are continuous problem. The ability and possible drawbacks of various MOEAs for complex combinatorial problems are still open issues. It might be interesting to address these issues in future researches.

Data Availability

The data used in the paper includes generated benchmarks and obtained results. All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 71501181 and 71871185.

References

[1] S. Hiromoto, *Fundamental capability portfolio management*, Ph.D. thesis, Pardee RAND Graduate School, Santa Monica, CA, USA, 2013.

[2] M. Workshop, "Capabilities based planning: the road ahead," Technical report, Institute for Defense Analyses, Arlington, VA, USA, 2004.

[3] J. Xiong, K.-w. Yang, J. Liu, Q.-s. Zhao, Y.-w. Chen, and Y. Chen, "A two-stage preference-based evolutionary multiobjective approach for capability planning problems," *Knowledge-Based Systems*, vol. 31, pp. 128–139, 2012.

[4] J. Kangaspunta, J. Liesiö, and A. Salo, "Cost-efficiency analysis of weapon system portfolios," *European Journal of Operational Research*, vol. 223, no. 1, pp. 264–275, 2012.

[5] J. Xiong, Z. Zhou, K. Tian, T. Liao, and J. Shi, "A multiobjective approach for weapon selection and planning problems in dynamic environments," *Journal of Industrial & Management Optimization*, vol. 13, no. 3, pp. 1189–1211, 2017.

[6] S. Arnone, A. Loraschi, and A. Tettamanzi, "A genetic approach to portfolio selection," *Neural Network World*, vol. 3, no. 6, pp. 597–604, 1993.

[7] D. Maringer and H. Kellerer, "Optimization of cardinality constrained portfolios with a hybrid local search algorithm," *OR Spectrum*, vol. 25, no. 4, pp. 481–495, 2003.

[8] T. Cura, "Particle swarm optimization approach to portfolio optimization," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 2396–2406, 2009.

[9] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 79–99, 2004.

[10] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck, "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm," *European Journal of Operational Research*, vol. 199, no. 3, pp. 684–693, 2009.

[11] G. A. V. Pai and T. Michel, "Evolutionary optimization of constrained k-means clustered assets for diversification in small portfolios," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1030–1053, 2009.

[12] W. J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk, "Multi-objective decision analysis for competence-oriented project portfolio selection," *European Journal of Operational Research*, vol. 205, no. 3, pp. 670–679, 2010.

[13] T. Kremmel, J. Kubalík, and S. Biffl, "Software project portfolio optimization with advanced multiobjective evolutionary algorithms," *Applied Soft Computing*, vol. 11, no. 1, pp. 1416–1426, 2011.

[14] 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.

[15] K. P. Anagnostopoulos and G. Mamanis, "A portfolio optimization model with three objectives and discrete variables," *Computers & Operations Research*, vol. 37, no. 7, pp. 1285–1297, 2010.

[16] K. P. Anagnostopoulos and G. Mamanis, "The mean-variance cardinality constrained portfolio optimization problem: an experimental evaluation of five multiobjective evolutionary algorithms," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14208–14217, 2011.

[17] K. Metaxiotis and K. Liagkouras, "Multiobjective evolutionary algorithms for portfolio management: a comprehensive literature review," *Expert Systems with Applications*, vol. 39, no. 14, pp. 11685–11698, 2012.

[18] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

[19] M. A. Greiner, J. W. Fowler, D. L. Shunk, W. M. Carlyle, and R. T. McNutt, "A hybrid approach using the analytic hierarchy process and integer programming to screen weapon systems projects," *IEEE Transactions on Engineering Management*, vol. 50, no. 2, pp. 192–203, 2003.

[20] S.-C. Yang, T.-L. Lin, T.-J. Chang, and K.-J. Chang, "A semi-variance portfolio selection model for military investment assets," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2292–2301, 2011.

[21] H. A. Abbass, A. Bender, H. H. Dam, S. Baker, J. Whitacre, and R. Sarker, "Computational scenario-based capability planning," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 1437–1444, Atlanta, GA, USA, July 2008.

[22] L. T. Bui, M. Barlow, and H. A. Abbass, "A multi-objective risk-based framework for mission capability planning," *New Mathematics and Natural Computation*, vol. 5, no. 2, pp. 459–485, 2009.

[23] B. Golany, M. Kress, M. Penn, and U. G. Rothblum, "Network optimization models for resource allocation in developing military countermeasures," *Operations Research*, vol. 60, no. 1, pp. 48–63, 2012.

[24] F. Ghasemzadeh, N. Archer, and P. Iyogun, "A zero-one model for project portfolio selection and scheduling," *Journal of the Operational Research Society*, vol. 50, no. 7, pp. 745–755, 1999.

[25] H. Sun and T. Ma, "A packing-multiple-boxes model for R&D project selection and scheduling," *Technovation*, vol. 25, no. 11, pp. 1355–1361, 2005.

[26] A. F. Carazo, T. Gómez, J. Molina, A. G. Hernández-Díaz, F. M. Guerrero, and R. Caballero, "Solving a comprehensive model for multiobjective project portfolio selection," *Computers & Operations Research*, vol. 37, no. 4, pp. 630–639, 2010.

[27] F. Streichert, H. Ulmer, and A. Zell, "Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 932–939, IEEE, Portland, OR, USA, June 2004.

[28] D. S. Yamashita, V. A. Armentano, and M. Laguna, "Robust optimization models for project scheduling with resource availability cost," *Journal of Scheduling*, vol. 10, no. 1, pp. 67–76, 2007.

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

[30] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, and T. Zhang, "On the effect of reference point in MOEA/D for multi-objective optimization," *Applied Soft Computing*, vol. 58, pp. 25–34, 2017.

[31] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[32] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[33] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.

[34] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1808–1820, 2014.

[35] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

[36] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016.

[37] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2015.

[38] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pp. 2508–2515, IEEE, Trondheim, Norway, May 2009.

[39] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 475–480, 2016.

[40] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[41] C. Zhang, K. C. Tan, L. H. Lee, and L. Gao, "Adjust weight vectors in MOEA/D for bi-objective optimization problems with discontinuous Pareto fronts," *Soft Computing*, vol. 22, no. 12, pp. 3997–4012, 2018.

[42] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE*

Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 284–302, 2009.

[43] F. Streichert, H. Ulmer, and A. Zell, "Comparing discrete and continuous genotypes on the constrained portfolio selection problem," in *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, vol. 3103, pp. 1239–1250, Seattle, WA, USA, June 2004.

[44] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[45] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: a survey," *ACM Computing Surveys*, vol. 52, no. 2, pp. 26–38, 2019.

[46] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

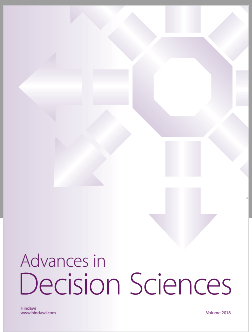
[47] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

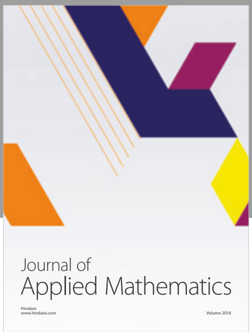
[48] J. Long, Z. Zheng, X. Gao, and P. M. Pardalos, "A hybrid multi-objective evolutionary algorithm based on NSGA-II for practical scheduling with release times in steel plants," *Journal of the Operational Research Society*, vol. 67, no. 9, pp. 1184–1199, 2016.

[49] H. Wang, Y. Fu, M. Huang, G. Q. Huang, and J. Wang, "A NSGA-II based memetic algorithm for multiobjective parallel flowshop scheduling problem," *Computers & Industrial Engineering*, vol. 113, pp. 185–194, 2017.

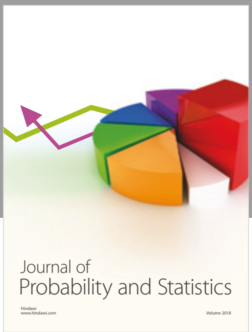
[50] Y. Fu, J. Ding, H. Wang, and J. Wang, "Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in industry 4.0-based manufacturing system," *Applied Soft Computing*, vol. 68, pp. 847–855, 2018.

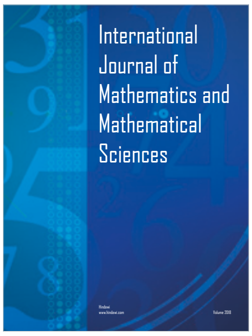
Advances in Operations Research


Advances in Decision Sciences

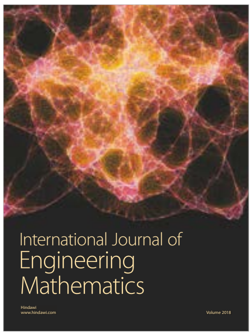

Journal of Applied Mathematics

The Scientific World Journal

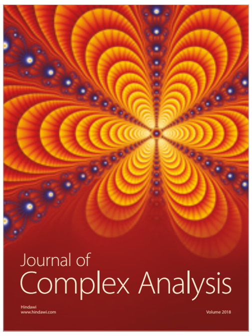

Journal of Probability and Statistics

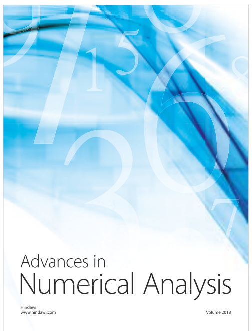

International Journal of Mathematics and Mathematical Sciences

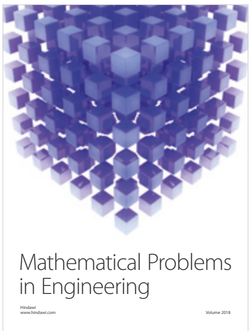
Journal of Optimization

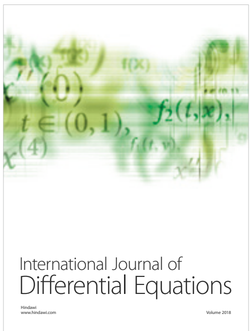

International Journal of Engineering Mathematics

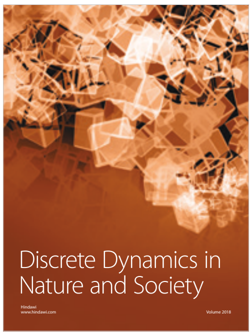
International Journal of Analysis

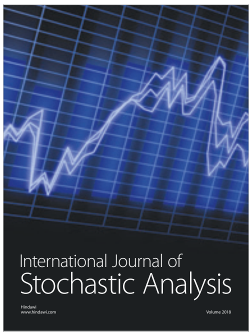

Journal of Complex Analysis


Advances in Numerical Analysis


Mathematical Problems in Engineering

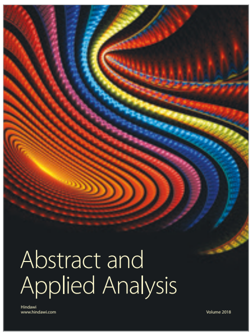

International Journal of Differential Equations

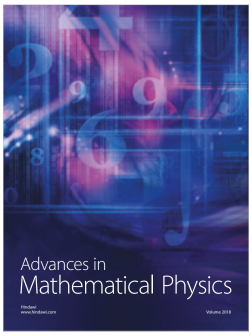

Discrete Dynamics in Nature and Society


International Journal of Stochastic Analysis

Journal of Mathematics

Journal of Function Spaces


Abstract and Applied Analysis


Advances in Mathematical Physics