Received October 18, 2020, accepted October 25, 2020, date of publication October 29, 2020, date of current version November 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034754

A Two-Stage Adjustment Strategy for Space Division Based Many-Objective Evolutionary Optimization

WEN ZHONG[1], XUEJUN HU[2], FA LU[3], JIANJIANG WANG[1], XIAOLU LIU[1], AND YINGWU CHEN[1]

[1]College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[2]Business School, Hunan University, Changsha 410082, China
[3]Joint Service Defense College, National Defense University, Beijing 100858, China

Corresponding author: Xuejun Hu (xuejun_hu@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71801218, and in part by the Research Project of National University of Defense Technology under Grant ZK18-03-18.

ABSTRACT Decomposition-based evolutionary algorithms, especially the branch based on objective space division using a set of uniformly distributed reference vectors, are widely envisioned as a promising technique to solve many-objective optimization problems. Nevertheless, their performance deteriorates severely when solving problems with irregular Pareto fronts (PFs), such as inverted, degenerated, and discontinuous PF shapes. So far, there are some works trying to design reference vector adjustment approaches to make up for such deficiencies of decomposition-based evolutionary algorithms. Unfortunately, the task of designing effective reference vector adjustment approaches adapting to irregular PFs remains challenging. To tackle this challenge, we propose a Two-Stage Adjustment Strategy, namely TSAS, for space division based many-objective evolutionary optimization to deal with the irregular PFs. To be specific, the first stage attempts to approach the boundaries of objective spaces having solutions by inserting new reference vectors between obtained solutions and reference vectors; for achieving better diversity, the second stage adjusts the reference vectors on the basis of both the reference vectors having solutions and the objective vectors of solutions with better diversity, and also adds some reference vectors to explore the sparse sub-spaces. To verify the effectiveness of the proposed TSAS, extensive experiments on benchmarks are carried out to compare it with five recent representative algorithms using three widely-used metrics. The compared results demonstrate the superior performance of our proposal, especially it significantly outperforms all the five algorithms in 45 out of 65 test instances with respect to Inverted Generational Distance (IGD) metric. Furthermore, to test the performance of TSAS in solving real-world problems, 6 test instances from agile satellite task planning are used to compare its performance with five other algorithms. The experimental results show that the TSAS has the best performance on 5 out of 6 test instances.

INDEX TERMS Evolutionary computation, multi/many-objective optimization, objective space division, irregular Pareto-front, adaptive adjustment.

I. INTRODUCTION

Real-world optimization problems encountered in various fields usually require to optimize multiple conflicting objectives simultaneously. For instance, the hybrid electric vehicle control problem requires to simultaneously optimize seven objectives: fuel consumption, battery stress, battery state of charge, operation changes, emission, noise, and urban operation [1]. Besides, the workflow scheduling problem in clouds involves optimizing uncertainty, makespan, cost and resource efficiency [2]. These problems are called multi-objective optimization problems (MOPs).

Relying on the advantages of obtaining multiple solutions in one single run [3], population-based evolutionary algorithms, such as differential evolution [4], [5], simulated binary crossover [6], and memetic algorithms [7], have been

The associate editor coordinating the review of this manuscript and approving it for publication was Shun-Feng Su.

envisioned as promising techniques to balance different optimization objectives for MOPs. During the past three decades, a large number of relevant multi-objective evolutionary algorithms (MOEAs) have been reported [8], [9]. According to their environmental selection frameworks, these algorithms can be roughly classified into three categories: Pareto dominance-based, indicator-based, and decomposition-based frameworks [10], [11]. When solving MOPs with more than three objectives (i.e., MaOPs), the algorithms based on the former two frameworks (i.e., Pareto dominance-based and indicator-based MOEAs) face the limitations of the ''dominance resistance'' [12] and exponentially growing computational cost with the increase in objective number [13], respectively.

Compared with the other two frameworks, the framework based on decomposition has witnessed remarkable success in solving MaOPs, and a number of relevant MOEAs have been proposed [11], [14]. Further, these MOEAs derived from decomposition-based framework can be classified into two categories. The first one decomposes an MOP into a series of subproblems using a set of reference vectors, and then solves them via collaborative ways [15]. The second one divides the objective space of an MOP into a series of subspaces using a set of reference vectors, and then co-evolve these subspaces [16]–[19].

Although decomposition-based MOEAs have shown superior performance in optimizing MOPs or MaOPs with regular PFs, their performance seriously deteriorates on MaOPs with irregular PFs, such as, inverted, degenerate, discontinuous shapes [20], [21]. The main reason is the unsuitable pre-defined configuration of reference vectors, which have strong influence on the search directions of decomposition-based MOEAs, and even the distribution of their output population. When solving MaOPs with irregular PFs, some reference vectors fail to intersect with the PFs, and these reference vectors correspond to the same Pareto-optimal solution, or the corresponding subspaces have no solution. This further leads to reduction of diversity.

An intuitive way to improve the performance of decomposition-based MOEAs in handling irregular PFs is to progressively adjust the reference vectors during the search process. So far, commendable efforts have been made along this direction, where the distribution of reference vectors is adjusted according to obtained solutions to fit the PF shapes of MaOPs [21]–[23]. Nevertheless, as the number of optimization objectives of an MaOP increases, the number of non-dominated solutions needed to fit the PF explodes exponentially. As analyzed in work [20], to approximate the PF of an MaOP with m objectives, the required number of solutions is about 20^{m-1}. Inferring from this, for an MaOP with 10 objectives, the number of non-dominated solutions required is up to 512 billion [20], and it is almost impossible to obtain a dense distribution of non-dominated solutions over the entire Pareto front of an MaOP. Therefore, the task of developing efficient strategies adapting to complex PFs remains challenging, and require more efforts.

In this paper, we propose a new two-stage reference vector adjustment strategy, namely TSAS, to improve the performance of objective space division based MOEAs in solving the challenges of irregular PF shapes. The key technical contributions can be summarized as follows.

- A strategy is designed for the first stage to insert new reference vectors between obtained solutions and associated reference vectors, such striving to push reference vectors towards the boundaries of objective spaces having solutions.
- A strategy is developed for the second stage to adjust the reference vectors on the basis of both the reference vectors having solutions and the objective vectors of solutions with better diversity. In this strategy, some reference vectors are added to explore the sparse sub-spaces for a better trade-off between convergence and diversity.
- Based on synthetic benchmarks and real-world application problems, a large number of experiments are conducted to verify the effectiveness of the proposal by comparing it with other five representative algorithms.

This paper is organized as follows. The relevant works on reference vector adjustment methods for decomposition-based MOEA are summarized and analyzed in Section II, followed by proposing the TSAS to make up for the shortcomings of the existing works in Section III. Then, the compared experiments and analyses are reported in Section IV. Finally, the conclusion and two future research directions are presented in Section V.

II. RELATED WORK

In this section, we first introduce the formal representation and basic concepts of multi-objective optimization problems. Then, some related studies are summarized and analyzed.

A. BASIC CONCEPTS

The multi-objective optimization problems (MOPs) are generally formulated as:

$$\begin{cases} \text{Minimize} & F(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \cdots, f_m(\vec{x})], \\ \text{S.t.} & \vec{x} \in \Omega, \end{cases} \tag{1}$$

where $\vec{x} = (x_1, x_2, \cdots, x_n)$ denotes a decision vector, m and n respectively indicate the count of objectives and decision variables, $\Omega \subseteq \mathbb{R}^n$ represents the feasible area of optimization problem. The objective function $F(\vec{x})$ is to map each n-dimensional feasiable decision vector to a m-dimensional objective vector. Similar to the works [24]–[26], a MOP having more than three objectives (i.e., $m > 3$) is deemed as a many-objective optimization problem (MaOP).

Because of the conflicting nature in the optimization objectives, improving one often comes at the expense of worsening other ones [27]–[29]. As a consequence, there exist a series of solutions that compromise different optimization objectives, rather than one single solution minimizing the m optimization objectives simultaneously [17]. For two solutions $\vec{x}_1, \vec{x}_2 \in \Omega$,

solution $\vec{x}_1$ is defined to *dominate* $\vec{x}_2$ (represented as $\vec{x}_1 \prec \vec{x}_2$) if all the objective values of $\vec{x}_1$ is less than or equal to that of $\vec{x}_2$ and at least one objective value of $\vec{x}_1$ is less than that of $\vec{x}_2$. A solution $x^* \in \Omega$ is defined as Pareto optimal only if there are no other solutions dominating it. Generally speaking, the set of all the Pareto-optimal solutions of an MOP is called as Pareto set (*PS*) in decision space, i.e., $PS = \{\vec{x}^* | \nexists \vec{x} \in \Omega, \vec{x} \prec \vec{x}^*\}$. In addition, the mapping of *PS* in the objective space is called the Pareto-front (*PF*), i.e., $PF = \{F(\vec{x}) | \vec{x} \in PS\}$.

B. RELATED STUDIES

In recent years, the shortcomings of decomposition-based MOEAs that their performance seriously depends on the PF shapes of MaOPs has attracted great attention, and a series of adjustment approaches for reference vectors have been reported [21]. Current approaches can be roughly classified into three categories: preference-based, neighbors-based, and fitting-based adjustment approaches.

The **preference-based adjustment approaches** generally suppose that the preference information of decision-makers is available before or during optimization process, and then transform the preference information into reference vectors to direct the evolutionary algorithms to search the subspaces of interest on the PFs. For example, Wang *et al.* regarded the preference information specified by the decision-makers as reference vectors, and constantly added random preference points to adjust the set of reference vectors to co-evolve the population [30]. Li *et al.* followed a similar path, and tuned the uniformly distributed reference vectors to the region of concern to decision-makers [31]. Tomczyk *et al.* reported an algorithm, called cwMOEA/D, to co-evolve the subproblems based on the relative importance of different objectives specified by decision-makers [32]. Liu *et al.* employed preference radius to construct the preference areas by interacting with decision-makers during the optimization process [33]. Li *et al.* suggested the interactive way to direct decomposition-based MOEAs to search the solutions in interested areas. During the interactive session, the decision-makers needs to sort some candidate solutions to construct an approximated value function, which will be transformed into a set of reference vectors [34]. Nevertheless, the populations of these approaches may be misled by the reference vectors being near to the PFs and fall into local optimum [35]. Unlike these existing approches, this paper focuses on approximating the entire PFs.

For **neighbors-based adjustment approaches**, they adjust each reference vector according to the distance or angle between the reference vector and its nearest neighbor. For instance, to adjust the reference vectors, Li *et al.* designed an algorithm EMOSA, which first employed a set of pre-defined reference vectors to direct the search of evolutionary algorithms. During the search process, the EMOSA adjusted the reference vectors to keep the associated solutions away from their nearest neighbors [36]. The work [37] reported an adjustment method that each subproblem evolves one solution with the reference vector, which was tuned for minimizing the subproblem fitness and maximizing the distance from itself to the nearest neighboring reference vector. Different from the above two approaches pushing the reference vectors away from their neighboring ones, Ge *et al.* employed an incremental learning method to delete invalid reference vectors and generate new reference vectors being near to valid reference vectors [38]. Xu *et al.* reported a hierarchy-based adjustment approach, namely MOEA/HD, to group the subproblems into different hierarchies, and adjust the reference vectors for lower-hierarchy subproblems using the perpendicular bisectors between the objective vectors of the solutions associated to the neighboring upper-hierarchy subproblems [39]. The neighbors-based adjustment approaches are based on local tuning procedures, and the distribution of the obtained solutions' objective vectors is not fully utilized. Besides, the approaches along this branch tend to be high time complexity and be ineffective in dealing with MaOPs with degenerated PFs [21].

Regarding the **fitting-based adjustment approaches**, they first explore to estimate the PF shapes of MaOPs via the fitness of the obtained solutions, and then construct a new set of reference vectors on the basis of the estimated PF shapes. For example, the works [40], [41] respectively employed a piecewise linear and a cubic spline interpolation on non-dominated solutions to fit the PF shapes of MOPs. Then reference vectors are uniformly constructed on the fitted PFs [40], [41]. Liu *et al.* chose a set of diverse solutions periodically, and normalized their objective vectors as reference vectors [42]. Wu *et al.* employed the gaussian process regression to estimate the population distribution to approximate the PF shapes of MaOPs, and then constructed reference vectors on the basis of the fitting model [43]. Liu *et al.* explored to learn the PF topology of MaOPs using the growing neural gas network, and then adjusts the reference vectors and scalarizing functions based on the learned topology [44]. To improve the performance of decomposition-based many-objective evolutionary algorithm, Han *et al.* designed two strategies to adaptively adjust the penalty factor for each subproblem and the reference vectors, respectively [45]. Elarbi *et al.* combined penalty-based boundary and normal-boundary intersection directions to handle the complex PFs [46]. But this hybrid approach still cannot guarantee that all reference vectors intersect the irregular PFs, and resorted to decision makers for adding more reference vectors. As stated by Ishibuchi *et al.* [20], the number of non-dominated solutions required to approximate the PF shapes of MaOPs explodes exponentially with the increase in the number of optimization objectives. For MOPs with many objectives, it is almost impossible to obtain a dense distribution of non-dominated solutions over the entire PFs for approximating their PF shapes. Unlike these existing works, in this paper, we strive to approach the boundaries of objective spaces having solutions, and add exploratory reference vectors in sparse areas where the solutions may exist.

III. ALGORITHM DESIGN

How to adaptively adjust the reference vectors to efficiently deal with irregular PF shapes of MOPs is still a challenge. In this section, we design a two-stage adjustment strategy to resolve this challenging task. First, a general framework for embedding adaptive adjustment strategies for reference vectors is introduced. Then, an environment selection strategy that mixing space division based mechanism with diversity-first mechanism is proposed. Finally, the proposed two-stage adjustment strategy for coping with irregular PF shapes of MOPs is detailed.

A. GENERAL FRAMEWORK

Space division-based multi-objective evolutionary algorithms (SD-MOEAs) are one branch of decomposition-based MOEAs, and they decompose the objective space of an MOP into N subspaces using N reference vectors [16], [17]. Fig. 1(a) provides a visual example to illustrate this decomposition framework in a 2-D objective space. As shown in Fig. 1(a), the reference vectors $v_1, v_2, \cdots, v_N$ are characterized as blue arrows, and the corresponding sub-spaces of these reference vectors are denoted as $\mathbb{S}_1, \mathbb{S}_2, \cdots, \mathbb{S}_N$. Besides, the dotted lines denote the boundaries of connected sub-spaces.

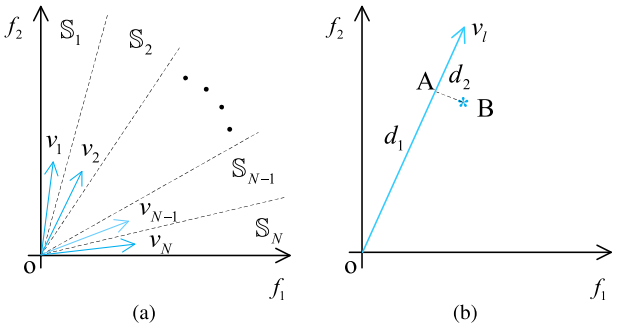

FIGURE 1. Examples of space division and fitness calculation.

In this work, we follow the framework of SD-MOEA, and develop a two-stage strategy to adjust the reference vectors for further improving their performance in dealing with irregular PF shapes. The framework of the proposed *TSAS* is shown in Algorithm 1.

As described in Algorithm 1, the proposed *TSAS* requires an user to provide three types of information: 1) the MOP to be solved; 2) the maximum number of evaluations; 3) the population size. After the *TSAS* runs, a population approximating the entire PF of the MOP will be given to users. Before iterative optimization, the algorithm *TSAS* first initializes the following six parameters: a set of uniformly distributed reference vectors (Line 1); one population (Line 2): the number of used fitness evaluations (Line 3); the maximal number of generations (Line 4); the current generation (Line 5); and switch flag of adjustment strategies (Line 6). Note that if the flag is TRUE, the first stage adjustment strategy will be triggered, otherwise, the second one will be triggered.

Algorithm 1: General Framework of *TSAS*

Input: MOP; maximum function evaluations (*MFE*); the population size N;
Output: An output population $\mathbb{P}$;

1 $V \leftarrow$ Generate a set of reference vectors;
2 $\mathbb{P} \leftarrow$ Randomly create a population with N individuals;
3 $FE \leftarrow N$;
4 $G_{max} \leftarrow \frac{MFE}{N}$;
5 $Gen \leftarrow 0$;
6 $Flag \leftarrow$ TRUE;
7 **while** $FE < MFE$ **do**
8 $\quad \mathbb{P}' \leftarrow$ OffspringGeneration($\mathbb{P}$);
9 $\quad FE \leftarrow FE + N$;
10 $\quad \mathbb{P} \leftarrow$ EnvironmentalSelection($V, \mathbb{P}' \bigcup \mathbb{P}$);
11 $\quad Gen \leftarrow Gen + 1$;
12 $\quad$ **if** $mod(Gen, \alpha \times G_{max})==0$ **then**
13 $\quad\quad$ **if** *Flag* **then**
14 $\quad\quad\quad V' \leftarrow$FirstStageAdjustStrategy($V, \mathbb{P}$);
15 $\quad\quad\quad V \leftarrow V \bigcup V'$;
16 $\quad\quad\quad Flag \leftarrow$ FALSE;
17 $\quad\quad$ **else**
18 $\quad\quad\quad V \leftarrow$SecondStageAdjustStrategy($V, \mathbb{P}, N$);
19 $\quad\quad\quad Flag \leftarrow$ TRUE;

Then, the *TSAS* enters the main loop, including new population generation (Line 8), environmental selection (Line 10), and two-stage reference vector adjustment (Lines 12-19). Similar to the works [15], [47], during each iteration, the simulated binary crossover is first employed to generate a new offspring population and then the polynomial mutation is used to modify the population. For the environmental selection strategy, it will be introduced in Algorithm 2.

During the optimization process, the two-stage adjustment strategy is periodically triggered to adjust the reference vectors to approach the boundaries of objective spaces having solutions, and explore the sparse areas where the solutions may exist. The parameter α in line 12 satisfies $0 < \alpha < 1$, and is used to control the adjustment frequency. The statement in line 12 means that the two-stage adjustment strategy is triggered every $\alpha \times G_{max}$ generations. The first stage adjustment strategy is described in Algorithm 3, and the second one is detailed in Algorithm 4.

B. ENVIRONMENTAL SELECTION

The environmental selection operator consists of three steps. The first step is to divide the objective space of an MaOP into a series of sub-spaces, associate each solution to a sub-space with the minimal acute angle. Then, a solution with the best fitness in each sub-space is selected during the second step. Facing an MaOP with an irregular PF shape, there exist some sub-spaces that do not contain solutions. Thus the number of selected solutions may be less than the population size. During the third step, the unselected solutions with the best

Algorithm 2: EnvironmentalSelection($V, \mathbb{Q}$)

Input: The set of reference vectors V; the combined population $\mathbb{Q}$; the population size N;
Output: The selected population $\mathbb{P}$;

1 $\mathbb{S}_i \leftarrow \emptyset, i = 1, 2, \cdots, |V|$;
2 **foreach** $q_j \in \mathbb{Q}$ **do**
3 $\quad i^* \leftarrow \arg\min_{v_i \in V} Angle(q_j, v_i)$;
4 $\quad \mathbb{S}_{i^*} \leftarrow \mathbb{S}_{i^*} \bigcup \{q_j\}$;
5 $\mathbb{P} \leftarrow \emptyset$;
6 **foreach** $i = 1 \longrightarrow |V|$ **do**
7 $\quad$ **if** $\mathbb{S}_i \neq \emptyset$ **then**
8 $\quad\quad q_i \leftarrow$ Select the solution with the best fitness in $\mathbb{S}_i$;
9 $\quad\quad \mathbb{P} \leftarrow \mathbb{P} \bigcup \{q_i\}$;
10 $\mathbb{A} \leftarrow \mathbb{Q} \setminus \mathbb{P}$;
11 **while** $|\mathbb{P}| < N \wedge \mathbb{A} \neq \emptyset$ **do**
12 $\quad q_i \leftarrow$ Select one solution from $\mathbb{A}$ having the largest angles with all the solutions in $\mathbb{P}$;
13 $\quad \mathbb{P} \leftarrow \mathbb{P} \bigcup \{q_i\}$;
14 $\quad \mathbb{A} \leftarrow \mathbb{A} \setminus \{q_i\}$;

Algorithm 3: FirstStageAdjustStrategy($V, \mathbb{P}$)

Input: The set of reference vectors V; the population $\mathbb{P}$;
Output: A set of new reference vectors V';

1 $\mathbb{S}_i \leftarrow \emptyset, i = 1, 2, \cdots, |V|$;
2 **foreach** $p_j \in \mathbb{Q}$ **do**
3 $\quad i^* \leftarrow \arg\min_{v_i \in V} Angle(p_j, v_i)$;
4 $\quad \mathbb{S}_{i^*} \leftarrow \mathbb{S}_{i^*} \bigcup \{p_j\}$;
5 $V' \leftarrow \emptyset$;
6 **foreach** $i = 1 \longrightarrow |V|$ **do**
7 $\quad$ **if** $\mathbb{S}_i \neq \emptyset$ **then**
8 $\quad\quad p_{j*} \leftarrow \min_{p_j \in \mathbb{S}_i} Angle(p_j, v_i)$;
9 $\quad\quad \theta \leftarrow Angle(p_{j*}, v_i)$;
10 $\quad\quad$ **if** $\theta > \Delta$ **then**
11 $\quad\quad\quad v' \leftarrow [\frac{f_1(p_{j*})}{\Sigma_{k=1}^{m} f_k(p_{j*})}, \cdots, \frac{f_m(p_{j*})}{\Sigma_{k=1}^{m} f_k(p_{j*})}]$;
12 $\quad\quad\quad v* \leftarrow 0.5 \times (v_i + v')$;
13 $\quad\quad\quad V' \leftarrow V' \bigcup \{v*\}$;

Algorithm 4: SecondStageAdjustStrategy($V, \mathbb{P}, N$)

Input: The set of reference vectors V; the population $\mathbb{P}$; the population size N;
Output: A set of new reference vectors V;

1 $\mathbb{S}_i \leftarrow \emptyset, i = 1, 2, \cdots, |V|$;
2 **foreach** $p_j \in \mathbb{Q}$ **do**
3 $\quad i^* \leftarrow \arg\min_{v_i \in V} Angle(p_j, v_i)$;
4 $\quad \mathbb{S}_{i^*} \leftarrow \mathbb{S}_{i^*} \bigcup \{p_j\}$;
5 $V' \leftarrow \emptyset$;
6 **foreach** $i = 1 \longrightarrow |V|$ **do**
7 $\quad$ **if** $\mathbb{S}_i \neq \emptyset$ **then**
8 $\quad\quad V' \leftarrow V' \bigcup \{v_i\}$;
9 **foreach** $p_j \in \mathbb{P}$ **do**
10 $\quad v' \leftarrow [\frac{f_1(p_j)}{\Sigma_{k=1}^{m} f_k(p_j)}, \cdots, \frac{f_m(p_j)}{\Sigma_{k=1}^{m} f_k(p_j)}]$;
11 $\quad V' \leftarrow V' \bigcup \{v_i\}$;
12 $V \leftarrow$ Select the extreme reference vectors from V';
13 $RV \leftarrow V' \setminus V$;
14 **while** $|V| < N \wedge RV \neq \emptyset$ **do**
15 $\quad rv* \leftarrow \arg\min_{rv \in RV, v \in V} Angle(rv, v)$;
16 $\quad V \leftarrow V \bigcup \{rv*\}$;
17 $\quad RV \leftarrow RV \setminus \{rv*\}$;
18 **while** $|V| < (1 + \beta) \times N$ **do**
19 $\quad [v_i, v_j] \leftarrow$ Select two adjacent reference vectors having the largest angles from V;
20 $\quad v' \leftarrow 0.5 \times (v_i + v_j)$;
21 $\quad V \leftarrow V \bigcup \{v'\}$;

diversity will be continuously selected until the number of selected solutions reaches the population size. The pseudocode of the environmental selection strategy is given in Algorithm 2.

As shown in Algorithm 2, the input parameters of the environmental selection operator are the set of reference vectors, the combined population, and the population size. After the operation, a selected population will be outputted. For space divison-based MOEA, each reference vector defines a subspace. The set $\mathbb{S}_i$ ($i = 1, 2, \cdots, |V|$) is used to record the solutions in the sub-space defined by the i-th reference vector, and initialized as empty (Line 1). Then, each solution in the combined population $\mathbb{Q}$ is associated to a reference vector with the minimal acute angle (Lines 2-4). After that, one solution with the best fitness in a sub-space is selected and added into $\mathbb{P}$ (Lines 6-9). When a solution p_j is associated to the sub-space defined by the reference vector v_i, its fitness is defined as $Fit(p_j, v_i) = ||F(p_j)|| \times \{\cos(p_j, v_i) + \sin(p_j, v_i)\}$. An instance in Fig. 1(b) is used to illustrated the fitness of a solution. Assuming that the solution B is associated to the sub-space defined by reference vector v_l and the projection point of solution B on v_l is A, its fitness is the sum of the segments $\overline{OA}$ and $\overline{AB}$.

Next, the unselected solutions are separated from the combined population, and recorded in $\mathbb{A}$ (Line 10). Before the number of selected solutions reaches the population size or all the solutions are selected, during each iteration, one solution in $\mathbb{A}$ with the largest acute angle to all the solutions in $\mathbb{P}$ is selected and added into $\mathbb{P}$ (Lines 12-13). The operator in line 14 is to remove the selected solution from the set $\mathbb{A}$.

C. ADAPTIVE ADJUSTMENT STRATEGY

As illustrated in line 11 of Algorithm 1, the adjustment strategy of first stage will be periodically triggered to add new reference vectors between the reference vectors and associated solutions to approach the boundaries

of areas having solutions. The pseudocode of function *FirstStageAdjustStrategy*() is shown in Algorithm 3.

From Algorithm 3, we can see that the inputs of this function are a set of reference vectors and the current popultion. The output of this function is a set of new reference vectors.

This function associates each solution in the current population $\mathbb{P}$ to a reference vector (Lines 1-4). After that, for each reference vector (Line 6), if it has associated solutions (Line 7), the acute angle between it and the objective vectors of these solutions will be calculated, and the solution having the minimal acute angle to this reference vector is selected (Line 8). Note that the formula $Angle(p_j, v_i)$ represents the acute angle between the objective vector of solution p_j and the reference vector v_i. Hereafter, if the minimal acute angle θ is larger than a threshold Δ (Line 10), the midpoint of the objective vector of selected solution and the reference vector is added as a new reference vector (Lines 11-13).

For space division-based evolutionary optimization, each reference vector directs some solutions forward to the PF along its opposite direction. If the acute angles between a reference vector and its associated solutions are very large, this generally means this reference vector does not intersect with the PF of an MaOP. Adding new reference vectors by the first stage adjustment strategy is conducive to continuously approaching the boundaries of PF.

In this work, the active reference vectors refer to these ones with associated solutions. The second stage adjustment strategy is to construct a set of active reference vectors having better diversity, and add exploratory reference vectors in sparse areas where the solutions may exist. The pseudocode of function *SecondStageAdjustStrategy*() is given in Algorithm 4.

As illustrated in Algorithm 4, the inputs of the second stage adjustment strategy are the set of reference vectors, the current population, and the population size. Its output is a set of new reference vectors. This strategy mainly includes four steps.

Step 1. The purpose of this step is to separate active and inactive reference vectors, which first associates each solution to a reference vector (Lines 1-4). Then, the reference vectors with associated solutions are regarded as active and added into the set V' (Lines 5-8).

Step 2. The objective vectors of solutions in the current population are normalized as candidate reference vectors (Lines 9-10), which are added into the set V' (Line 11).

Step 3. The reference vectors with the better diversities in V' are gradually selected. The extreme reference vectors are defined as the ones with minimum values in at least one dimension, and first selected (Line 12). Afterwards, the unselected reference vectors with the largest acute angle to all the selected ones are constantly selected until the number of selected ones reaches the population size or the unselected ones are empty (Lines 13-17).

Step 4. Its purpose is to add a proportion of exploratory reference vectors in the regions where solutions may exist. The proportion is represented by the parameter β (Line 18). In this step, two adjacent reference vectors with the largest angle are first selected (Line 19). Hereafter, the midpoint of these two selected reference vectors is added as a new exploratory reference vector (Lines 20-21).

IV. EXPERIMENTAL STUDIES

In this section, extensive comparative experiments are conducted to testify the effectiveness of our proposed *TSAS*. The five representative peer competitors are: *A-NSGA-III* [47], *MOEA/D-AM2M* [42], *MOEA/D* [15], *MOEA/D-AWA* [22], and *RVEA* [17]. The brief descriptions of these algorithms are shown as follows.

A-NSGA-III: It is an improved version of the classic MaOEA, NSGA-III. This algorithm first discriminates the inactive reference vectors, and then adjust them according to the distribution of current population.

MOEA/D-AM2M: It is a representative MOEA based on space division, and an adjustment strategy is integrated for adjusting the reference vectors. In this approach, the distribution of obtained population is used to estimate the PF shape of an MaOP, and periodically update the reference vectors. Then, the search efforts are adaptively assigned to promising subspaces.

MOEA/D: It decomposes an MaOP into a series of single objective sub-problems, and co-evolves these sub-problems in a collaborative way. For MOEA/D, there are three widely-used methods, i.e., weighted sum approach, Tchebycheff approach, and penalty-based boundary intersection approach, for calculating the fitnesses of solutions. In the experiments, the penalty-based boundary intersection method is chose for MOEA/D.

MOEA/D-AWA: This algorithm improves the **MOEA/D** with an adaptive adjustment approach for weight vectors. An external archive is employed to store non-dominated solutions for adjusting weight vectors. Then, the adaptive approach periodically deletes overcrowded weight vectors and adds new weight vectors into sparse subspaces.

RVEA: In this algorithm, a set of reference vectors are used to partition the objective space into a series of connected subspaces, and the angle-penalized distance is suggested to sort solutions in each subspace. Also, the reference vectors are periodically adapted according to the nadir and ideal points of obtained solutions.

Unless otherwise specified, the parameter settings of these five algorithms directly use the recommended parameters in the platform PlatEMO.[1]

A. EXPERIMENT DESIGN

1) POPULATION SIZE

Referring to existing works [42], [47], [48], the setting of population size N only considers the number of optimization objectives. For the 3-, 5-, 8-, 10-, and 15-objective test instances, their population sizes are set as 91, 126, 156, 230, and 240, respectively.

[1] https://github.com/BIMK/PlatEMO

2) COMPARISON METRICS

The inverted generational distance (*IGD*) [49] and Hypervolume (*HV*) [50] are two widely-used metrics to assess the performance of MOEAs with respect to both the convergence and diversity. Besides, the pure diversity (*PD*) [51] and *Spread* [52] are two popular metrics to measure the population diversity. The experiments in this paper employ the above four metrics to compare our proposal with other five peer competitors.

1) *IGD*: Regarding a population $\mathbb{P}$, the IGD value can be calculated as follows:

$$IGD(\mathbb{P}) = \frac{\sum_{v\in\mathbb{P}^*} \min_{p\in\mathbb{P}} dist(F(v), F(p))}{|\mathbb{P}^*|}, \tag{2}$$

where $\mathbb{P}^*$ corresponds to a set of well-distributed solutions on the *PF*, $dist(F(v), F(p))$ is the distance between solution v and p in objective space. On the basis of the definition in (2), an algorithm generating a smaller *IGD* value implies it performs better. Similar to the works, the $\mathbb{P}^*$ in the experiments is set to contain around 8000 points for each test instance.

2) *HV*: It is defined as the volume of space, which consists of a reference point and all the output solutions in objective space. The larger *HV* value means the better performance of the corresponding algorithm with respect to both the convergence and diversity. For each test instance, we set the reference point as 1.5 times of the upper bounds of its *PF*.

$$HV(\mathbb{P}) = L(\bigcup_{p\in\mathbb{P}}[f_1(p), r_1] \times \cdots \times [f_m(p), r_m]), \tag{3}$$

where $L(\bigtriangleup)$ stands for the Lebesgue measure.

3) *PD*: This metric for an output population $\mathbb{P}$ can be calculated as follows.

$$PD(\mathbb{P}) = \max_{p\in\mathbb{P}}(PD(\mathbb{P}-p) + dis(p, \mathbb{P}-p)), \tag{4}$$

where,

$$dis(p, \mathbb{P}-p) = \min_{q\in\mathbb{P}}(dissimilarity(p, q)). \tag{5}$$

Note that the formula $dis(p, \mathbb{P}-p)$ represents the dissimilarity of solution p to the population $\mathbb{P}$.

4) For metric *Spread* ($\bigtriangleup$), it measures the extent of spread archived among non-dominated solutions in a population. For an output population $\mathbb{P}$, its *Spread* $\bigtriangleup(\mathbb{P})$ can be calculated as follows:

$$\bigtriangleup(\mathbb{P}) = \frac{\sum_{i=1}^{m} d(E_i, \mathbb{P}) + \sum_{p\in\mathbb{P}} |d(p, \mathbb{P}) - \overline{d}|}{\sum_{i=1}^{m} d(E_i, \mathbb{P}) + (|\mathbb{P}| - m)\overline{d}}, \tag{6}$$

$$d(p, \mathbb{P}) = \min_{q\in\mathbb{P}, p\neq q} \|F(p) - F(q)\|, \tag{7}$$

$$\overline{d} = \frac{1}{|\mathbb{P}|}\sum_{p\in\mathbb{P}} d(p, \mathbb{P}), \tag{8}$$

where $(E_1, E_2, \cdots, E_m)$ denotes m extreme solutions in the population $\mathbb{P}$.

3) STOP CRITERIA

For fair comparisons, the maximum function evaluations are employed as stop conditions for the six algorithms.

B. COMPARISON RESULTS ON SYNTHETIC BENCHMARKS

The test suite reported in work [53] is the new versions of DTLZ and WFG test suites, and specially designed for evaluating many-objective optimization. Besides, it is embedded with a wide variety of challenging characteristics in both decision and objective spaces. From the perspective of decision space, this test suite contains a series of tough single-objective functions, such as the Rosenbrock's function, Rastrigin's function, and Griewank's function. For the objective spaces, this test suite is designed with various irregular PF shapes, such as disconnected, degenerate, inverted, and complex geometries. To verify the effectiveness of the proposed *TSAS* in dealing with irregular PF shapes, we choose thirteen benchmark functions, i.e., MaF1-MaF13, from this challenging test suite to compare *TSAS* with other five representative algorithms.

Generally, a test instance is regarded as a benchmark function with a specific objective count, for instance, the 10-objective MaF1 is a test instance. The number of optimization objectives is set as $m \in \{3, 5, 8, 10, 15\}$. Then, for the thirteen benchmark functions with five settings of the objective number, there exist 65 test instances in the experiments. Since the function MaF3 has a lot of local optima, the number of maximal function evaluations for all the test instances derived from this function is set as 400,000. For other test intances, this parameter is set as 40,000.

For each test instance, all the six algorithms run 30 times independently. Similar to the works [29], [48], [54]–[56], we use the Wilcoxon Ranksum test with 5% confidence level to test the significant differences between each compared algorithm and the proposed *TSAS*. The symbols $+$, $-$, and $\approx$ mean that the compared algorithm performs better than, worse than, and similar to the propsed *TSAS*.

With respect to the four metrics, the statistical comparison results of the six algorithms are summarized in the Table 1. For each metric, the results in the three rows indicate the ratio of test instances that the corresponding algorithm is superior, inferior and similar to the *TSAS*, respectively. Take the first three values (i.e., 15/65, 44/65, and 6/65) in the first column as an example, they corrspond to the ratio of test instances that the compared algorithm *A-NSGA-III* performs better than, worse than, and similar to the *TSAS* with respect to the metric PD.

From the results highlighted with the blue background, we can observe that the proposed *TSAS* prevails over the five comparison algorithms on the four metrics. For the metric PD, among the 65 test instances, the algorithm *TSEA* significantly performs better than *A-NSGA-III*, *MOEA/D-AM2M*, *MOEA/D*, *MOEA/D-AWA*, and *RVEA* on 44, 60, 60, 41, and 58 test instances, respectively. For the other three metrics, i.e., HV, IGD, and Spread, the proposed *TSAS* also illustrates the similar advantages. For instance, among the five comparison algorithm, the *A-NSGA-III* shows the best performance in terms of HV. The proposed *TSAS* still significantly outperforms the *A-NSGA-III* on 40 out of 65 test instances.

TABLE 1. The ratio of test instances where each compared algorithm performs better than (+), worse than (−), and similar to (≈) the propsed *TSAS* in terms of PD, HV, IGD, and Spread metrics.

Metrics		*A-NSGA-III*	*MOEA/D-AM2M*	*MOEA/D*	*MOEA/D-AWA*	*RVEA*
PD	+	15 / 65	5 / 65	2 / 65	14 / 65	2 / 65
	−	44 / 65	60 / 65	60 / 65	41 / 65	58 / 65
	≈	6 / 65	0 / 65	3 / 65	10 / 65	5 / 65
HV	+	23 / 65	20 / 65	10 / 65	22 / 65	20 / 65
	−	40 / 65	43 / 65	53 / 65	39 / 65	40 / 65
	≈	2 / 65	2 / 65	2 / 65	4 / 65	5 / 65
IGD	+	15 / 65	2 / 65	7 / 65	17 / 65	17 / 65
	−	48 / 65	62 / 65	58 / 65	47 / 65	46 / 65
	≈	2 / 65	1 / 65	0 / 65	1 / 65	2 / 65
Spread	+	12 / 65	12 / 65	24 / 65	27 / 65	22 / 65
	−	44 / 65	49 / 65	40 / 65	35 / 65	38 / 65
	≈	9 / 65	4 / 65	1 / 65	3 / 65	5 / 65

To compare the performance of six algorithms more concretely, we also report the average and variance of their IGD values on each test instance, and the results is shown in Table 2. For each test instance, the lowest *IGD* value among the six algorithms is highlighted with grey background. As illustrated in Table 2, the proposed *TSAS* shows competitive performance by generating lower IGD values than all the five comparison algorithms on 27 out of 65 test instances.

The algorithms *A-NSGA-III* and *MOEA/D-AWA* are two representative MaOEAs in dealing with MaOPs with complicated PF shapes, and show promising performance in the test suites DTLZ and WFG. By comparing the proposed *TSAS* and them on the challenging versions of these test suites, the comparison results in Table 2 demonstrate the competitve performance of the proposed *TSAS* in dealing with MaOPs with challenging PF shapes.

The comparison algorithm *MOEA/D-AM2M* also transforms the objective vectors of obtained solutions with better diversity as reference vectors. But, the performance of the proposed *TSAS* still outperforms *MOEA/D-AM2M*. This can be interpreted as that the proposed *TSAS* employs mechanisms of approaching boundaries of subspaces having solutions and adding exploratory reference vectors to explore the sparse areas.

Compared with PBI-based *MOEA/D*, the main difference of the proposed *TSAS* is the adaptive strategy for adjusting the reference vectors. From their comparison results, the effectiveness of the two-stage adjustment strategy in the proposed *TSAS* is verified.

The algorithm *RVEA* dynamically adjusts the reference vectors on the basis of the scales of objective values and employs the angle-penalized distance to sort solutions in each subspace. As the comparison results in Table 2, the proposed *TSAS* produces significantly lower IGD values than the *RVEA* on 46 out of 65 test instances.

The parallel coordinate [57] has been used to transform high-dimensional vectors into a two-dimensional graph. In a two-dimensional graph, each dimension of high-dimensional vectors is plotted on a vertical axis, then the polyline connecting the points on each axis corresponds to a high-dimensional vector. It has been frequently employed to visualize the distributions of output populations in evolutionary many-objective optimization community.

To visually compare the features of different algorithms, the parallel coordinate is adopted to illustrate the output population of each algorithm with the lowest IGD value among the 30 runs on 10-objective MaF1, MaF6, and MaF8. These functions have irregular PF shapes, such as inverted and degenerate, which are challenging for decomposition-based multi-objective optimization algorithms. The relevant results are shown in Figs. 2-4.

The MaF1 is designed for testing the capabilities of MaOEAs in dealing with inverted PFs, and the value range of its PF is between 0 and 1 in each dimension [53]. As illustrated in As shown in Figs. 2(a), (c), and (e), the algorithms *A-NSGA-III*, *MOEA/D*, and *RVEA* perform better in convergence, but there is still much room for improvement in their diversity. Among the five comparison algorithms, the *MOEA/D-AM2M* and *MOEA/D-AWA* exhibit good performance in both convergence and diversity. Comparing the Fig. 2(b) with Fig. 2(f), we can observe that the proposed *TSAS* shows the similarity to the convergence of *MOEA/D-AM2M*. Also, the diversity of *TSAS* is much better than *MOEA/D-AM2M*, especially in the third, fifth, eighth, ninth, and tenth objectives. These results are basically consistent with the IGD values in the fourth row of Table 2. The Fig. 2(d) illustrates that the *MOEA/D-AWA* has good convergence and diversity. But, the objective values of some solutions coming from *MOEA/D-AWA* is larger than 1. Besides, the diversity of *MOEA/D-AWA* in the second and ninth objectives is insufficient. The above two reasons can explain why the proposed *TSAS* has lower IGD value than *MOEA/D-AWA* on 10-objective MaF1.

The benchmark function MaF6 is a classical MaOP having degenerate PF shape, and we also choose it with 10 objectives to compare the six algorithms in handling such irregular PF. For test instance 10-objective MaF6, the value ranges of the ninth and tenth objectives on the PF are around [0,0.71] and [0,1.0], respectively. From Fig. 3(a) and Fig. 3(e), it is obvious

TABLE 2. The *IGD* values of the 6 algorithms on solving benchmarks MaF1-MaF13 with 3, 5, 8, 10, and 15 objectives.

MaOPs	m	*A-NSGA-III*	*MOEA/D-AM2M*	*MOEA/D*	*MOEA/D-AWA*	*RVEA*	*TSAS*
MaF1	3	4.4834e-2 (2.40e-4) −	6.9328e-2 (8.89e-3) −	7.0475e-2 (1.70e-7) −	4.2755e-2 (1.83e-4) −	8.2096e-2 (2.04e-4) −	4.2399e-2 (2.82e-4)
	5	2.5556e-1 (1.29e-2) −	1.4314e-1 (8.43e-3) −	1.5939e-1 (3.90e-3) −	1.2971e-1 (9.54e-4) −	3.4061e-1 (1.13e-1) −	1.2835e-1 (3.28e-3)
	8	2.8537e-1 (1.01e-2) −	3.0556e-1 (4.94e-2) −	4.3408e-1 (9.47e-3) −	2.5943e-1 (4.53e-2) −	6.2574e-1 (7.45e-2) −	2.0543e-1 (2.20e-2)
	10	2.9644e-1 (4.71e-3) −	3.3430e-1 (4.33e-2) −	4.6594e-1 (1.06e-2) −	3.4460e-1 (5.40e-2) −	6.6669e-1 (5.75e-2) −	2.2259e-1 (2.76e-2)
	15	3.1665e-1 (7.52e-3) −	3.8679e-1 (4.66e-2) −	5.7595e-1 (1.16e-4) −	3.5540e-1 (2.96e-2) −	7.0242e-1 (6.14e-2) −	2.7210e-1 (1.87e-2)
MaF2	3	3.1290e-2 (7.35e-4) −	3.2507e-2 (6.37e-4) −	3.8205e-2 (2.15e-4) −	3.0207e-2 (3.15e-4) ≈	4.0483e-2 (4.80e-4) −	3.0528e-2 (7.45e-4)
	5	1.2407e-1 (3.22e-3) −	1.1288e-1 (4.95e-3) −	1.2908e-1 (9.51e-5) −	1.0422e-1 (1.44e-3) −	1.3141e-1 (2.09e-3) −	1.0219e-1 (1.39e-3)
	8	2.3083e-1 (2.70e-2) −	2.3138e-1 (1.01e-1) −	2.1699e-1 (1.08e-4) −	1.4622e-1 (2.59e-3) +	2.1286e-1 (2.98e-3) −	1.7866e-1 (2.66e-3)
	10	2.2672e-1 (2.94e-2) −	5.2019e-1 (2.08e-1) −	3.2282e-1 (5.82e-4) −	1.6141e-1 (2.71e-3) +	2.5004e-1 (8.80e-2) −	1.9988e-1 (3.06e-3)
	15	2.2764e-1 (2.16e-2) −	8.4120e-1 (5.63e-2) −	3.6377e-1 (7.01e-5) −	2.1505e-1 (7.76e-3) −	6.9969e-1 (1.89e-1) −	2.0930e-1 (3.70e-3)
MaF3	3	5.0578e-2 (6.78e-3) +	6.8039e-1 (1.77e-1) −	5.4617e-2 (8.87e-7) +	1.3592e-1 (4.64e-2) −	4.1614e-2 (6.55e-5) +	6.7832e-2 (9.21e-4)
	5	8.4001e-2 (3.15e-4) +	7.7620e-1 (1.77e-1) −	1.1501e-1 (2.06e-5) +	2.0752e+10 (4e+10) −	7.3531e-2 (2.08e-3) +	1.2822e-1 (1.04e-1)
	8	4.3186e-1 (1.19e-1) −	7.1983e-1 (1.87e-1) −	1.6036e-1 (2.57e-5) −	5.5786e+10 (9e+10) −	9.1369e-2 (3.87e-3) +	1.3994e-1 (2.90e-3)
	10	2.6544e-1 (7.71e-2) −	6.0934e-1 (2.22e-1) −	1.3948e-1 (1.29e-5) −	1.4289e-1 (6.81e-3) −	7.9831e-2 (2.34e-3) +	1.3081e-1 (1.04e-2)
	15	1.8496e-1 (2.89e-2) +	6.6815e-1 (1.95e-1) −	1.3008e-1 (2.80e-5) +	1.7795e-1 (1.12e-1) +	9.5566e-2 (6.44e-3) +	4.5034e-1 (2.85e-1)
MaF4	3	3.3638e-1 (1.99e-2) +	3.3127e+0 (1.21e+0) −	6.5353e-1 (2.07e-3) −	2.6022e-1 (4.60e-3) +	4.1305e-1 (1.49e-1) +	4.8376e-1 (1.63e-2)
	5	3.1588e+0 (9.03e-1) +	1.1251e+1 (5.02e+0) −	8.3220e+0 (1.96e-1) −	1.9430e+0 (3.12e-2) +	4.3426e+0 (1.05e+0) −	3.4251e+0 (3.17e-1)
	8	3.0897e+1 (2.64e+0) −	1.2151e+2 (5.81e+1) −	1.0524e+2 (2.13e+0) −	1.5204e+1 (1.22e+0) +	5.1181e+1 (1.68e+1) −	2.7249e+1 (2.81e+0)
	10	1.0707e+2 (4.98e+0) ≈	2.9081e+2 (1.69e+2) −	4.3406e+2 (1.45e+1) −	6.1558e+1 (6.09e+0) +	1.8393e+2 (3.68e+1) −	1.0643e+2 (1.31e+1)
	15	4.0855e+3 (3.57e+2) −	1.1013e+4 (7.00e+3) −	1.8063e+4 (9.75e+2) −	1.7209e+3 (4.57e+2) +	7.4301e+3 (1.97e+3) −	2.8858e+3 (4.41e+2)
MaF5	3	4.2436e-1 (4.32e-1) +	4.9683e-1 (5.18e-2) +	2.9656e-1 (1.27e-6) +	2.9692e-1 (2.26e-1) +	3.7757e-1 (4.48e-1) +	8.2615e-1 (1.25e+0)
	5	2.4516e+0 (2.39e-1) −	2.4032e+0 (4.52e-1) −	8.5678e+0 (4.39e-1) −	3.1800e+0 (1.88e+0) −	2.5450e+0 (6.60e-1) −	1.9815e+0 (2.53e-2)
	8	1.9099e+1 (5.80e+0) −	1.4691e+1 (5.69e-1) −	8.1831e+1 (9.69e-1) −	2.2021e+1 (9.91e+0) −	2.2524e+1 (1.47e+0) −	1.4087e+1 (4.41e-1)
	10	8.0835e+1 (4.16e+0) −	5.0927e+1 (2.09e+0) −	3.0062e+2 (8.92e-1) −	7.6587e+1 (3.07e+1) −	1.0506e+2 (5.14e+0) −	4.8600e+1 (1.51e+0)
	15	1.9215e+3 (6.48e+2) −	1.3254e+3 (7.47e+1) +	7.3232e+3 (4.41e-1) −	2.2803e+3 (6.17e+2) −	3.5063e+3 (8.68e+2) −	1.3914e+3 (1.09e+2)
MaF6	3	1.3383e-2 (1.62e-3) −	5.2878e-1 (1.88e-1) −	3.3931e-2 (8.08e-8) −	5.6816e-3 (4.32e-5) −	3.6291e-2 (1.47e-3) −	5.1058e-3 (1.50e-4)
	5	4.2022e-2 (1.25e-2) −	3.7179e-1 (1.58e-1) −	8.6863e-2 (1.67e-1) −	3.4073e-3 (5.29e-5) +	1.2373e-1 (1.13e-1) −	4.8320e-3 (2.84e-4)
	8	1.2677e-1 (1.37e-1) −	4.5390e-1 (1.82e-1) −	1.1210e-1 (2.01e-1) −	2.7682e-3 (2.41e-5) +	9.0159e-2 (1.89e-2) −	4.1247e-3 (1.76e-4)
	10	2.9299e-1 (1.01e-1) −	4.3810e-1 (2.03e-1) −	2.5944e-1 (2.95e-1) −	2.1517e-3 (3.66e-5) −	1.2859e-1 (1.80e-2) −	2.0287e-3 (1.27e-5)
	15	4.0474e-1 (1.52e-1) ≈	3.3408e-1 (2.05e-1) ≈	1.3548e-1 (1.80e-1) +	2.3865e-3 (6.82e-5) +	2.5742e-1 (2.04e-1) +	3.7707e-2 (1.89e-2)
MaF7	3	8.6049e-2 (5.49e-2) +	1.9461e-1 (2.62e-2) −	1.9575e-1 (1.64e-1) −	7.9333e-2 (1.23e-2) +	1.0539e-1 (8.23e-4) −	8.8617e-2 (7.17e-2)
	5	3.3264e-1 (1.10e-2) −	4.5258e-1 (3.84e-2) −	6.8561e-1 (2.56e-2) −	4.1316e-1 (8.34e-2) −	5.6196e-1 (1.02e-2) −	2.8465e-1 (4.68e-3)
	8	7.6957e-1 (2.50e-2) −	1.0287e+0 (3.00e-1) −	1.8426e+0 (2.19e-1) −	8.4817e-1 (4.06e-2) −	1.9195e+0 (4.94e-2) −	6.8172e-1 (9.27e-3)
	10	1.0444e+0 (6.82e-2) −	1.6036e+0 (4.05e-1) −	2.1193e+0 (3.68e-1) −	1.2006e+0 (3.56e-2) −	2.8678e+0 (2.31e-1) −	8.1744e-1 (3.75e-2)
	15	4.5804e+0 (9.89e-1) −	1.1023e+1 (8.80e-1) −	3.8990e+0 (9.81e-1) −	1.6343e+0 (7.55e-2) +	4.1675e+0 (8.57e-1) −	2.1967e+0 (3.79e-1)
MaF8	3	1.0131e-1 (7.88e-3) −	3.7158e-1 (4.00e-2) −	1.1465e-1 (4.94e-5) −	7.0547e-2 (1.35e-3) −	1.4377e-1 (1.37e-2) −	6.8483e-2 (1.17e-3)
	5	1.8252e-1 (1.37e-2) −	5.6048e-1 (5.91e-2) −	2.0850e-1 (3.75e-3) −	1.0479e-1 (1.32e-3) +	4.1256e-1 (5.42e-2) −	1.2315e-1 (6.51e-3)
	8	3.6698e-1 (5.06e-2) −	7.1310e-1 (7.27e-2) −	6.6064e-1 (8.00e-3) −	1.7350e-1 (3.39e-2) −	7.7940e-1 (1.23e-1) −	1.6590e-1 (1.12e-2)
	10	4.4478e-1 (6.69e-2) −	7.3004e-1 (8.82e-2) −	9.2174e-1 (5.72e-3) −	2.9053e-1 (1.01e-1) −	7.9616e-1 (1.03e-1) −	1.3886e-1 (9.72e-3)
	15	4.0975e-1 (9.40e-2) −	8.6828e-1 (9.71e-2) −	1.3554e+0 (3.56e-3) −	3.5783e-1 (1.53e-1) −	1.2885e+0 (2.27e-1) −	1.5975e-1 (3.51e-3)
MaF9	3	6.3113e-2 (1.30e-3) +	2.4999e-1 (3.36e-2) −	6.2079e-2 (2.63e-4) +	7.0210e-2 (8.22e-3) −	6.1986e-2 (2.66e-6) +	6.4003e-2 (4.50e-4)
	5	6.0483e-1 (1.66e-1) −	6.8561e-1 (2.15e-1) −	1.2162e-1 (1.10e-4) +	1.0504e-1 (4.31e-3) +	2.9914e-1 (3.31e-2) ≈	3.3176e-1 (2.11e-1)
	8	5.1993e-1 (9.93e-2) −	9.0469e-1 (2.40e-1) −	2.6707e-1 (1.06e-4) −	4.1192e-1 (7.71e-2) −	6.3250e-1 (7.75e-2) −	2.4647e-1 (6.20e-2)
	10	6.3030e-1 (1.63e-1) −	7.2716e-1 (1.59e-1) −	7.8542e-1 (1.21e+0) −	1.1805e+0 (2.34e-1) −	8.1237e-1 (1.97e-1) −	3.6457e-1 (4.42e-2)
	15	3.9136e-1 (7.56e-2) −	8.3991e-1 (2.69e-1) −	2.7296e+0 (4.08e+0) −	7.3927e-1 (2.05e+0) −	1.5008e+0 (4.07e-1) −	2.7198e-1 (7.25e-2)
MaF10	3	1.6236e-1 (4.16e-3) +	5.2758e-1 (1.41e-1) −	2.1616e-1 (1.80e-3) −	3.1919e-1 (7.73e-2) −	1.5118e-1 (4.03e-3) +	1.7297e-1 (2.04e-2)
	5	4.3994e-1 (4.96e-3) +	1.1170e+0 (1.39e-1) −	7.1472e-1 (1.09e-2) −	5.8987e-1 (3.49e-2) −	4.2405e-1 (3.73e-3) +	4.7299e-1 (3.53e-2)
	8	8.7805e-1 (2.11e-2) +	1.7008e+0 (2.05e-1) −	1.4606e+0 (5.23e-2) −	1.0991e+0 (2.65e-2) −	9.0566e-1 (1.53e-2) +	1.0005e+0 (2.63e-2)
	10	1.0226e+0 (1.85e-2) +	1.7422e+0 (1.19e-1) −	1.8146e+0 (1.67e-1) −	1.2664e+0 (9.25e-2) −	1.0317e+0 (1.23e-2) +	1.1278e+0 (4.02e-2)
	15	1.4449e+0 (3.05e-2) +	2.2007e+0 (1.48e-1) −	2.4171e+0 (2.04e-1) −	2.0420e+0 (1.11e-1) −	1.6604e+0 (8.36e-2) ≈	1.6723e+0 (4.56e-2)
MaF11	3	1.7205e-1 (4.55e-3) +	2.7427e-1 (2.94e-2) −	2.1017e-1 (1.97e-2) −	1.8309e-1 (5.10e-3) −	1.7361e-1 (2.93e-3) +	1.7504e-1 (2.71e-3)
	5	4.7382e-1 (4.56e-3) −	6.5487e-1 (4.25e-2) −	6.9796e-1 (1.83e-2) −	7.3171e-1 (1.30e-1) −	4.5655e-1 (5.92e-3) +	4.7037e-1 (1.71e-2)
	8	1.3520e+0 (1.41e-1) −	1.2356e+0 (9.80e-2) −	1.6396e+0 (4.88e-3) −	1.2933e+0 (2.39e-1) −	9.6223e-1 (3.05e-2) +	1.0983e+0 (2.52e-2)
	10	1.3910e+0 (1.01e-1) −	1.3199e+0 (9.03e-2) −	1.8687e+0 (1.53e-2) −	1.5147e+0 (3.94e-1) −	1.2661e+0 (2.73e-2) −	1.1801e+0 (2.52e-2)
	15	1.6249e+0 (1.07e-1) +	3.1290e+0 (1.38e+0) −	2.3744e+0 (9.01e-3) −	1.7948e+0 (4.89e-2) −	1.7904e+0 (1.03e-1) −	1.7542e+0 (4.37e-2)
MaF12	3	2.4588e-1 (2.45e-2) −	6.5508e-1 (1.09e-1) −	2.5162e-1 (2.91e-2) −	2.2552e-1 (3.09e-2) −	2.3127e-1 (7.51e-4) −	2.2214e-1 (2.56e-3)
	5	1.1427e+0 (9.47e-3) −	1.6939e+0 (2.17e-1) −	1.4395e+0 (8.11e-2) −	1.0616e+0 (1.08e-2) +	1.1619e+0 (1.74e-3) −	1.0888e+0 (8.84e-3)
	8	3.0786e+0 (6.64e-2) −	3.2041e+0 (1.35e-1) −	6.0781e+0 (2.69e-1) −	2.8729e+0 (2.73e-2) +	2.9364e+0 (1.04e-2) −	2.9314e+0 (2.21e-2)
	10	4.4948e+0 (9.91e-2) −	4.4159e+0 (2.19e-1) −	8.6708e+0 (3.66e-1) −	4.1314e+0 (3.73e-2) −	4.6649e+0 (2.49e-2) −	4.0286e+0 (2.50e-2)
	15	8.0693e+0 (9.38e-2) −	9.6307e+0 (9.83e-1) −	1.4829e+1 (8.41e-1) −	7.5154e+0 (9.48e-2) −	6.8419e+0 (1.26e-1) +	7.1828e+0 (7.24e-2)
MaF13	3	9.9419e-2 (7.94e-3) −	3.6225e-1 (9.39e-2) −	8.8205e-2 (8.79e-3) −	7.1682e-2 (6.96e-3) −	6.8455e-2 (3.82e-3) −	6.1257e-2 (1.68e-3)
	5	2.6171e-1 (3.48e-2) −	4.3475e-1 (8.78e-2) −	1.7680e-1 (5.30e-3) −	1.5878e-1 (1.63e-2) −	1.1365e+0 (4.20e+0) −	1.3022e-1 (2.77e-2)
	8	2.7992e-1 (2.70e-2) −	4.4452e-1 (8.89e-2) −	7.6915e-1 (3.86e-2) −	2.2109e-1 (5.24e-2) −	7.3375e-1 (2.27e-1) −	1.2686e-1 (1.55e-2)
	10	2.4950e-1 (2.01e-2) −	3.9661e-1 (9.41e-2) −	1.1270e+0 (6.17e-2) −	1.9078e-1 (1.93e-2) −	1.0870e+0 (6.00e-1) −	1.3151e-1 (2.87e-2)
	15	2.9286e-1 (3.76e-2) −	4.4511e-1 (1.19e-1) −	1.3693e+0 (1.50e-1) −	1.9257e-1 (2.38e-2) −	1.3316e+0 (7.79e-1) −	1.3599e-1 (2.13e-2)

that the output populations of algorithms *A-NSGA-III* and *RVEA* are far from the PF. Although the two comparison algorithms, i.e., *MOEA/D* and *MOEA/D-AM2M*, have good convergence, their diversity is a bit poor. Among the six

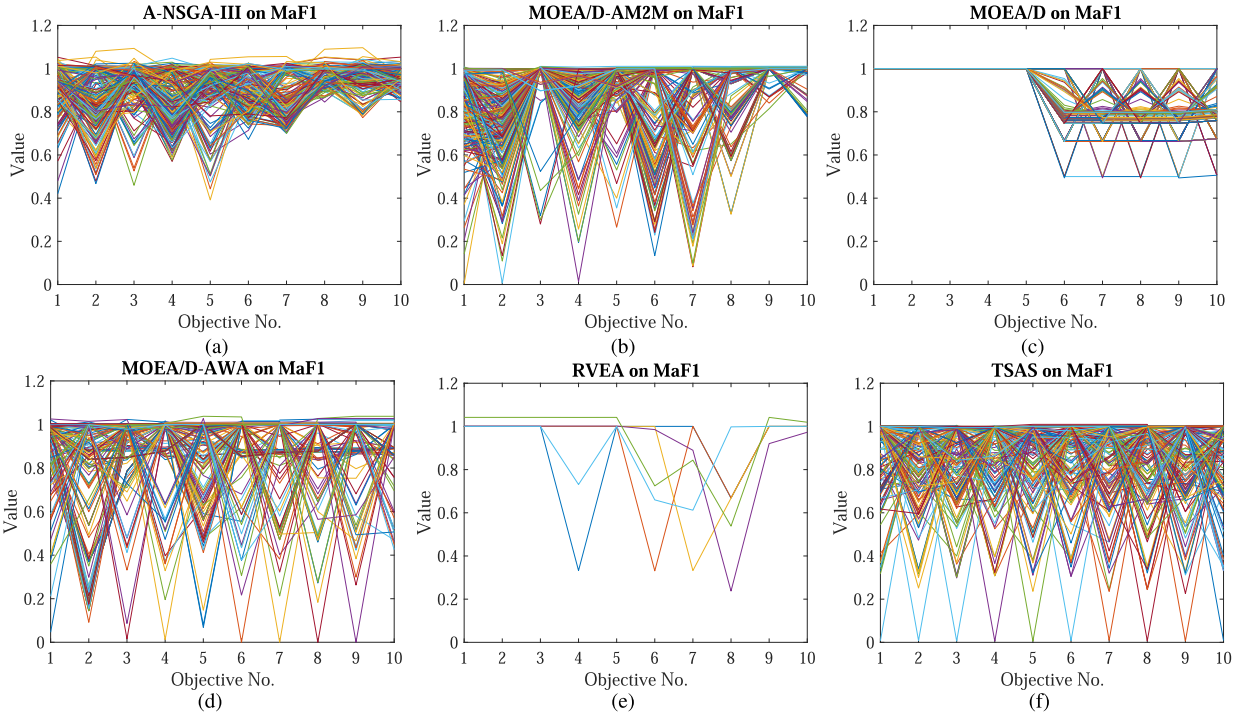

FIGURE 2. Performance comparison of the six algorithms on 10-objective MaF1.

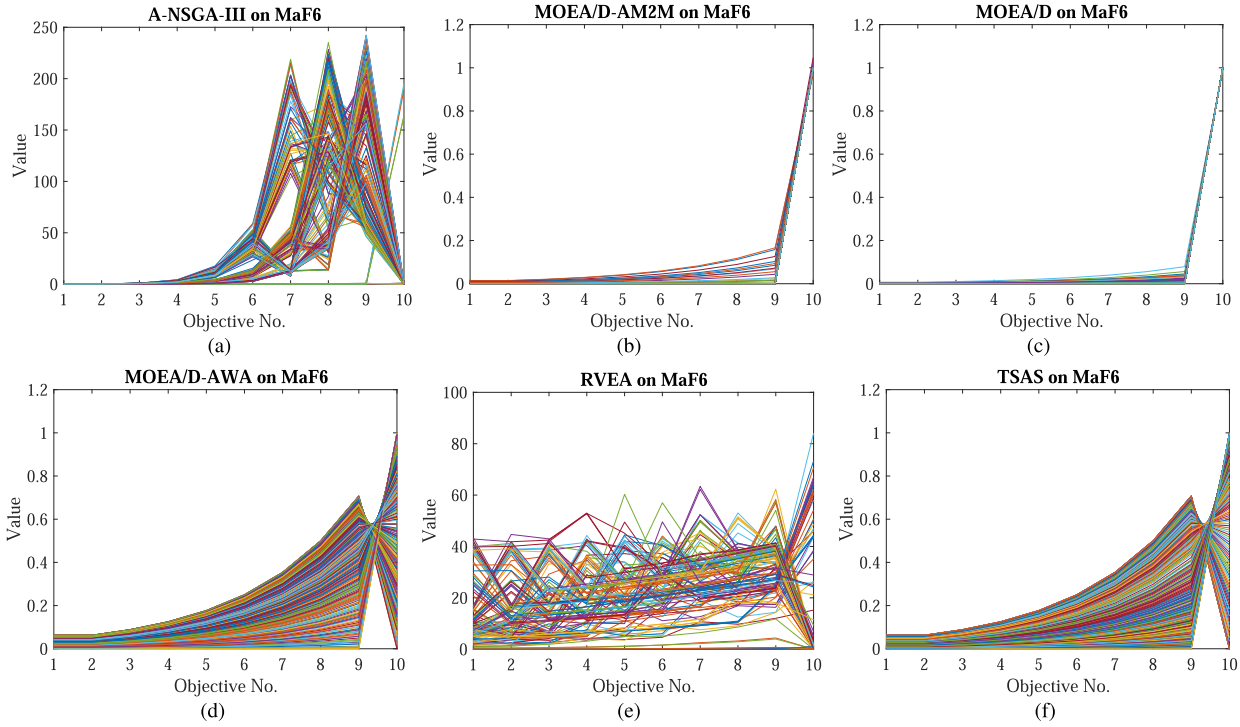

FIGURE 3. Performance comparison of the six algorithms on 10-objective MaF6.

comparison algorithms, the *MOEA/D-AWA* shows the best performance in both convergence and diversity. Comparing Fig. 3(d) and Fig. 3(f), we can see that the distribution of output population of the proposed *TSAS* is similar to that of *MOEA/D-AWA*. According to the IGD values in Table 2, we can know that the proposed *TSAS* is better than *MOEA/D-AWA* in terms of convergence and diversity. The comparison results in Fig. 3 illustrate the competitive performance of

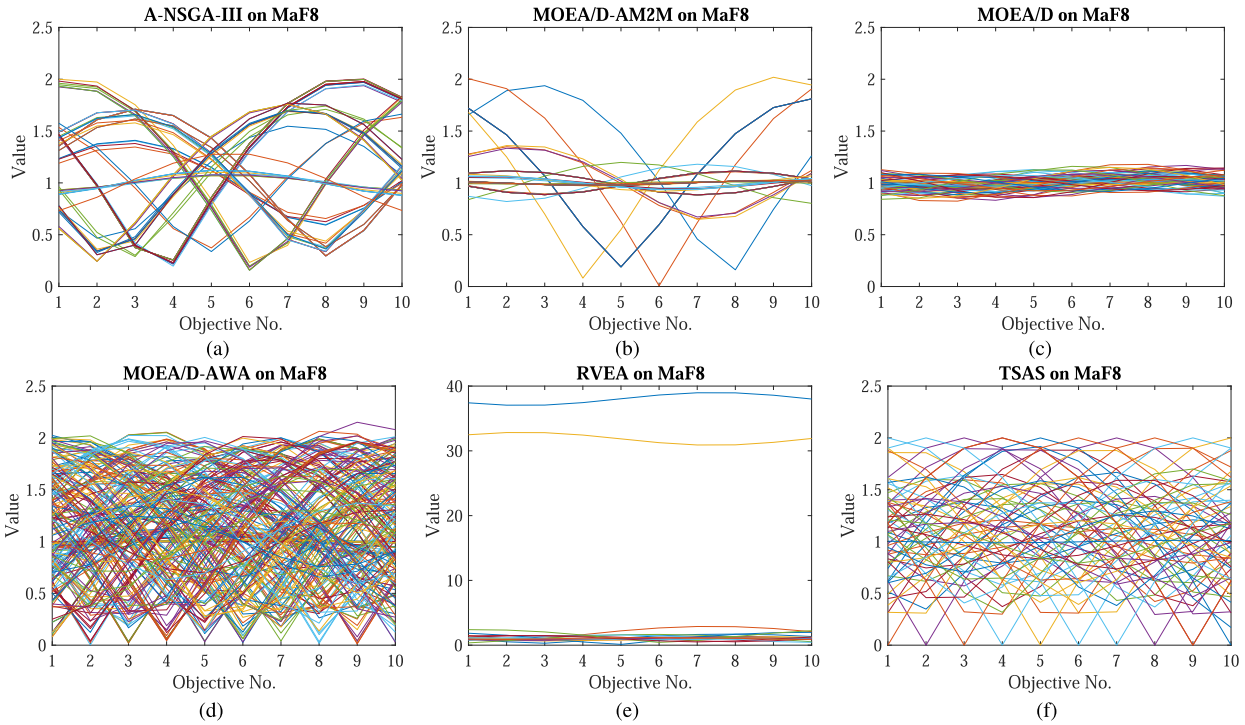

FIGURE 4. Performance comparison of the six algorithms on 10-objective MaF8.

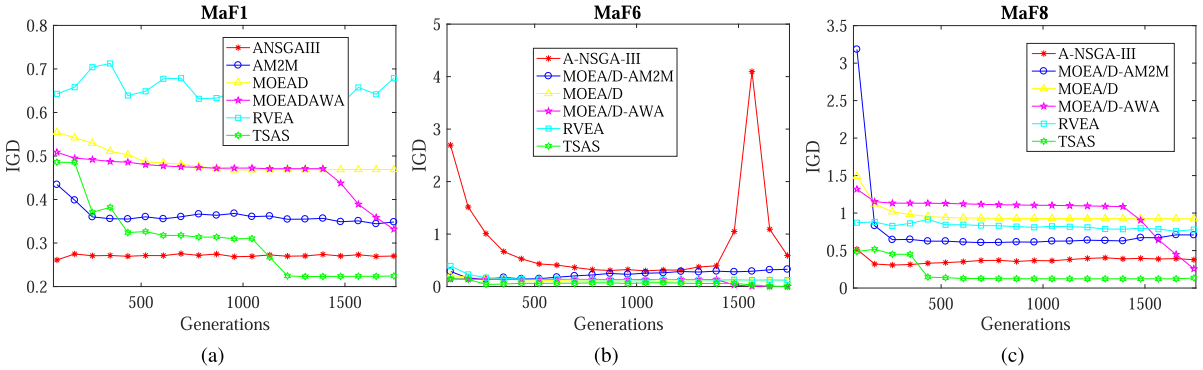

FIGURE 5. Downward trend of IGD values of the six algorithms on 10-objective MaF1, MaF6, and MaF8.

the proposed *TSAS* in handling MaOPs with degenerate PF shapes.

For benchmark function MaF8, the values of PF in each dimension approximately range from 0 to 2. The comparison results in Fig. 4 show that the proposed *TSAS* behaves competitive performance in both the convergence and diversity.

In addition, the downward trends of the IGD values of the six algorithms on 10-objective MaF1, MaF6, and MaF8 are given in Fig. 5. For the 10-objective MaF1, as shown in Fig. 5(a), the IGD of algorithm *TSAS* declines much faster than algorithms *MOEA/D-AM2M*, *MOEA/D*, *MOEA/D-AWA*, and *RVEA*. In the early stage, the convergence speed of the *A-NSGA-III* is obviously faster than that of the proposed *TSAS*, but the IGD value of the *A-NSGA-III* does not change after that. After 1,200 generations, the proposed *TSAS* generates lower IGD values than *A-NSGA-III*. This can be attributed to the two-stage adjustment strategy in *TSAS* to improve the diversity of the population. Also, from Fig. 5(b) and (c), we can observe the similar advantage of *TSAS* in descent speed of IGD values.

C. COMPARISON RESULTS ON REAL-WORLD APPLICATIONS

To test the effectiveness of the proposed *TSAS* in solving real-world many-objective optimization problems. Six test instances from agile satellite task planning [58] are employed to compare the performance of the *TSAS* with the five comparison algorithms. Since the real PFs of these real-world test instances are not available, the *IGD* metric cannot be

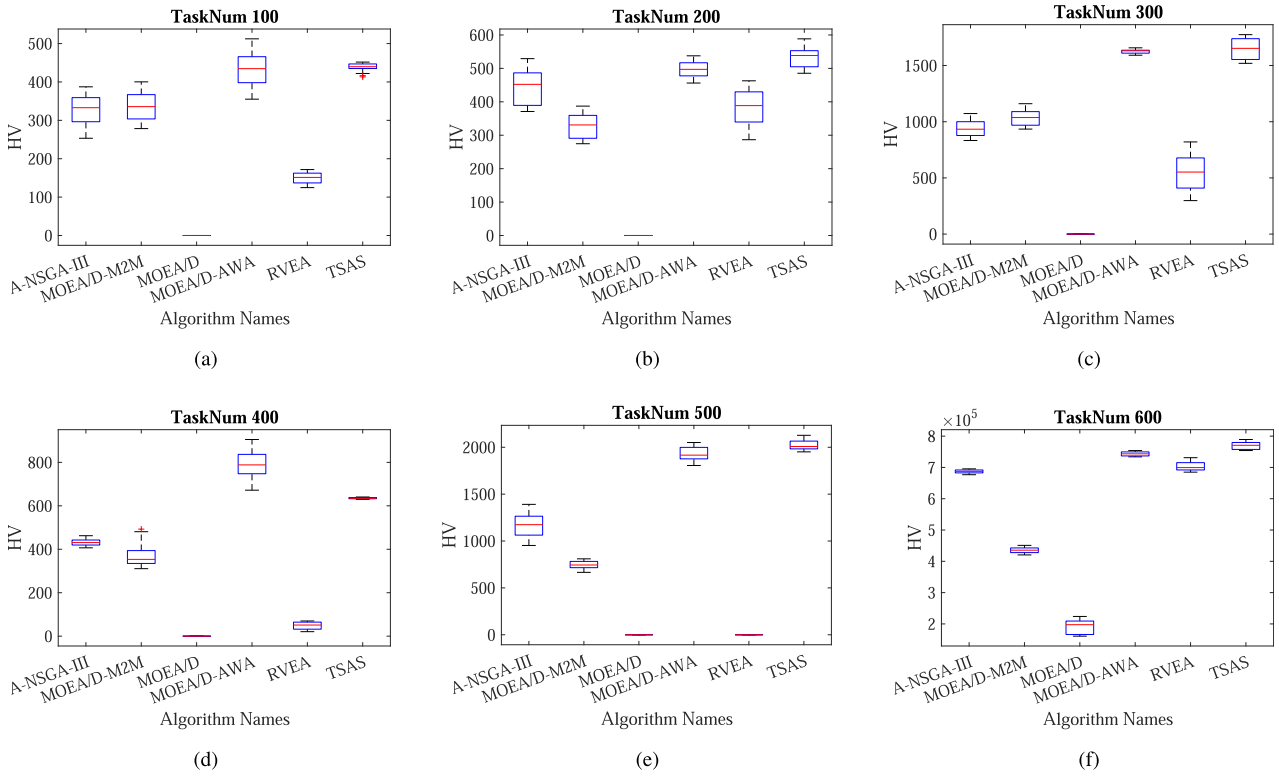

FIGURE 6. HV values of the six algorithms on 5-objective agile satellite task planning.

calculated directly. In this subsection, we use the *HV* metric for comparison.

In the experiments, the test instances from agile satellite task planning involves simultaneously optimizing the following five objectives, i.e., profit, ratio of completed tasks, energy consumption, balance, and timeliness. The value range of different objectives varies greatly. For example, the ratio of completed tasks ranges from 0 to 1, while the energy consumption may range from 0 to 10^5 or even greater. Thus, these test instances are representative MaOP with irregular PFs. For more details about the model of these many-objective agile satellite task planning, please refer to the work [58].

In the experiments, we assume that there are 36 satellites evenly distributed on 12 orbital planes. That is, the angle between two orbital planes is 30°, and the angle between two satellites in one plane is 120°. The height of each satellite is about 975 km, and the orbital period is about 6275 seconds.

The location of satellite observation tasks is randomly set in the area between 60° south latitude and 60° north latitude. For each task, its observation duration is a random number between 5 and 25 seconds, and its priority is a random integer between 1 and 10. The visible time windows between tasks and satellites are calculated by Systems Tool Kit,[2] and the simulation time is 8 hours.

The number of tasks is denoted as *TaskNum*, and we change the parameter as one of {100, 200, 300, 400, 500, 600} to construct 6 test instances. To set the reference point for calculating HV values, 5000 solutions are randomly generated, and the maximum value of each objective is used to construct the reference point. On the context of the six test instances, the HV values obtained by the six algorithms are plotted as boxplots, which are shown in Fig. 6.

As shown in Fig. 6, the HV values of some algorithms are zero. The reason is that all the output solutions of these algorithms cannot dominate the reference point. From the Fig. 6, we can observe that the proposed *TSAS* outperforms all the five comparison algorithm on 5 out of the 6 test instances with respect to the mean HV values. When the number of satellite observation task is 400, the mean HV value generated by the proposal is smaller than the algorithm *MOEA/D-AWA*, but still larger than the other four comparison algorithms. The comparison results demonstrate that the proposal has competitive performance in solving real-world many-objective optimization problems.

V. CONCLUSION AND FUTURE WORK

This work mainly focuses on the many-objective optimization problems having irregular PF shapes, such as inverted, degenerate, discontinuous, and badly scaled. To resolve this challenging task, we design a two-stage adjustment strategy to improve the performance of the space division based MOEAs. This strategy first strives to approach the boundaries of objective spaces having solutions, and then adds exploratory reference vectors to explore the sparse areas where the solutions may exist. To assess the effectiveness

[2] https://www.agi.com/products/stk/

of the proposed *TSAS*, on the basis of three metrics, extensive comparison experiments are carried out. The numerical results demonstrate the advantages of *TSAS* in solving MaOPs, especially the ones with complicated PF shapes.

How to solve MaOPs with complex PSs is also an ongoing challenge in the multi-objective evolutionary optimization community. Up to now, there are few MOEAs being able to solve these MaOPs very well. Thus, it is an interesting and meaningful research direction. In addition, with the increase in decision variables, the objective spaces of MaOPs often explode exponentially. This phenomenon seriously challenges the existing MOEAs. Developing efficient MOEAs for MaOPs with large-scale decision variables is also a future direction.

REFERENCES

[1] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 97–111, Apr. 2017.

[2] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2019, doi: 10.1109/tsc.2018.2866421.

[3] Y. Tian, R. Cheng, X. Zhang, M. Li, and Y. Jin, "Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems," *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 61–74, Aug. 2019.

[4] G. Sun, B. Yang, Z. Yang, and G. Xu, "An adaptive differential evolution with combined strategy for global numerical optimization," *Soft Comput.*, vol. 24, pp. 6277–6296, Mar. 2019.

[5] J. Liu, C. Wu, G. Wu, and X. Wang, "A novel differential search algorithm and applications for structure design," *Appl. Math. Comput.*, vol. 268, pp. 246–269, Oct. 2015.

[6] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[7] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107327.

[8] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–35, 2015.

[9] B. Cao, J. Zhao, Z. Lv, Y. Gu, P. Yang, and S. K. Halgamuge, "Multiobjective evolution of fuzzy rough neural network via distributed parallelism for stock prediction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 5, pp. 939–952, May 2020.

[10] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

[11] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[13] K. Shang and H. Ishibuchi, "A new hypervolume-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 839–852, Oct. 2020.

[14] Q. Xu, Z. Xu, and T. Ma, "A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions," *IEEE Access*, vol. 8, pp. 41588–41614, 2020.

[15] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[16] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[17] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[18] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Mar. 12, 2019, doi: 10.1109/tsmc.2019.2898456.

[19] H. Bai, J. Zheng, G. Yu, S. Yang, and J. Zou, "A Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation," *Inf. Sci.*, vol. 478, pp. 186–207, Apr. 2019.

[20] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

[21] X. Ma, Y. Yu, X. Li, Y. Qi, and Z. Zhu, "A survey of weight vector adjustment methods for decomposition based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 634–649, Aug. 2020.

[22] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, p. 231, 2014.

[23] J. Li, G. Chen, M. Li, and H. Chen, "An adaptative reference vector based evolutionary algorithm for many-objective optimization," *IEEE Access*, vol. 7, pp. 80506–80518, 2019.

[24] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proc. Annu. Meeting North Amer. Fuzzy Inf. Process. Soc. (NAFIPS-FLINT)*, Jun. 2002, pp. 233–238.

[25] S. Zapotecas-Martinez, C. A. Coello Coello, H. E. Aguirre, and K. Tanaka, "A review of features and limitations of existing scalable multiobjective test suites," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 130–142, Feb. 2019.

[26] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, "Hyperplane assisted evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3367–3380, Jul. 2020, doi: 10.1109/tcyb.2019.2899225.

[27] V. Palakonda and R. Mallipeddi, "An evolutionary algorithm for multi and many-objective optimization with adaptive mating and environmental selection," *IEEE Access*, vol. 8, pp. 82781–82796, 2020.

[28] J. H. Zheng, Y. N. Kou, Z. X. Jing, and Q. H. Wu, "Towards many-objective optimization: Objective analysis, multi-objective optimization and decision-making," *IEEE Access*, vol. 7, pp. 93742–93751, 2019.

[29] H. Chen, R. Cheng, W. Pedrycz, and Y. Jin, "Solving many-objective optimization problems via multistage evolutionary search," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 7, 2019, doi: 10.1109/TSMC.2019.2930737.

[30] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[31] K. Li, R. Chen, G. Min, and X. Yao, "Integration of preferences in decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3359–3370, Dec. 2018.

[32] M. K. Tomczyk and M. Kadzinski, "Decomposition-based interactive evolutionary algorithm for multiple objective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 320–334, Apr. 2020.

[33] R. Liu, R. Wang, W. Feng, J. Huang, and L. Jiao, "Interactive reference region based multi-objective evolutionary algorithm through decomposition," *IEEE Access*, vol. 4, pp. 7331–7346, 2016.

[34] K. Li, R. Chen, D. Savic, and X. Yao, "Interactive decomposition multiobjective optimization via progressively learned value functions," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 5, pp. 849–860, May 2019.

[35] G. Yu, J. Zheng, R. Shen, and M. Li, "Decomposing the user-preference in multiobjective optimization," *Soft Comput.*, vol. 20, no. 10, pp. 4005–4021, Oct. 2016.

[36] H. Li and D. Landa-Silva, "Evolutionary multi-objective simulated annealing with adaptive and competitive search direction," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 3311–3318.

[37] F. Gu, Y.-M. Cheung, H.-L. Liu, and Z. Lin, "A parameterless decomposition-based evolutionary multi-objective algorithm," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Mar. 2018, pp. 842–845.

[38] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. L. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 572–586, Aug. 2019.

[39] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.

[40] F. Gu, H. L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput., Inf. Control*, vol. 8, no. 5, pp. 3677–3688, 2012.
[41] M. Pilat and R. Neruda, "General tuning of weights in MOEA/D," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 965–972.
[42] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433–448, Jun. 2018.
[43] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376–390, Jun. 2019.
[44] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, "Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 439–453, Jun. 2020.
[45] D. Han, W. Du, W. Du, Y. Jin, and C. Wu, "An adaptive decomposition-based evolutionary algorithm for many-objective optimization," *Inf. Sci.*, vol. 491, pp. 204–222, Jul. 2019.
[46] M. Elarbi, S. Bechikh, C. A. C. Coello, M. Makhlouf, and L. B. Said, "Approximating complex Pareto fronts with predefined normal-boundary intersection directions," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 809–823, Oct. 2020.
[47] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
[48] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "A new two-stage evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 748–761, Oct. 2019.
[49] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
[50] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[51] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1510–1522, Jun. 2017.
[52] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Comput.*, vol. 14, no. 3, p. 193, 2010.
[53] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, Mar. 2017.
[54] N. Le Chau, T.-P. Dao, and V. A. Dang, "An efficient hybrid approach of improved adaptive neural fuzzy inference system and teaching learning-based optimization for design optimization of a jet pump-based thermoacoustic-stirling heat engine," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7259–7273, Jun. 2020.
[55] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput.*, vol. 13, no. 10, p. 959, 2009.
[56] N. Le Chau, N. T. Tran, and T.-P. Dao, "A multi-response optimal design of bistable compliant mechanism using efficient approach of desirability, fuzzy logic, ANFIS and LAPO algorithm," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106486.
[57] M. Li, L. Zhen, and X. Yao, "How to read many-objective solution sets in parallel coordinates [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 88–100, Nov. 2017.
[58] L. Li, H. Chen, J. Li, N. Jing, and M. Emmerich, "Preference-based evolutionary many-objective optimization for agile satellite mission planning," *IEEE Access*, vol. 6, pp. 40963–40978, 2018.

WEN ZHONG received the M.S. degree from the College of Information and System Management, National University of Defense Technology, China, in 2011. He is currently pursuing the Ph.D. degree with the College of Systems Engineering, National University of Defense Technology. His research interests include multi-objective optimization, task scheduling, complex networks, and blockchain consensus mechanism.

XUEJUN HU received the bachelor's and Ph.D. degrees in management from the Huazhong University of Science and Technology (HUST), in 2011 and June 2016, respectively. The title of the Ph.D. dissertation was "The Research on Project Schedule Control Methods Based on Buffer Management." She is currently an Associate Professor with the Business School, Hunan University, Changsha, China. She teaches several courses on operations management, logistics, and supply chain management and project management for undergraduate students, respectively. Her research interests include multi-objective optimization, project scheduling and project management, satellite scheduling, and uncertainty programming.

FA LU received the B.S. and M.S. degrees in management science and engineering from the College of Information and System Management, National University of Defense Technology, China, in 2012 and 2014, respectively. He is currently working with the Joint Service Defense College, PLA National Defense University. He has authored or coauthored around ten articles in core journals and conferences. His research interests include information system design and application, logistics information platform, and evolutionary computation.

JIANJIANG WANG received the B.S. degree in information systems and the Ph.D. degree in operations research from the National University of Defense Technology (NUDT), Changsha, China, in 2009 and 2015, respectively. He is currently an Assistant Professor with the College of Systems Engineering, NUDT. He has published a dozen of peer-reviewed journal articles in *Computers & Operations Research* (COR), IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), *Information Systems Journal* (ISJ), *Journal of Computer and System Sciences* (JCSS), and *Flexible Services and Manufacturing Journal* (FSMJ). His research interests include combinational optimization, multi-objective evolutionary optimization, satellite scheduling, and project scheduling.

XIAOLU LIU received the B.E. degree in system engineering and the Ph.D. degree in management science and engineering from the National University of Defense Technology, China, in 2006 and 2011, respectively. She is currently an Associate Professor with the College of System Engineering, National University of Defense Technology. Her research interests include artificial intelligence and metaheuristics, solving distribution management, and satellite scheduling problems.

YINGWU CHEN received the B.S. degree in automation, the M.S. degree in system engineering, and the Ph.D. degree in engineering from the National University of Defense Technology (NUDT), Changsha, China, in 1984, 1987, and 1994, respectively.

He was a Lecturer from 1989 to 1994, and an Associate Professor from 1994 to 1999 with NUDT. Since 1999, he has been a Distinguished Professor and the Director of the Department of Management Science and Engineering, College of Information Systems and Management, NUDT, where he focuses on management theory and its applications. His current research interests include assistant decision-making systems for planning, decision-making systems for project evaluation, management decisions, and artificial intelligence.