IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 21, 2018, accepted July 17, 2018, date of publication July 23, 2018, date of current version August 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2859028

Preference-Based Evolutionary Many-Objective Optimization for Agile Satellite Mission Planning

LONGMEI LI[1], HAO CHEN[1], JUN LI[1], NING JING[1], AND MICHAEL EMMERICH[2]
[1]College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China
[2]Leiden Institute of Advanced Computer Science, Leiden University, 2333CA Leiden, The Netherlands

Corresponding author: Longmei Li (longmeili@nudt.edu.cn)

This work was supported by the National Science Foundation of China under Grant 61101184 and Grant 61174159.

ABSTRACT With the development of aerospace technologies, the mission planning of agile earth observation satellites has to consider several objectives simultaneously, such as profit, observation task number, image quality, resource balance, and observation timeliness. In this paper, a five-objective mixed-integer optimization problem is formulated for agile satellite mission planning. Preference-based multi-objective evolutionary algorithms, i.e., T-MOEA/D-TCH, T-MOEA/D-PBI, and T-NSGA-III are applied to solve the problem. Problem-specific coding and decoding approaches are proposed based on heuristic rules. Experiments have shown the advantage of integrating preferences in many-objective satellite mission planning. A comparative study is conducted with other state-of-the-art preference-based methods (T-NSGA-II, T-RVEA, and MOEA/D-c). Results have demonstrated that the proposed T-MOEA/D-TCH has the best performance with regard to IGD and elapsed runtime. An interactive framework is also proposed for the decision maker to adjust preferences during the search. We have exemplified that a more satisfactory solution could be gained through the interactive approach.

INDEX TERMS Preferences, evolutionary many-objective optimization, EOS mission planning, target region, MOEA/D.

I. INTRODUCTION

Earth Observation Satellites (EOSes) acquire photographs of the earth surface from orbit, using the onboard instruments. They play an important role in environmental monitoring, meteorology, map making and other fields. Agile EOS (AEOS), on which the camera can move around three axes (roll, pitch, yaw) [1], is addressed in this paper. Fig. 1 illustrates the difference bewteen non-agile EOS and AEOS. There are three candidate tasks to be observed. The access window decides when the satellite can take images of the target. Due to the flexibility of AEOS, its access window is much wider than needed. The observation window can slide within the access window, making the originally conflicting tasks (task 1 and task 2) compatible. As is shown in Fig. 1, AEOS can accomplish all the three tasks, while non-agile EOS can only observe two of them. The advancement of AEOS largely strengthens its capability, but increases complexity for the mission planning problem. It has to decide not only which targets to observe, but also when to start the observation.

Usually, the ground control center collects image requests from different users, and makes a mission plan (a sequence of observation actions) for the satellite to execute. Because of the NP-hard nature of the problem [2], heuristic and meta-heuristic approaches have been widely used by researchers to solve the problem, such as genetic algorithms [3], [4], ant colony optimization [5], [6], tabu search [7] and so on.

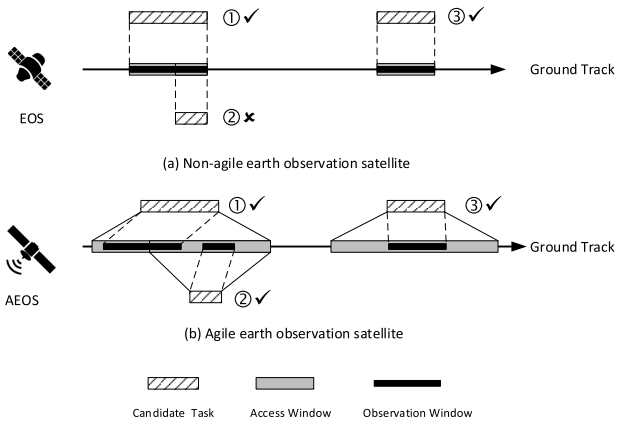

FIGURE 1. Comparison of agile and non-agile EOS.

The mission planning has to consider several objectives simultaneously, such as total profit, image quality, resource equilibrium and so on. From the perspective of image users, they require high-resolution images as early as possible. From the point of view of the satellite control agencies, they want to maximize the total profit and balance the usage of satellites. These objectives are in conflict to some extent. When modeling the problem as a Multi-objective Optimization Problem (MOP), utilizing multi-objective metaheuristics is popular. Researchers have adopted Nondominated Sorting Genetic Algorithm-II (NSGA-II) [8], [9], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [10], multi-objective local search [11], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [12], multiobjective scatter search [13] to investigate the problem.

The resolution of a MOP is a set of non-dominated solutions. However, only one solution will be chosen as a final plan. With the increase of objective numbers, obtaining the whole set of Pareto-optimal solutions will be difficult. What's more, selecting from a large number of candidate solutions is not a trivial task for the decision maker (DM).

In the last decade, integrating user preferences into multi-objective evolutionary algorithms has become prevalent. Preference information, provided by the DM, can be utilized to guide the search and obtain merely solutions that are of interest to the DM. As a result, the difficulty of many-objective optimization could be overcome to some extent, and the selection burden of the DM will be relieved. The preference information could be imported through reference point [14], achievement scalarizing function [15], Desirability function [16], [17], relative importance of objectives [18], preference polyhedron [19] and other approaches. According to when the preference is incorporated with optimization, methods can be classified into three categories: *a-priori* (preference before optimization), *interactive* (preference during optimization) and *a-posteriori* (preference after optimization). In this paper, we investigate both *a-priori* and *interactive* methods.

Recently, Li et al. proposed a preference-based multi-objective evolutionary algorithm (T-NSGA-II) for AEOS mission planning [20]. A target region in the objective space was utilized to express the preferences. Three objectives, i.e., total profit, averaged quality and timeliness are optimized simultaneously. However, with the development of aerospace technologies, more objectives should be involved in the planning, to achieve a plan satisfied by both image users and satellite control agencies. In this paper, we extend the previous work by adding two more objectives: the total number of the observed targets, resource usage equilibrium. This changes the problem from multi-objective to many-objective, which is more challenging to deal with. Usually, problems with more than four objectives are referred to as many-objective optimization problems (MaOPs) [21]. The performance of traditional Multi-Objective Evolutionary Algorithms (MOEAs) deteriorates seriously when handling MaOPs. We propose to use preference-based MOEAs for this many-objective AEOS mission planning. The new contributions of this paper include:

- A five-objective mixed-integer optimization problem is formulated considering the total profit, quantity, quality, resource balance and timeliness in AEOS mission planning.
- Problem-specific integer coding and decoding strategies, as well as variation operators, are devised for employing evolutionary algorithms to this real-world application.
- Three new preference-based evolutionary algorithms, i.e, T-MOEA/D-TCH, T-MOEA/D-PBI and T-NSGA-III are adopted to solve the problem. Results show that compared with the non-preference-based algorithms, the proposed algorithms can obtain preferred Pareto optimal solutions more effectively. T-MOEA/D-TCH has the best performance among the three.
- The proposed algorithms are compared with three state-of-the-art preference-based MOEAs, i.e., T-NSGA-II [22], T-RVEA [23] and MOEA/D-c [24]. Experiments show that T-MOEA/D-TCH outperform all the other algorithms considering the inverted generational distance within the target region.
- An interactive approach is proposed that the DM can adjust the preferences during the optimization process. We exemplified that a more reliable solution could be gained by interacting with the DM.

Section II introduces the AEOS mission planning problem in our context and gives the mathematical formulation. The new algorithms and coding/decoding strategies are proposed in Section III. Numerical experiments are designed and conducted in Section IV. Conclusions and future works are given in Section V.

II. PROBLEM DESCRIPTION AND FORMULATION

A. PROBLEM DESCRIPTION

To simplify the problem, we made the following assumptions:

- Once an observation is started, it cannot be interrupted.
- The onboard storage capacity is infinite, a satellite can observe as many targets as possible if it satisfies the operational constraints.
- The data transmission planning is not considered, we suppose all the images can be transmitted to the ground after observation.

Given a set of AEOS and a set of target on the earth surface, the mission planning is to decide which targets to observe and when to start the observation for a time period in the future. A typical 24 hours is adopted in our research. Since the access windows can be calculated based on the satellite orbit and the target's position, the problem boils down to selecting a subset of alternative access windows and setting the observation start time for each window. It is a complicated problem due to its many-objective, constrained and mixed-integer features.

1) MANY-OBJECTIVE

Five objectives are to be optimized simultaneously. The first objective is to maximize the total profit in a plan. Each target is assigned a profit, which can be considered as the price of the image. The number of requested target usually exceeds the capability of the satellites, the second objective is to maximize the total quantity of the observed targets. The image quality (resolution) is the third objective, which is related to the start time of each observation. From the viewpoint of satellite control agencies, use the satellites in a balanced way is good for a sustainable development. The fourth objective is to minimize the standard deviation of the overall observation time for every satellite. The fifth objective focuses on how fast the target is observed. In some urgent situations such as disaster rescue, timeliness is of great importance in AEOS mission planning. Each target could be observed for several times within 24 hours, the earlier it is observed, the higher timeliness it has.

2) CONSTRAINED

There are several constraints to be satisfied in the plan. (1) Each target should be observed at most once. (2) The observation start time should be within the corresponding access window. (3) There should be sufficient preparation time between two consecutive observations, during which the satellite adjusts the angle of the camera and gets ready for the next observation.

3) MIXED-INTEGER

The planning problem has to determine which access windows to choose (discrete variable) and set the start time for each chosen access window (continuous variable).

B. PROBLEM FORMULATION

The satellite set is denoted by $S = \{s_1, \ldots, s_M\}$, where M is the number of satellites. $\forall s_i \in S$, $s_i = (s_{ID}, pt)$, in which s_{ID} is the identification of the satellite and pt is the preparation time between two consecutive observations.

The target set is represented by $T = \{t_1, \ldots, t_N\}$, where N is the amount of targets. $\forall t_i \in T$, $t_i = (t_{ID}, pr, rt)$, meaning the identification of the target is t_{ID}, the profit of this target is pr and the requested time it should be observed is rt. rt is specified by the image user, indicating how much time the camera should spend in taking photos of this target, according to his/her needs. This is an input parameter deciding how long an observation lasts.

The access window set is $AW = \{aw_1, \ldots, aw_K\}$, where K denotes the quantity of the access windows. $\forall aw_i \in AW$, $aw_i = (s_{ID}, t_{ID}, st, et, select, est, qu, ti)$. It indicates that satellite s_{ID} can observe target t_{ID} from time st to time et. Whether to execute this observation depends on the Boolean variable $select$, when to start the observation is defined by the double variable est if $select$ is **true**. qu and ti are quality and timeliness metrics of this observation.

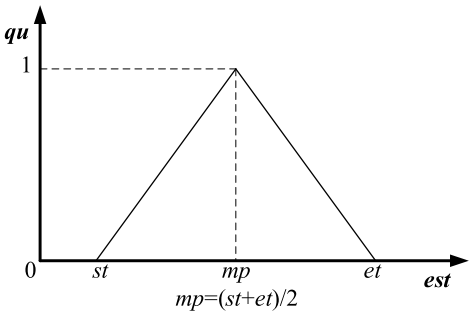

FIGURE 2. Calculation of quality metric *qu* of an observation.

The observed access window set includes all the chosen observations in AW: $OAW = \{aw_i \in AW | aw_i.select = \textbf{true}\}$.

The observed target set contains all the targets that have been observed by some satellite: $OT = \{t_i \in T | \exists aw_i \in OAW,\ s.t.\ aw_i.t_{ID} = t_i.t_{ID}\}$.

It should be noted that $select$ and est are decision variables of the optimization, all the other variables are either input data or intermediate variables.

The five objectives are formulated as follows.

- **Profit**: maximize the total profit of all the observed targets.
$$\sum_{ot \in OT} ot.pr \rightarrow \max$$
- **Quantity**: maximize the total number of the observed targets.
$$|OT| \rightarrow \max$$
- **Quality**: maximize the averaged image quality of all the observed targets.
$$\frac{1}{|OT|} \sum_{ot \in OT} ot.qu \rightarrow \max$$
where qu is calculated by st, et and est. In general, the image quality depends on the distance and angle from the camera to the target. Since the observation time is much shorter than the duration of one access window, the best qu can be obtained when est is at the middle point of the access window. The further est is away from the middle point, the worse qu will be. For simplicity we utilize a piecewise linear function in this paper, as shown in Fig. 2.
- **Balance**: minimize the standard deviation of the overall observation time for each satellite.
Supposing OT_{si} is the set of all the targets observed by satellite si, the overall observation time for satellite si is $t_{si} = \sum_{ot \in OT_{si}} ot.rt$.
$$\sqrt{\frac{\sum_{i=1}^{M} (t_{si} - t_{av})^2}{M-1}} \rightarrow \min$$
where t_{av} is the average value of t_{si} among different satellites.

- **Timeliness**: maximize the averaged timeliness metric [25] for all the observed targets.

$$\frac{1}{|OT|}\sum_{ot\in OT} ot.ti \rightarrow \max$$

where *ti* depends on the number of access windows that can observe this target and how early this access window is. Earlier observation corresponds to bigger *ti*.

The constraints considered in this paper is formulated as follows.

- **Temporality**: exact start time should be within the access window.

$$st \leq est \leq et$$

- **Uniqueness**: each target should be observed at most once.

$$\forall t_i \in OT,\ |\{aw_i \in OAW \,|\, aw_i.t_{ID} = t_i.t_{ID}\}| = 1$$

- **Transformation**: there must be enough preparation time for the transformation from one observation to the next one.

$$aw_2.est - (aw_1.est + aw_1.t_{ID}.du) \geq aw_1.s_{ID}.pt$$

where aw_1 and aw_2 are two consecutive access windows of the same satellite.

III. THE PROPOSED APPROACH

In the past decades, MOEAs have shown great success in solving multi-objective optimization problems [26]. Preference-based methods, which utilize the preference information offered by the DM to guide the search, can obtain merely preferred parts of the Pareto front (PF) and alleviate the selection burden of the DM [27].

In AEOS mission planning, preferences may vary according to the purpose of observation. For example, in global environmental supervision, quantity is more important than timeliness. However, in disaster monitoring and rescue, timeliness is the most critical objective. Utilizing this information can help to get a plan that is both optimal (in the sense of Pareto dominance) and preferred (in the sense of decision making).

The framework of the proposed approach is shown in Fig. 3, which consists of a DM and a target region-based MOEA (shaded part). A target region is provided by the DM to express the preferences, and guide the MOEA to Pareto optimal solutions complying with the preferences. The initial population is generated from the whole search space using specific encoding, then the target region takes effect in the variation process. Two well-known MOEAs, MOEA/D [28] and NSGA-III [29] are employed to embed the preference information. In the following, we will first introduce the target region, then, how it is integrated with MOEA/D and NSGA-III will be illustrated. Next, the problem-specific coding and decoding strategies will be elaborated. The last subsection presents the variation operators.

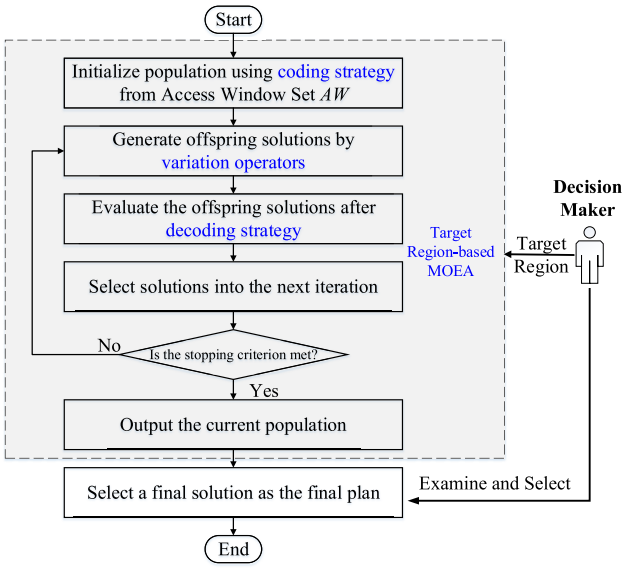

FIGURE 3. Framework of the proposed approach for AEOS mission planning.

A. TARGET REGION PREFERENCES

A target region is defined by preferred range of each objective, constituting a hypercube in the objective space. In an *M*-objective optimization problem, the preferred range of objective *i* is $[L_i, U_i]$, where L_i and U_i are the preferred lower bound and upper bound. The target region can be represented as

$$T = \prod_{i=1}^{M} [L_i, U_i]$$

The aim is to find well-spread Pareto optimal solutions within this region, if it has intersection with the true PF. When the target region does not intersect with the true PF, Pareto optimal solutions that are close to the target region are preferred.

Recently, two target region-based evolutionary algorithms were proposed for handling MaOPs [30]. The essence is a coordinate transformation from the original objective space to the target region space, which is shown in Fig. 4. The target region is regarded as a new coordinate system with the lower bound as the origin. New coordinates of a solution is calculated as

$$f_i'(\mathbf{x}) = f_i(x) - L_i, \quad i = 1, \cdots, M$$

where $\mathbf{L} = (L_1, \cdots, L_M)$ is the lower bound of the target region, $f_i'(\mathbf{x})$ and $f_i(\mathbf{x})$ are new coordinate and old coordinate of objective *i*, respectively.

The new algorithms, i.e., T-MOEA/D and T-NSGA-III (where T represents *Target*), have shown promising results in DTLZ benchmark problems [30]. We will introduce how the target region is incorporated to guide the search in the next two subsections.

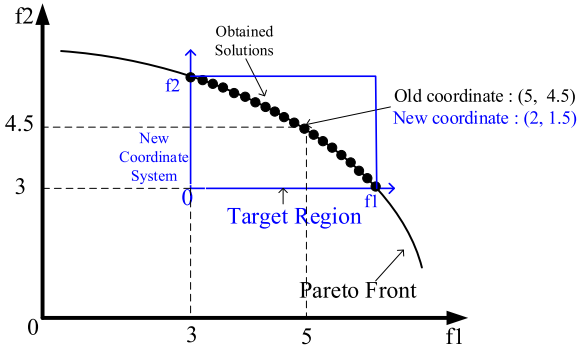

FIGURE 4. Illustration of coordinate transformation and the obtained solutions.

B. FROM MOEA/D TO T-MOEA/D

First introduced by Zhang and Li in 2007 [28], MOEA/D is now one of the most popular algorithms in Evolutionary Multi-objective Optimization (EMO) field. MOEA/D utilizes decomposition approaches to convert a MOP into several single-objective subproblems and solve them simultaneously in a collaborative manner.

There are two most commonly used decomposition methods: Tchebycheff Approach (TCH) and penalty-based boundary intersection Approach (PBI).

TCH:

$$\min\ g^{te}(\mathbf{x}\,|\lambda_i, \mathbf{z}^*) = \max_{1\le j\le M}\left\{\lambda_i^j\left|f_j(\mathbf{x}) - z_j^*\right|\right\}$$

PBI:

$$\begin{aligned}\min\ g^{pbi}(\mathbf{x}\,|\lambda_i, \mathbf{z}^*) &= d_1 + \theta d_2\\ d_1 &= \frac{\left\|(f(x)-\mathbf{z}^*)^T\boldsymbol{\lambda}_i\right\|}{\|\boldsymbol{\lambda}_i\|}\\ d_2 &= \left\|f(x) - (\mathbf{z}^* + d_1\frac{\boldsymbol{\lambda}_i}{\|\boldsymbol{\lambda}_i\|})\right\|\end{aligned}$$

where $\mathbf{z}^* = (z_1^*, \cdots, z_M^*)$ is the ideal point and $\boldsymbol{\lambda}_i = (\lambda_i^1, \cdots, \lambda_i^M)$ is a weight vector indicating a search direction. There are N uniformly distributed weight vectors, defining N sub-problems in total. N is also the population size.

T-MOEA/D is based on MOEA/D [31], the main differences for preference incorporation are underlined in Algorithm 1.

In the ideal point initialization step (line 3), the lower bound of the target region is used as the ideal point. It accelerates the evolution process to the target region. The fitness of each solution depends on its TCH or PBI function values, fitness assignment (line 9) is the core step to embed preferences. At first, an original fitness is calculated using lower bound of the target region as the ideal point. Then, if the solution is within the target region, a coordinate transformation is performed and its fitness will be updated using the new objective values and new ideal point. Otherwise, its fitness will be added a penalty value. This makes sure that solutions within the target region have better fitness than solutions outside the target region. In the solution update process (line 11), Pareto dominance is considered before fitness comparison. It aims at urging the solutions to reach the PF if the target region is set behind the true PF.

In this paper, both TCH and PBI approaches are employed, bringing in T-MOEA/D-TCH and T-MOEA/D-PBI respectively.

Algorithm 1 T-MOEA/D

Input: Target region T, maximal number of generations $t_{\max}$, population size N

Output: Final Population $P_{t_{\max}}$

1: $\vec{\lambda} = \text{GenerateUniformWeights}()$
2: $P_0 = \text{InitializePopulationAndNeigborhoods}(\vec{\lambda}\,)$
3: $\vec{z} = \underline{initilizeIdealPoint(T)}$ /* initialize ideal point using the lower bound of target region T */
4: $t = 0$
5: **while** $t < t_{\max}$ **do**
6: $\quad P_t = \text{MatingSelection}(P_t)$
7: $\quad Q_t = \text{OffspringGeneration}(P_t)$
8: $\quad \text{OffspringEvaluation}(Q_t)$
9: $\quad$ Calculate fitness considering target region T: $\underline{FitnessT(Q_t, T)}$
10: $\quad$ Update the new ideal point in the target region: $\vec{z}_n = \underline{UpdateIdealPoint(Q_t)}$
11: $\quad$ Update solutions considering Pareto dominance relation: $\underline{P_t = UpdateSolutions(P_t, Q_t)}$
12: $\quad t \leftarrow t + 1$
13: **end while**

C. FROM NSGA-III TO T-NSGA-III

NSGA-III was devised by Deb and Jain in 2014 [29]. Different from MOEA/D's explicit decomposition in the problem, NSGA-III implicitly decomposes the objective space using several well-spread reference points to ensure diversity. The framework of NSGA-III keeps the same as the well-known NSGA-II [32], but replaces the crowding distance with a niche-preservation selection. In this selection operator, NSGA-III associates each solution with the closest reference point. Solutions associated with less crowded reference point have a higher chance to be selected.

T-NSGA-III is based on NSGA-III [29], the revisions for preference integration are underlined in Algorithm 2.

The normalization of NSGA-III aims at uniformly distributed solutions when the objective values of the PF are differently scaled. In T-NSGA-III, since the goal is to achieve solutions within the target region, lower bound of the target region is used as the ideal point, the target ranges of each objective are employed as the intercepts for the new normalization process (line 17). This step has the same effect as coordinate transformation.

In the new niche-based selection (line 20), solutions in the last front F_l are separated into two parts: within the target

Algorithm 2 T-NSGA-III

Input: Target region T, maximal number of generations $t_{\max}$, population size N
Output: Final Population $P_{t_{\max}}$

$P_0 =$ Initialize( )
$Z^r =$ GenerateReferencePoints( )
$t = 0, i = 1$
while $t < t_{\max}$ **do**
  $Q_t =$ OffspringGeneration(P_t) /* generate N offspring solutions by variation */
  OffspringEvaluation(Q_t)
  $P_t = P_t \cup Q_t$
  $\{F_1, F_2, ..., F_v\} =$ Non-dominated-sort(P_t)
  repeat
    $S_t = S_t \cup F_i$ and $i = i + 1$
  until $|S_t| \geq N$
  Last front to be included: $F_l = F_i$
  if $|S_t| = N$ **then**
    $P_{t+1} = S_t$
  else $P_{t+1} = \cup_{j=1}^{l-1} F_j$
    Points to be chosen from F_l: $K = N - |P_{t+1}|$
    Normalize objective using target region T: $NormalizeT(T, S_t)$
    Associate every solution in S_t with a reference point: Associate (S_t, Z^r)
    Compute niche count of every reference point in Z^r: ComputeNiche (S_t, Z^r,F_l)
    Choose K solutions from F_l considering target region T and niche count: $P_{t+1} = NichingT(T, K, F_l)$
  end if
  $t \leftarrow t + 1$
end while

region and outside the target region. Solutions within the target region are selected first, in the order of the original niche-preservation approach of NSGA-III. If more solutions are still needed, add solutions outside the target region in the order of niche-preservation approach until the last front is full.

D. CODING STRATEGY

Geng et al. proposed a hybrid coding strategy for single objective AEOS mission planning [33]. A boolean and an integer represent the observation of one target. We improve this strategy by combining the boolean and integer into only one integer. A solution x is represented by the following integer array:

$$x = \{x_1, x_2, \cdots, x_N\}$$
$$x_i = \begin{cases} 0, & \text{if task i is not selected} \\ K, & \text{if task i is observed in its } K^{\text{th}} \text{ access window} \end{cases}$$

The length of the integer array is the target number (N). Each integer corresponds to one target. If this integer is non-zero, it indicates the chosen access window. If this integer is zero, the target will not be observed by any satellite. Figure 5 gives an example of one solution. There are eight targets to be observed in total. Target 1, 2, 4, 6 will be observed at the 2nd, 1st, 1st and 3rd access window that can observe the corresponding target. Note that the value range of each integer is different, it depends on how many access windows could observe the target. Supposing that target 1 has three access windows, then the first integer can be {0, 1, 2, 3}.

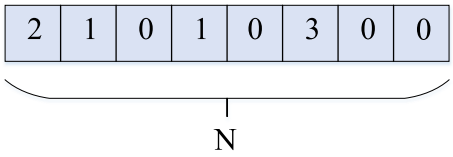

FIGURE 5. Integer coding of one solution.

This coding strategy decides which access windows to choose, but when to start the observation is determined in the decoding strategy.

E. DECODING STRATEGY

Heuristic rules are utilized for setting the start time. The main idea is to satisfy the temporality constraint and transformation constraint (refer to Section II) by sliding the observation window within the access window. The start time initializes at the middle point of each access window. The observed access window set *OAW* (according to the integer array) is sorted by chronological order. Algorithm 3 gives the detail process to set the start time. After this procedure, the observed targets and the start time for each observation are fixed, a solution plan is obtained.

F. VARIATION OPERATORS

Variation operators are required for generating offspring solutions, which include crossover operator and mutation operator.

1) CROSSOVER OPERATOR

Two crossover positions are selected randomly, and then all the integers between the two positions are swapped between two parent solutions. This operator is the so-called *two-point crossover*.

2) MUTATION OPERATOR

One integer is selected randomly with a specified probability. This chosen integer is updated with another value in its range. For example, if the range of one integer is {0, 1, 2, 3} and the current value is 2, after the mutation it becomes a value in the set {0, 1, 3} randomly.

IV. EXPERIMENTAL STUDIES

A. SCENARIO SETTINGS

The problem instances (including satellites, targets, access windows) x System Tool Kit (STK).[1] The satellite is

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

TABLE 1. Problem instance list. The target region is in the format of $\{(f_{1l}, ..., f_{5l}), (f_{1u}, ..., f_{5u})\}$, where '$f_{il}$' and '$f_{iu}$' stand for the lower bound and upper bound for objective i.

Problem instance	Target Count	AccessWindow Count	Distribution	Target Region
1	50	251	C	$\{(0.7, 0.6, 0.8, 0, 0), (1, 1, 1, 0.1, 1)\}$
2	50	266	R	$\{(0.9, 0.9, 0.9, 0, 0.6), (1, 1, 1, 0.1, 1)\}$
3	100	512	C	$\{(0.3, 0.3, 0.9, 0, 0.7), (0.6, 0.6, 1, 0.1, 1)\}$
4	100	576	R	$\{(0.9, 0.9, 0.9, 0, 0.8), (1, 1, 1, 0.1, 1)\}$
5	150	762	C	$\{(0.3, 0.2, 0.8, 0, 0.7), (0.5, 0.5, 1, 0.1, 1)\}$
6	150	820	R	$\{(0.95, 0.95, 0.9, 0, 0.8), (1, 1, 1, 0.1, 1)\}$
7	200	1009	C	$\{(0.2, 0.2, 0.9, 0, 0.6), (0.5, 0.5, 1, 0.1, 1)\}$
8	200	1128	R	$\{(0.9, 0.9, 0.9, 0, 0.7), (1, 1, 1, 0.1, 1)\}$

Algorithm 3 Start Time Setting

Input: The observed access window set *OAW*
Output: Solution plan *P*

1: **for** $\forall aw \in OAW$ **do**
2: Add *aw* to the solution plan *P*
3: **if** the transformation constraint (see II-B) is violated **then**
4: go to line 8
5: **else**
6: break;
7: **end if**
8: **if** *aw* is the first window in *P* **then**
9: go to line 15
10: **else if** *aw* is the last window in *P* **then**
11: go to line 20
12: **else**
13: go to line 25
14: **end if**
15: Move the start time *est* forwards to $st1$ when the transformation constraint is resolved.
16: **if** $st1$ is earlier than *st* **then**
17: The constraint cannot be satisfied, change *select* to **false**.
18: **end if**
19: break;
20: Move the start time *est* backwards to $st2$ when the transformation constraint is resolved.
21: **if** $st2$ is later than *et* **then**
22: The constraint cannot be satisfied, change *select* to **false**.
23: **end if**
24: break;
25: Move *est* within the range of *aw* until the transformation constraint is resolved. Both the previous and next access windows should be considered when moving *est*.
26: **if** The constraint cannot be satisfied **then**
27: change *select* to **false**
28: **end if**
29: **end for**

Chinese HJ constellation (HJ-1A and HJ-1B) and the scheduling period is 24 hours. The targets are selected from STK database. There are two kinds of distribution, which are randomly distributed all over the world and concentrated distributed inside the mainland of China. Using different numbers of the targets and varied distribution, eight problem instances are designed, as TABLE 1 shows. The target number ranges from 50 to 200, the corresponding number of access window ranges from 251 to 1128. The distribution "C" and "R" stand for concentrated distribution and random distribution respectively. The profit of each target (pr) is randomly generated in the range [1, 10] and the requested time of observation (rt) is a random value from 3 seconds to 6 seconds. A target region is set for each problem instance to stress one or several objectives. In practice, this can be done by expert knowledge or let the DM check an approximate PF and ask for the preferences. Note that all the objectives are normalized to [0,1] except for balance. All the objectives are to be maximized except for balance.

The implementation is based on MOEAFramework.[2] All the experiments are run on Microsoft Window 7 (64 bit) operational system with Intel(R) Core(TM) i5-4590 CPU and 8GB RAM.

Parameter settings are the following: The population size is 200, the maximum number of evaluation is 150000 for instance 1-6, 200000 for instance 7 and 8. The mutation probability is 0.01. In T-MOEA/D, the neighbourhood size is set as 20 and the maximum number an offspring can replace in the neighbourhood is 2. $\theta = 5$ for the PBI approach.

To examine the performance of the proposed approaches, we devise three sections of experiments. At first, the proposed algorithms are compared with non-preference-based algorithms in section IV-B. More specifically, T-MOEA/D-TCH, T-MOEA/D-PBI and T-NSGA-III are compared with MOEA/D-TCH, MOEA/D-PBI and NSGA-III, respectively. The purpose is to test the effectiveness of the preference-based algorithms. Then, in section IV-C, the proposed algorithms are compared with state-of-the-art preference-based MOEAs, i.e. T-NSGA-II [22], T-RVEA [23] and MOEA/D-c [24]. It is supposed that the preference is given before optimization, the proposed methods belong to *a-priori* approach. However, in section IV-D, we will show that the proposed algorithms can also be applied in an *interactive* way. The DM can change the preferences when checking the intermediate solutions after several iterations. A more reliable solution could be generated by interacting with the DM.

[2]http://moeaframework.org/

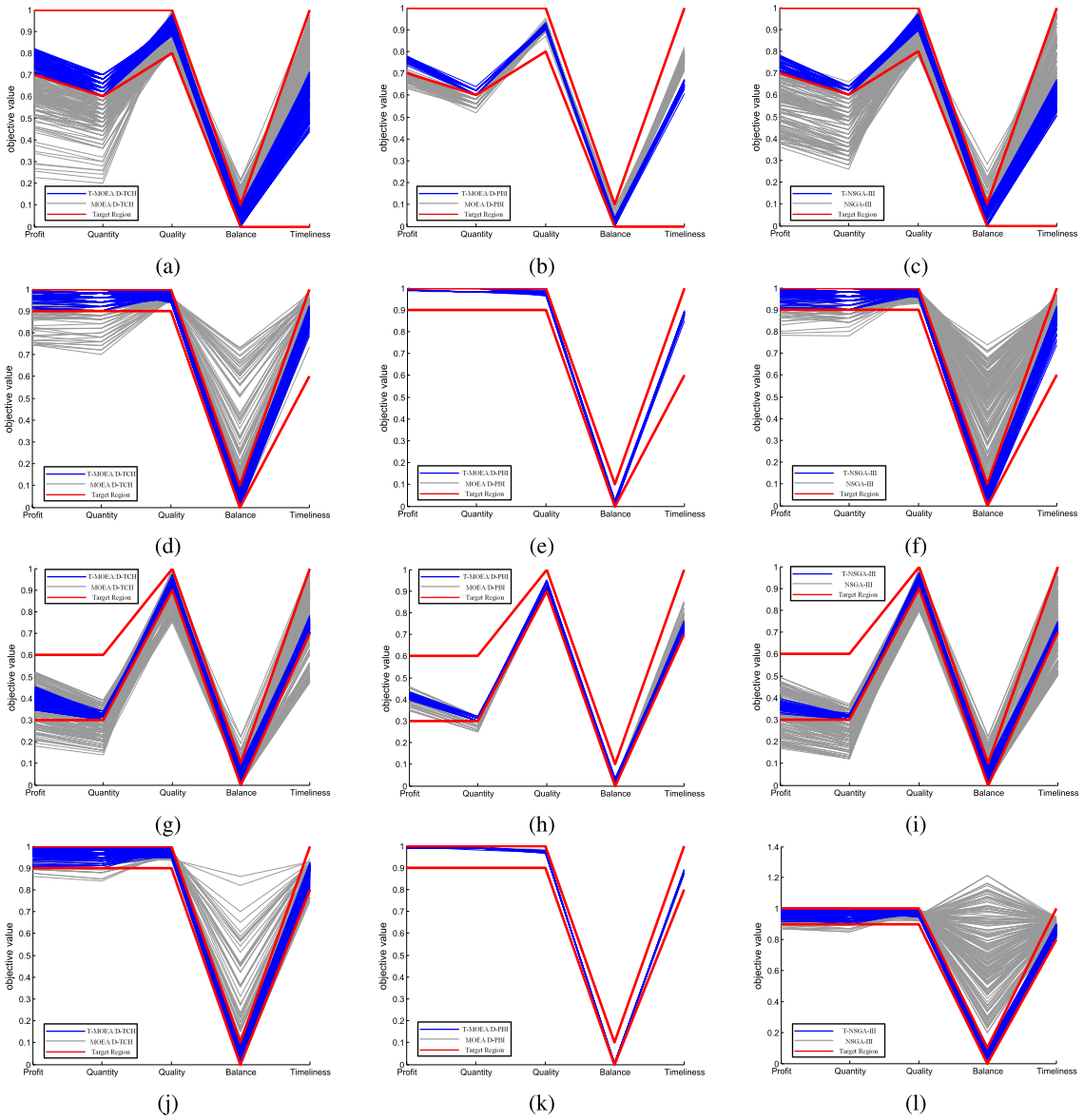

FIGURE 6. Target region-based MOEAs (blue lines) and the corresponding non-preference-based MOEAs (gray lines) on problem instance 1 (first row), instance 2 (second row), instance 3 (third row) and instance 4 (fourth row). The red lines indicate the target regiont. (a) T-MOEA/D-TCH vs MOEA/D-TCH. (b) T-MOEA/D-PBI vs MOEA/D-PBI. (c) T-NSGA-III vs NSGA-III. (d) T-MOEA/D-TCH vs MOEA/D-TCH. (e) T-MOEA/D-PBI vs MOEA/D-PBI. (f) T-NSGA-III vs NSGA-III. (g) T-MOEA/D-TCH vs MOEA/D-TCH. (h) T-MOEA/D-PBI vs MOEA/D-PBI. (i) T-NSGA-III vs NSGA-III. (j) T-MOEA/D-TCH vs MOEA/D-TCH. (k) T-MOEA/D-PBI vs MOEA/D-PBI. (l) T-NSGA-III vs NSGA-III.

B. COMPARISON WITH NON-PREFERENCE-BASED MOEAS

The proposed algorithms (T-MOEA/D-TCH, T-MOEA/D-PBI, T-NSGA-III) and the corresponding non-preference-based algorithms (MOEA/D-TCH, MOEA/D-PBI, NSGA-III) are run 20 times each. Since the purpose is to obtain a more fine-grained resolution within the target region, both convergence and diversity within the target region should be considered. Inverted Generational Distance (IGD) [34], which can measure convergence ad diversity simultaneously, is adopted as the performance metric. IGD is calculated as follows:

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

in which P is a reference set of the true PF, Q is the set of result solutions. $d(v, Q)$ is the Euclidean distance of a solution v to the closest solution in set Q. Since the true PF of this real-world application is unknown, we collect all the Pareto optimal solutions using all the algorithms from 20 runs to form a reference set. Noting that our aim is to acquire solutions in the target region, so solutions outside the

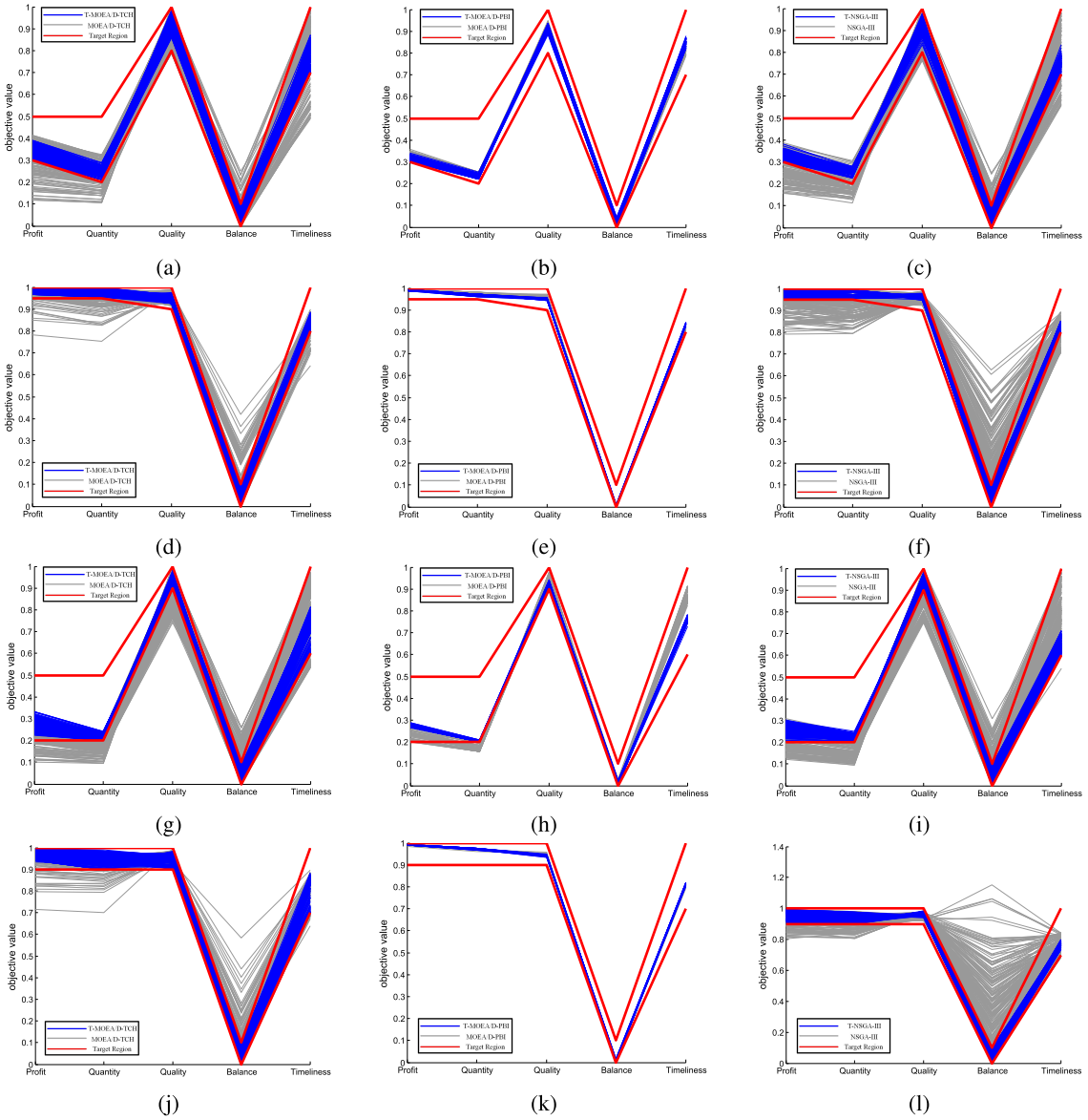

FIGURE 7. Target region-based MOEAs (blue lines) and the corresponding non-preference-based MOEAs (gray lines) on problem instance 5 (first row), instance 6 (second row), instance 7 (third row) and instance 8 (fourth row). The red lines indicate the target regiont. (a) T-MOEA/D-TCH vs MOEA/D-TCH. (b) T-MOEA/D-PBI vs MOEA/D-PBI. (b) T-NSGA-III vs NSGA-III. (c) T-MOEA/D-TCH vs MOEA/D-TCH. (d) T-MOEA/D-PBI vs MOEA/D-PBI. (e) T-NSGA-III vs NSGA-III. (f) T-MOEA/D-TCH vs MOEA/D-TCH. (g) T-MOEA/D-PBI vs MOEA/D-PBI. (h) T-NSGA-III vs NSGA-III. (i) T-MOEA/D-TCH vs MOEA/D-TCH. (j) T-MOEA/D-PBI vs MOEA/D-PBI. (k) T-NSGA-III vs NSGA-III.

target region are removed, for both the reference set P and the solution set Q.

The representative result is shown in Fig. 6 and Fig. 7, from which we can have the following observations. All the preference-based algorithms only obtain solutions within the target region. Generally speaking, T-MOEA/D-PBI has worse diversity than T-MOEA/D-TCH and T-NSGA-III. Similarly, the diversity of MOEA/D-PBI is also worse than MOEA/D-TCH and NSGA-III through visual inspection. Comparing different problem instances, we can find that the profit and quantity of concentrated distribution are smaller than that of random distribution. This is because there are more conflicts in problems of concentrated distribution, fewer proportions of targets could be observed owing to the Transformation constraint. It should be noted that profit and quantity are normalized values, which can be interpreted as a proportion, so they decrease with the increase of target number in concentrated distribution instances.

TABLE 2 presents the mean and standard deviation of IGD in 20 independent runs. Kruskal-Wall test [35] is adopted to test whether some results are from the same distribution. The best and second-best algorithms in each group are marked in dark gray and light gray background, respectively. Indifferent algorithms in Kruskal-Wall test are given the same rank. If the

TABLE 2. Mean and standard deviation of IGD within the target region in 20 independent runs. The best and second-best algorithms in each group are marked in dark gray and light gray respectively. The algorithms with '∗' are indifferent in the Kruskal-Wall test.

Group	Preference-based algorithms			Non-preference-based algorithms		
Instance	T-NSGA-III	T-MOEA/D-TCH	T-MOEA/D-PBI	NSGA-III	MOEA/D-TCH	MOEA/D-PBI
1	0.3145(0.0376)	0.1914(0.0226)	0.3470(0.0471)	0.3819(0.0562)	0.2865(0.0366)	0.5629(INF)
2	0.1515(8.66e-3)	0.1422(5.20e-3)	0.5199(0.0498)	0.3786(0.1181)	0.1960(6.29e-3)	0.6036(0.0719)
3	0.9012(INF)	0.2090(0.0489)	0.3202(0.0651)	INF(INF)	0.6326(INF)	0.3497(0.0337)
4	0.1936(0.0227)	0.14215(5.69e-3)	0.5474(0.0172)	0.5535(INF)*	0.1718(0.0128)	0.5903(0.0425)*
5	0.2666(0.0292)	0.1684(9.21e-3)	0.3227(0.0291)	0.5699(0.1043)	0.2880(0.0424)	0.3578(0.0381)
6	0.2893(INF)	0.1743(7.53e-3)	0.4201(0.0264)	0.6436(INF)	0.2944(0.0425)	0.5216(0.0519)
7	0.3262(INF)	0.1817(0.0142)	0.4998(0.0426)	0.7179(INF)	0.4163(0.1043)	INF(INF)
8	0.2712(0.0251)	0.1742(9.91e-3)	0.5241(0.0230)	0.3477(0.0565)	0.2030(0.0180)	0.5314(0.0285)

TABLE 3. Mean and standard deviation of IGD within the target region in 20 independent runs. The best, second-best and third-best algorithms are marked in dark gray, gray and light gray respectively. The algorithms with '∗' are indifferent in the Kruskal-Wall test.

Instance	T-NSGA-III	T-MOEA/D-TCH	T-MOEA/D-PBI	T-NSGA-II	T-RVEA	MOEA/D-c
1	0.3126(0.0279)	0.1866(0.0122)	0.3726(0.0339)*	0.2380(0.0199)	0.4320(0.0710)	0.3529(0.0615)*
2	0.1527(7.29e-3)	0.1468(4.91e-3)	0.5428(0.0687)*	0.1710(8.93e-3)	0.5554(0.0206)*	0.5484(0.0236)*
3	0.4712(INF)	0.2053(0.0403)	0.3328(0.0821)	0.2753(0.0519)	INF(INF)*	INF(INF)*
4	0.1962(0.0165)	0.1388(4.88e-3)	0.5516(0.0301)*	0.2240(0.0204)	0.5495(0.0351)*	0.5816(0.0504)
5	0.2723(0.0833)	0.1648(9.62e-3)	0.3399(0.0225)	0.2394(0.0192)	0.4115(0.0474)	0.5378(INF)
6	0.2779(0.0524)	0.1832(9.99e-3)	0.4324(0.0420)	0.2035(0.0147)	0.6484(INF)*	0.6756(INF)*
7	0.3098(0.0814)	0.1765(0.0113)	0.5005(0.0418)	0.2548(0.0370)	INF(INF)	0.6944(INF)
8	0.2811(0.0333)*	0.1656(6.66e-3)	0.5216(0.0218)	0.2893(0.0199)*	0.5760(0.0385)	0.5447(0.0222)

algorithm couldn't find solutions within the target region in one run, this is called a ''failed run''. If the standard deviation is infinity (INF), at least one failed run exists in the 20 runs. If the mean value is INF, more than half runs are failed run.

From TABLE 2 we can find that in the group of preference-based algorithms, T-MOEA/D-TCH has the best performance in all the 8 instances. T-NSGA-III ranks second for 7 times. T-MOEA/D-PBI is the worst algorithm except for instance 3, in which it outperforms T-NSGA-III. When it comes to the non-preference-based algorithms, MOEA/D-TCH ranks first in all the instances except for instance 3, in which MOEA/D-PBI performs best. NSGA-III and MOEA/D-PBI have second place for five and three times respectively. The poor performance of T-MOEA/D-PBI and MOEA/D-PBI is in accordance with the graphical results that they have worse diversity. Comparing the preference-based algorithm with the corresponding non-preference-based algorithm (T-NSGA-III vs NSGA-III, T-MOEA/D-PBI vs MOEA/D-PBI, T-MOEA/D-TCH vs MOEA/D-TCH), we can observe that the preference-based algorithm is always better. This proves the effectiveness of the proposed algorithms in finding Pareto optimal solutions complying with the preferences. Comparing the probability of failed run, preference-based algorithms have three infinity standard deviation. However, non-preference-based algorithms have seven infinity standard deviation as well as two infinity mean. It demonstrates that the proposed algorithms are more stable than the non-preference-based ones. It should be noted that MOEA/D-TCH is the best among the three non-preference-based algorithms. Coincidently, T-MOEA/D-TCH is the best among the three preference-based algorithms. It hints that a powerful preference-based algorithm should be based on an excellent non-preference-based optimizer.

C. COMPARISON WITH PREFERENCE-BASED MOEAS

Three state-of-the-art preference-based MOEAs are chosen to compare with the proposed approaches. The first one is T-NSGA-II [20], which shares the same coordinate transformation method with the proposed algorithms. To improve the diversity maintenance of T-NSGA-II for MaOP, the pruning process has been updated using the approach proposed by Kukkonen and Deb [36].

The second algorithm is T-RVEA [23], which is based on Reference Vector guided Evolutionary Algorithm (RVEA) [37]. Latin hypercube sampling is employed in the target region to obtain reference vectors, which will guide the search to Pareto optimal solutions preferred by the DM.

MOEA/D-c is a reference-point based algorithm that has promising results in solving MaOPs [24]. It applies an iterative weight approach to generate weight vectors for importing the preferences. The purpose is to find Pareto optimal

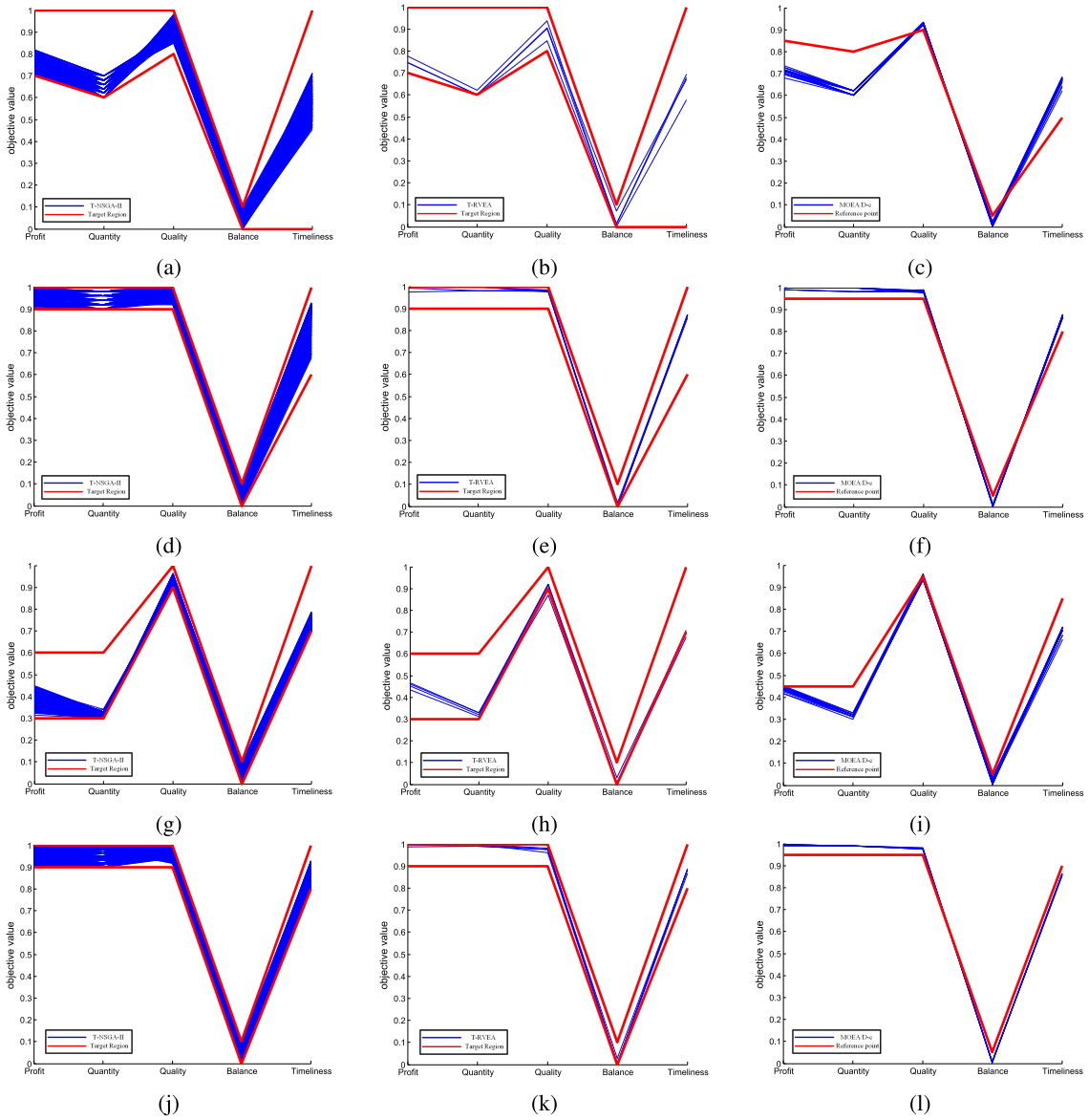

FIGURE 8. Representative results (blue lines) of T-NSGA-II, T-RVEA and MOEA/D-c on problem instance 1 (first row), instance 2 (second row), instance 3 (third row) and instance 4 (fourth row). The red lines indicate the target region (for T-NSGA-II and T-RVEA) and the reference point (for MOEA/D-c). (a) T-NSGA-II. (b) T-RVEA. (c) MOEA/D-c. (d) T-NSGA-II. (e) T-RVEA. (f) MOEA/D-c. (g) T-NSGA-II. (h) T-RVEA. (i) MOEA/D-c. (j) T-NSGA-II. (k) T-RVEA. (l) MOEA/D-c.

solutions near the provided reference point and in the vicinity of it. The extent of region of interest (ROI) is controlled by a parameter ϵ. In the experiments, we use the intermediate point of the target region as the reference point and ϵ is set as 0.5 for each objective.

The representative result of the three preference-based MOEAs is shown in Fig. 8 and Fig. 9. We can observe that the results of T-NSGA-II are similar to T-MOEA/D-TCH (refer to section IV-B). T-RVEA and MOEA/D-c have worse diversity compared to T-NSGA-II. To compare them quantitatively, the proposed algorithms and the three preference-based MOEAs are run independently for 20 times each, the mean and standard deviation of IGD are given in TABLE 3. The best, second-best and the third-best algorithm are marked in dark gray, gray and light gray background, respectively.

From the table we can find that T-MOEA/D-TCH ranks first in all the eight instances, followed by T-NSGA-II with six second places and two third places. T-NSGA-III is also competitive, which ranks second for three times and third for four times. Although T-NSGA-III outperforms T-NSGA-II in most of the 4-15 objective DTLZ benchmark problems [30], it is superior to T-NSGA-II in only 2/8 problem instances. T-NSGA-III is inferior to T-NSGA-II in 5/8 problem instances and the two algorithms are indifferent in instance 8. Comparing the remaining three algorithms, T-MOEA/D-PBI

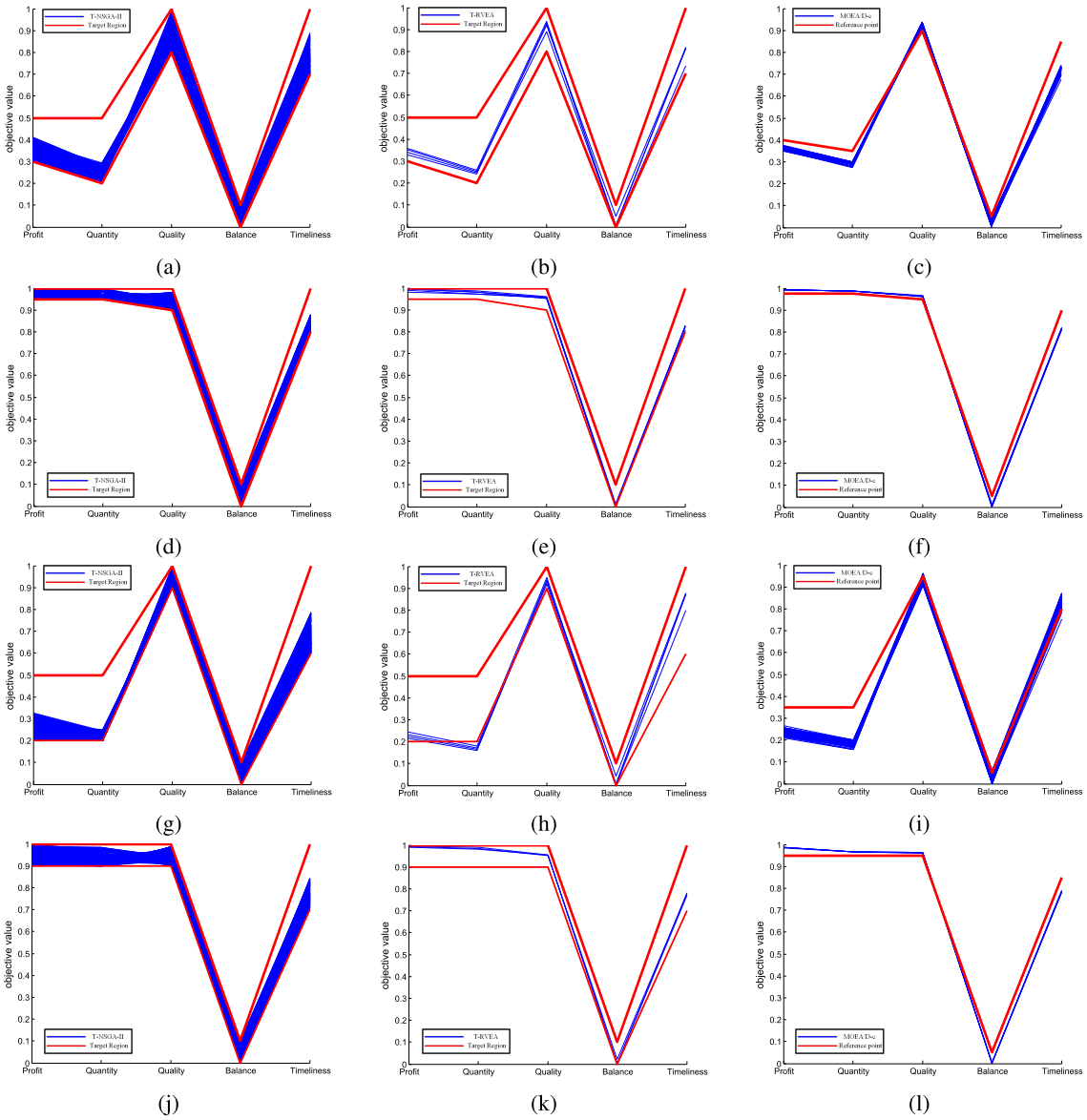

FIGURE 9. Representative results (blue lines) of T-NSGA-II, T-RVEA and MOEA/D-c on problem instance 5 (first row), instance 6 (second row), instance 7 (third row) and instance 8 (fourth row). The red lines indicate the target region (for T-NSGA-II and T-RVEA) and the reference point (for MOEA/D-c). (a) T-NSGA-II. (b) T-RVEA. (c) MOEA/D-c. (d) T-NSGA-II. (e) T-RVEA. (f) MOEA/D-c. (g) T-NSGA-II. (h) T-RVEA. (i) MOEA/D-c. (j) T-NSGA-II. (k) T-RVEA. (l) MOEA/D-c.

performs better than T-RVEA in 6/8 problem instances, better than MOEA/D-c in 6/8 problem instances. T-RVEA surpasses MOEA/D-c in 2/8 problem instances, worse than MOEA/D-c in 3/8 problem instances and indifferent to it in 3/8 problem instances. With regard to the stability, T-MOEA/D-TCH, T-MOEA/D-PBI and T-NSGA-II do not have failed runs. T-NSGA-III has only one infinity standard deviation, while T-RVEA and MOEA/D-c have several infinity standard deviations and infinity mean values.

The averaged runtime of the six algorithms is compared in TABLE 4. T-RVEA and MOEA/D-c are in general faster than the other algorithms, but their result solutions are relatively poor according to TABLE 3. T-MOEA/D-TCH runs faster than T-NSGA-III and T-NSGA-II in all the problem instances, expect for instance 8. Taking the result performance into consideration, we can conclude that T-MOEA/D-TCH is the most efficient algorithm among the six algorithms. T-NSGA-II is better than T-NSGA-III with regard to the IGD values, but it takes T-NSGA-II almost twice time to have a single run, comparing with T-NSGA-III. The reason lies in the improved pruning approach of T-NSGA-II, it enhances the diversity and the final performance, but this procedure is time-consuming.

D. INTERACTIVE APPROACH

In practice, the DM may have no precise preferences without knowing the trade-off of different objectives. Sometimes, he/she wants to change the preferences during the

TABLE 4. Averaged CPU runtime (seconds) for a single run. The fastest, second-fastest and third-fastest algorithms are marked in dark gray, gray and light gray respectively.

Instance	T-NSGA-III	T-MOEA/D-TCH	T-MOEA/D-PBI	T-NSGA-II	T-RVEA	MOEA/D-c
1	11.90	8.85	8.95	39.70	6.35	6.25
2	12.75	10.80	10.45	36.55	8.20	7.70
3	13.80	11.45	11.50	52.65	9.90	10.10
4	30.55	28.60	28.55	77.60	27.35	26.05
5	18.55	15.30	14.95	72.70	12.65	13.25
6	68.10	66.95	67.65	119.40	65.20	63.15
7	30.55	25.05	24.75	97.65	21.50	21.80
8	86.65	90.10	97.85	160.45	89.80	84.65

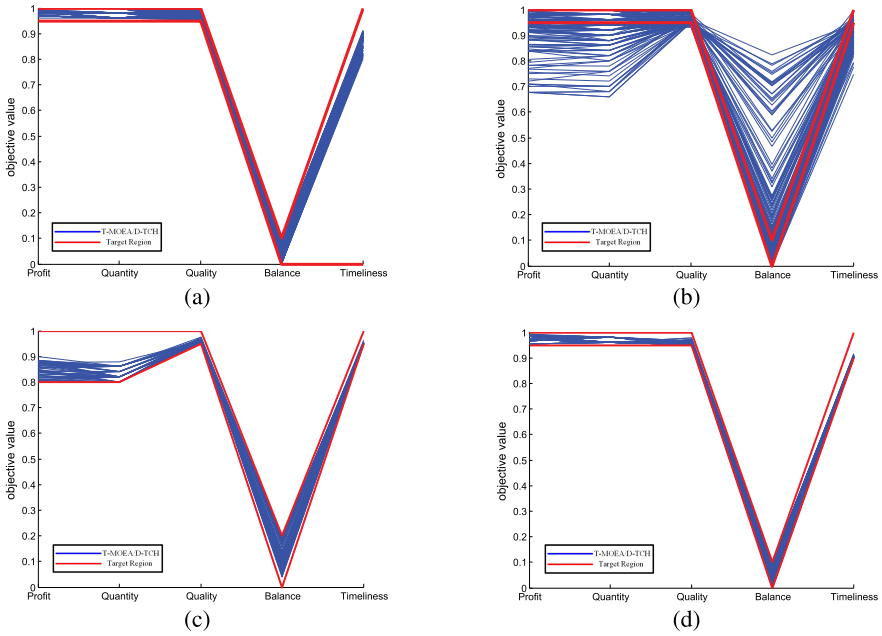

FIGURE 10. Result of T-MOEA/D-TCH (blue lines) at different interaction phases. (a) Interaction 1. (b) Interaction 2. (c) Interaction 3. (d) Interaction 4.

optimization process. Interactive approach, which makes it possible to change the preference information during the search, can help to result in a more reliable decision. A framework of the interactive version of the proposed algorithms is presented in Algorithm 4.

At the preference input step, the DM should provide upper and lower bounds of each objective as the preference information. This can be done according to the purpose of the observation (global supervision or emergency aiding) and refer to the historical plans.

We give an example of the interactive process using problem instance 2. At first, the DM requires high profit, large quantity, and high-quality solutions. He/She sets the lower bound of these three objectives as 0.95. According to the historical records, he/she chooses 0.1 as the upper bound of balance. Timeliness is not an important factor at the moment, so the preferred range of timeliness equals to the original range of $[0, 1]$. T-MOEA/D-TCH is executed for 500 generations with the chosen target region $\{(0.95, 0.95, 0.95, 0, 0), (1, 1, 1, 0.1, 1)\}$. The result is shown in Fig. 10a. After examining the solutions, the DM changes his/her mind, he/she thinks that profit, quantity, quality and balance are satisfied, but timeliness is lower than expected. He/she changes the lower bound of timeliness to be 0.95 and runs another 500 generations with the new target region $\{(0.95, 0.95, 0.95, 0, 0.95), (1, 1, 1, 0.1, 1)\}$.

From Fig. 10b we can observe that no solutions are within this region. It indicates that the specified target region is infeasible. Note that solutions with timeliness higher than 0.95 do exist, so the DM relaxes the other objectives and provides a new target region as $\{(0.8, 0.8, 0.95, 0, 0.95), (1, 1, 1, 0.2, 1)\}$. With another 500 generations in interaction 3, we get the solutions shown in Fig. 10c. At this time, the DM finds the

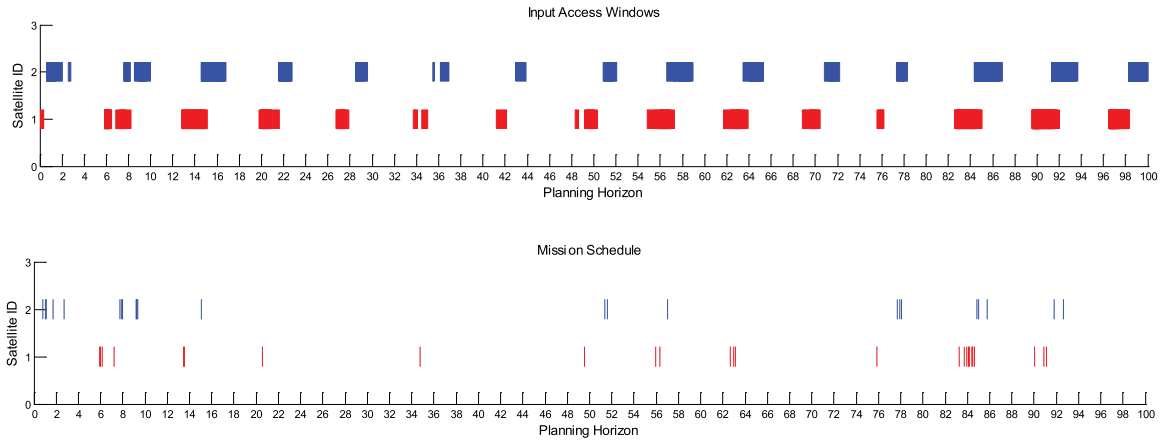

FIGURE 11. Gantt chart of the input access windows and the final solution (in decision space).

Algorithm 4 Interactive Approach

1: **Step 1: Preference input.** Ask the DM for upper and lower bounds of each objective to form the target region.
2: **Step 2: Optimization with TMOEA.** Run T-MOEA/D or T-NSGA-III with the provided target region for a predefined number of generations.
3: **Step 3: Interaction with the DM.** Show the result to the DM.
4: **if** The DM satisfies with the result **then**
5: terminate and output the result
6: **else**
7: go to **Step 1**
8: **end if**

timeliness is to his/her satisfaction, but profit and quantity should be further improved. The DM insists that profit and quantity should be larger than 0.95, timeliness could be relaxed to 0.9. T-MOEA/D-TCH is run for another 500 generations using the updated target region {(0.95, 0.95, 0.95, 0, 0.9), (1, 1, 1, 0.1, 1)}. The result of interaction 4 is given in Fig. 10d. Finally, the DM is satisfied with the solution (0.9618, 0.96, 0.9644, 0.0475, 0.9023) and the interaction process terminates. Fig. 11 presents the input access windows and the final solution using Gantt chart.

V. CONCLUSIONS AND FUTURE WORKS

This paper addresses AEOS mission planning based on evolutionary many-objective optimization. Five objectives are to be optimized simultaneously to generate a plan: total profit, the quantity of the observed targets, averaged observation quality, satellite resource equilibrium and averaged timeliness of observation. A target region, defined by a specified range of each objective, is used to express the DM's preferences and guide the optimization search. Three preference-based MOEAs, i.e., T-MOEA/D-TCH, T-MOEA/D-PBI and T-NSGA-III are applied to solve the problem. Problem-specific coding and decoding approaches are proposed, numerical experiments are conducted to test the performance of the proposed algorithms. Experiments show that compared with the non-preference-based algorithms, preference-based algorithms are better at obtaining Pareto optimal solutions complying with the preferences. Compared with other preference-based MOEAs (T-NSGA-II, T-RVEA, MOEA/D-c), T-MOEA/D-TCH has the best performance with regard to IGD and elapsed runtime. An interactive framework is also proposed for the DM to adjust preferences during the optimization process. We exemplify the benefit of interacting with the DM to obtain a more satisfactory solution.

As a future work, we want to extend the algorithms for multiple target regions. Besides, as the AEOS mission planning repeats every 24 hours, there exist a large number of historical plans. A future direction is to use machine learning method to estimate the target region, thus relieving the burden of setting a target region when the algorithm is used. The target region is only one type of preference information, sometimes a reference point or a desirability function is more suitable to model the preferences. An algorithm that can deal with diverse types of preferences will also be considered in the future.

REFERENCES

[1] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerosp. Sci. Technol.*, vol. 6, no. 5, pp. 367–381, 2002.
[2] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, "Daily imaging scheduling of an Earth observation satellite," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 35, no. 2, pp. 213–223, Mar. 2005.
[3] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite," *Comput. Ind. Eng.*, vol. 58, no. 3, pp. 509–520, 2010.
[4] Z. Yuan, Y. Chen, and R. He, "Agile earth observing satellites mission planning using genetic algorithm based on high quality initial solutions," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 603–609.
[5] C. Iacopino, P. Palmer, A. Brewer, N. Policella, and A. Donati, "EO constellation MPS based on ant colony optimization algorithms," in *Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2013, pp. 159–164.
[6] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Comput. Oper. Res.*, vol. 40, no. 7, pp. 1884–1894, 2013.

[7] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite," *Comput. Optim. Appl.*, vol. 47, no. 2, pp. 307–333, 2010.

[8] J. Wang, N. Jing, J. Li, and Z. H. Chen, "A multi-objective imaging scheduling approach for earth observing satellites," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 2211–2218.

[9] J. Li, S. Zhang, X. Liu, and R. He, "Multi-objective evolutionary optimization for geostationary orbit satellite mission planning," *J. Syst. Eng. Electron.*, vol. 28, no. 5, pp. 934–945, Oct. 2017.

[10] K. Sun, J. Li, Y. Chen, and R. He, "Multi-objective mission planning problem of agile Earth observing satellites," *AIAA J.*, vol. 4, pp. 2802–2810, 2012.

[11] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite," *Eur. J. Oper. Res.*, vol. 245, no. 2, pp. 542–554, 2015.

[12] H. Chen, C. Du, J. Li, N. Jing, and L. Wang, "An approach of satellite periodic continuous observation task scheduling based on evolutionary computation," in *Proc. Genet. Evol. Comput. Conf. Companion*, 2017, pp. 15–16.

[13] Z. Shen, H. Zou, and H. Sun, "Task scheduling for imaging reconnaissance satellites using multiobjective scatter search algorithm," in *Computational Intelligence and Intelligent Systems*. Springer, 2012, pp. 240–249.

[14] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, 2006, pp. 635–642.

[15] D. Gong, F. Sun, J. Sun, and X. Sun, "Set-based many-objective optimization guided by a preferred region," *Neurocomputing*, vol. 228, pp. 241–255, Mar. 2017.

[16] D. Gong, G. Wang, X. Sun, and Y. Han, "A set-based genetic algorithm for solving the many-objective optimization problem," *Soft Comput.*, vol. 19, no. 6, pp. 1477–1495, 2015.

[17] T. Wagner and H. Trautmann, "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 688–701, Oct. 2010.

[18] D. Gong, X. Ji, J. Sun, and X. Sun, "Interactive evolutionary algorithms with decision-maker's preferences for solving interval multi-objective optimization problems," *Neurocomputing*, vol. 137, pp. 241–251, Aug. 2014.

[19] D. Gong, J. Sun, and X. Ji, "Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems," *Inf. Sci.*, vol. 233, pp. 141–161, Jun. 2013.

[20] L. Li, Y. Wang, H. Trautmann, N. Jing, and M. Emmerich, "Multi-objective evolutionary algorithms based on target region preferences," *Swarm Evol. Comput.*, vol. 40, pp. 196–215, Jun. 2018, doi: 10.1016/j.swevo.2018.02.006.

[21] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, 2015, Art. no. 13.

[22] Y. Wang, L. Li, K. Yang, and M. T. M. Emmerich, "A new approach to target region based multiobjective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2017, pp. 1757–1764.

[23] J. Hakanen, T. Chugh, K. Sindhya, Y. Jin, and K. Miettinen, "Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2017, pp. 1–8.

[24] J. Zheng, G. Yu, Q. Zhu, X. Li, and J. Zou, "On decomposition methods in interactive user-preference based optimization," *Appl. Soft Comput.*, vol. 52, pp. 952–973, Mar. 2016.

[25] L. Li, F. Yao, N. Jing, and M. Emmerich, "Preference incorporation to solve multi-objective mission planning of agile earth observation satellites," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1366–1373.

[26] 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, 2011.

[27] L. Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, and M. Emmerich, "Building and using an ontology of preference-based multiobjective evolutionary algorithms," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* New York, NY, USA: Springer, 2017, pp. 406–421.

[28] 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.

[29] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Apr. 2013.

[30] L. Li, H. Chen, J. Li, N. Jing, and M. Emmerich, "Integrating region preferences in multiobjective evolutionary algorithms based on decomposition," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Mar. 2018, pp. 379–384.

[31] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[33] X. Geng, J. Li, W. Yang, and H. Gong, "Agile satellite scheduling based on hybrid coding genetic algorithm," in *Proc. 12th World Congr. Intell. Control Automat.*, Jun. 2016, pp. 2727–2731.

[34] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[35] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952.

[36] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Parallel Problem Solving from Nature—PPSN IX*. New York, NY, USA: Springer, 2006, pp. 553–562.

[37] 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.

LONGMEI LI was born in Xiaogan, Hubei, China, in 1989. She received the B.S. and M.S. degrees in information engineering from the National University of Defense Technology, China, in 2011 and 2013, respectively, where she is currently pursuing the Ph.D. degree. Her fields of interest include preference modeling, evolutionary multi-objective optimization, and satellite mission planning.

HAO CHEN was born in 1982. He received the Ph.D. degree from the National University of Defense Technology in 2010. He is currently with the National University of Defense Technology as an Associate Professor. His research interests include data mining and computational intelligence.

JUN LI was born in 1973. He received the Ph.D. degree from the National University of Defense Technology in 2000. He is currently a Professor with the National University of Defense Technology. His research interests include the management and analysis of big data, satellite intelligent scheduling, and controlling.

NING JING received the B.Sc. degree in communication and information systems, the M.Sc. degree in signal and information processing, and the Ph.D. degree in computer science from the National University of Defense Technology in 1983, 1986, and 1990, respectively. He has served as an Expert in earth observation and navigation area of the National High-Tech R&D Program of China. He is currently a Professor and the Director of the Department of Information Engineering, National University of Defense Technology. His research interests include geographical information systems, database systems, planning and decision support in spatial resources, spatial data analysis, and visualization. He is a Senior Fellow of the China Computer Federation (CCF), a fellow of the Technical Committee of Database System of CCF, a Vice Director of the Technical Committee of Public Security, and a fellow of the Technical Committee of Principles and Methods of the China GIS Association.

MICHAEL EMMERICH was born on 1973, Coesfeld, Germany. He received the Ph.D. degree from Dortmund University (H.-P. Schwefel Promoter) in 2005. He is currently an Associate Professor with LIACS, Leiden University, and a Leader of the Multicriteria Optimization and Decision Analysis Research Group. He carried out projects as a Researcher at ICD e.V., Germany, IST Lisbon, the University of the Algarve, Portugal, ACCESS Material Science e.V., Germany, and the FOM/AMOLF Institute on Fundamental Science of Matter, The Netherlands. He is known for pioneering work on model-assisted and indicator-based multiobjective optimization, and has edited four books and co-authored over120 papers in multicriteria optimization algorithms and their application in computational chemistry and engineering.