Swarm and Evolutionary Computation 85 (2024) 101460

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Knowledge-transfer based genetic programming algorithm for multi-objective dynamic agile earth observation satellite scheduling problem

Luona Wei [a,1], Ming Chen [b,1], Lining Xing [c], Qian Wan [d], Yanjie Song [e], Yuning Chen [b,*], Yingwu Chen [b]

[a] College of Electronics and Information Engineering, South-Central Minzu University, Wuhan, 430074, China
[b] College of Systems Engineering, National University of Defense Technology, Changsha, 410073, China
[c] College of Electronic Engineering, Xidian University, Xi'an, 710126, China
[d] National Engineering Research Center of Educational Big Data, Central China Normal University, Wuhan, 430079, China
[e] National Defense University, Haidian District, Beijing, 100091, China

ARTICLE INFO

Keywords:
Agile satellite scheduling
Multi-objective
Genetic programming
Knowledge transfer

ABSTRACT

The multi-objective dynamic agile earth observation satellite scheduling problem (MO-DAEOSSP) aims to schedule a set of real-time arrival requests and form a reasonable observation plan to satisfy various criteria. According to the requirements in practical applications, the total profit and the average image quality of scheduled requests are taken as optimization goals in this study. Compared to manually designed heuristics and iterative-based methods used in previous research, genetic programming based hyper heuristics (GPHH) can automatically evolve high-quality heuristic rules (HRs) for real-time scheduling without being highly dependent on expert knowledge. In this paper, a knowledge-transfer based multi-objective GPHH framework (KT-MOGP) is proposed, equipped with a heuristic-based simulation considering the idle monitoring, to evolve non-dominated HRs for solving MO-DAEOSSP. The heuristic-based simulation generates feasible schedules and returns fitness values for given HRs, which are the individuals evolved by KT-MOGP. KT-MOGP applies a knowledge transfer mechanism to accelerate convergence. Once a source problem is trained, its non-dominated solutions are extracted and their feature importance is transferred to guide the initialization of another target problem, by which the knowledge generated during the training process can be fully utilized. Experimental results on three sets of instances show that KT-MOGP outperforms the existing GPHH-based method and that the evolved HRs are competitive compared to several classical constructive heuristics and multi-objective evolutionary algorithms. The results also show the effectiveness of the proposed knowledge transfer-based initialization. To the best of our knowledge, this study is the first attempt to consider both multi-objective scenarios and real-time arrival requests.

1. Introduction

With the rapid development of aerospace and informatization, earth observation satellites (EOSs) have become increasingly significant in various fields such as disaster rescue, weather forecasting, environmental monitoring and city planning [1]. The earth observation satellite scheduling problem aims to process a number of requests (which are proposed by different users) through onboard optical instruments while satisfying the operational constraints [2]. For traditional EOS, the scheduling request is mainly to select the observation window for imaging it, which is the visible time window (VTW) of the request since the EOS can only observe a request when it flies directly above the target, where the orbits can be forecasted in advance through trajectory prediction. Trajectory prediction is an essential technique in earth science and space geodesy, and is also a fundamental preliminary for AEOS scheduling. Through trajectory prediction, ephemerides containing attributes of every orbit can be created. Thus, the satellite information of every moment can be obtained in advance. Once trajectory prediction is completed, the orbits for a given satellite are considered deterministic. Recent decades have witnessed the wide application of a new generation of EOS, the agile earth observation satellite (AEOS). As an EOS of superior capability, an AEOS equipped with flexibility in three axes (roll, pitch and yaw) can observe a target within a longer

* Corresponding author.
E-mail addresses: wlnelysion@163.com (L. Wei), cmself@163.com (M. Chen), xinglining@gmail.com (L. Xing), wanq8228@163.com (Q. Wan), songyj_2017@163.com (Y. Song), cynnudt@hotmail.com (Y. Chen), ywchen@nudt.edu.cn (Y. Chen).
[1] Luona Wei and Ming Chen contributed equally to this article.

https://doi.org/10.1016/j.swevo.2023.101460
Received 17 June 2022; Received in revised form 13 November 2023; Accepted 19 December 2023
Available online 9 January 2024
2210-6502/© 2024 Elsevier B.V. All rights reserved.

VTW, providing the request with more opportunity to be executed. The maneuvering improvement gives the AEOS a stronger ability to resolve conflicts between requests, for that overlapped observation requests that cannot be imaged by the traditional EOS can now be scheduled by the AEOS through maneuvering satellite attitude angles on multiple axes. In addition, the AEOS can perform more difficult requirements such as stereoscopic imaging and time-series observation owing to its advanced ability.

Although the satellite capability of the AEOS has been greatly enhanced compared to EOS, the scheduling issue of the AEOS (denoted as AEOSSP) is rather difficult to solve. Unlike the EOS scheduling problem, which is basically a selection problem, addressing an AEOSSP requires simultaneously selecting the requests to be scheduled and determining the start time of observation for each request. As discussed before, the AEOS can observe a request within a longer VTW since it has three degrees of freedom. That is, the search space of the AEOSSP is much larger and the choice of the observation window for each request is theoretically infinite. Apart from the difficulty of determining the observation window, agility also introduces new constraints and problem characteristics to the AEOSSP. Due to the time-dependent feature of the AEOSSP, even a slight change to the observation window of selected requests may cause drastic bidirectional propagated changes toward the transition time between consecutive requests. In addition, image quality can be affected by different azimuths and observing start times. A request can be observed with the highest image quality at its nadir point, the point at which the pitch angle is equal to zero. With all the above analyses, it is mandatory to investigate the AEOSSP, which has been proven to be an NP-hard problem.

During the past decades, the AEOSSP has been extensively studied, with most of the research focused on static scheduling taking the total observation profit as the optimization goal. Discretization models such as the mix integer linear programming model, graph model and constrained optimization model are established for small-scale AEOSSPs, and the exact method and several commercial solvers are adopted. Some of the early research viewed the AEOSSP as a variant of other classical combinatorial optimization problems, including the vehicle routing problem (VRP) [3,4], traveling salesman problem (TSP) and parallel machine scheduling problem [5]. Metaheuristics and problem-targeted heuristics are applied to this problem including tabu search [4,6], local search [7], iterated local search [8,9], adaptive large neighborhood search algorithm [10,11], genetic algorithms [12–14], ant colony optimization (ACO) [15], and priority-based heuristics [16,17] also have shown remarkable performance. For more details, refer to the introduction in [1]. All of the above studies treat the AEOSSP as a single-objective problem, neglecting other optimization goals in practical applications. There are some researchers working on the AEOSSP considering multiple criteria. Tangpattanakul et al. [18] are the first to establish a multi-objective AEOSSP (MO-AEOSSP) model, considering the fairness between users and the total profit simultaneously. From then on, there appear some related studies on the MO-AEOSSP considering different problem characteristics, including the preference of the decision-maker [19,20], the data transmission process [21] and the load-balance degree [22]. In summary, most of the previous studies made efforts to maximize the total profit of the scheduled requests in the static AEOSSP. There are also a few studies that consider other objectives in real-world applications, but most of their studies are still based on certain scenarios where the scheduling environment does not change.

Despite the numerous studies focus on conventional AEOSSP under a deterministic environment, the scheduling of the AEOS is always a process full of uncertainties, such as new request arrivals, breakdown of optical sensors and cloud obscuration [23]. He et al. [15] studied the scheduling problem of the AEOS considering dynamic cloud coverage and designed a hierarchical scheduling mechanism. A specific ACO approach was then proposed and showed outstanding performance in various dynamic scenarios. Cui and Zhang [24] addressed the scheduling problem of EOS requiring emergency response and designed a mission priority-based scheduling method to address the dynamic arrival of emergency requests. Povéda et al. [25] discussed the uncertain factors brought by weather and future incoming requests in satellite scheduling. To optimize request priorities, the potential of evolutionary algorithms based on local search and Population-Based Incremental Learning (PBIL) is investigated in their work. Yang et al. [26] considered the real-time response of emergent scheduling requests and proposed a dynamic-distributed organizational structure, then proposed a improved ICNP and BM mechanism to coordinate satellites. Li et al. [27] adopted an reactive method for the scheduling of spacecraft observations under uncertainties and designed a improved adaptive hybrid ACO algorithm. Kramer and Smith [28] studied the problem of dynamic satellite rescheduling and proposed an iterative repair strategy-based dynamic method equipped with several heuristic rules. Wang et al. [29] considered a multi-objective dynamic real-time scheduling scenario of the conventional EOS, taking maximizing the scheduling revenue and minimizing the perturbation while scheduling as optimization targets. A novel dynamic method DMTRH was presented equipped with some merging policy and task retracting operation, with which part of the conflicts between tasks can be resolved and the satellite resource utilization can be better enhanced. Most of the previous studies decomposed dynamic AEOSSPs into multi-stage static AEOSSPs with manually designed heuristics used to insert newly arrival observation requirements. However, manually designed heuristics may have some inherent weaknesses, for they highly rely on the prior knowledge of professionals [30]. Although there are some iterative-based schemes which can improve solution quality, their time cost is often unacceptable for dynamic environments.

According to the above review, despite various studies on the AEOSSP, so far no one has ever considered multiple criteria and the uncertainty of the AEOSSP at the same time. In this study, we focus on a specific MO-AEOSSP considering dynamic arrival observation requests (MO-DAEOSSP), which need not only to optimize two objectives simultaneously (the total profit and the average image quality of scheduled requests) but also to handle the random arrival of observation requests in real-time. To our knowledge, this study is a new attempt that fits practical applications.

Obviously, dynamic arrival observation requests introduce more difficult and uncertain characteristics. Most of the methods applied to static AEOSSP are no longer suitable for this study since requests cannot be known in advance before they arrive. During the decision process, the agility, the time-dependent feature and the uncertainty of scenario need to be considered synthetically, which further complicates the problem. Regarding the optimization criteria, the total profit and the average image quality are conflicting objectives. The former has been studied sufficiently by various single-objective and multi-objective AEOSSP research, while the latter defines the quality of the scheduled imaging tasks observed by the AEOS. For the sake of obtaining a higher total profit, it is necessary to accomplish the observation requests as much as possible within the capabilities of the AEOS. But the increase in the number of scheduled requests may lead to a decrease in the average image quality of the whole schedule, since the severe conflict between requests makes many requests unable to be observed at the moment when they can obtain the best image quality. Previous studies [9–11] involving analysis of image quality show that in most cases, a request can gain its picture with the best quality at the nadir point. That is, the quality of a given request is determined by the pitch, roll, and yaw when it is observed. Based on the aforementioned analysis, the image quality can be transformed into a time-dependent objective.

Fig. 1 is presented to better illustrate the relationship between the objectives. In the figure, r_i represents observation request and Tr_{ij} is the transition time from r_i to r_j. If all the requests are observed in their nadir point, only 4 requests can be observed as the top part of Fig. 1 shows. More requests can be scheduled as the bottom part shows. There is enough transition time (Tr_{15}, Tr_{52}) and observation window for r_5

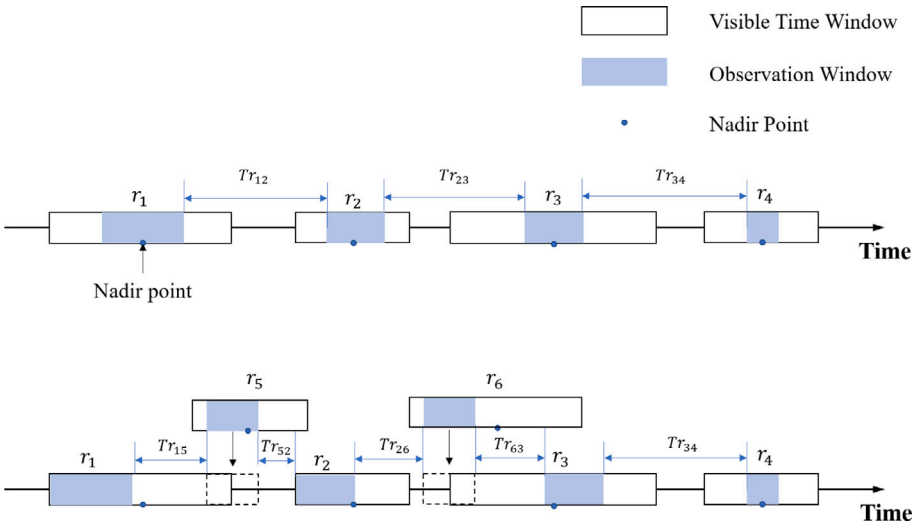


Fig. 1. A simple example of the conflicting objectives. To maximize the average image quality of solution, requests are observed at their nadir point and only four requests can be executed. On the contrary, if the scheduling goal is to maximize the number of scheduled requests, six requests can be observed, but their image quality is inferior accordingly.

to be inserted if the observations of r_1 and r_2 are rearranged. Similarly, r_6 can be scheduled by rearranging the observations of other requests. In this way, more requests can be scheduled, but not all of them can be observed with the best image quality.

To overcome the above issue, formulating the problem as a timeline-based scheduling problem and applying heuristic rules (HRs) to make real-time decisions is a promising alternative. Manually designing HRs requires extensive prior expertise, and for multi-objective problems, it becomes more complicated and time-consuming. Genetic programming (GP) seems to be an effective hyper-heuristic approach since it automatically generates superior HRs for scheduling. In fact, recent years have witnessed the application of GP-based hyper-heuristics in combinatorial optimization problems [31]. The hyper-heuristic aims to produce approximate optimal heuristics for addressing problems efficiently, and GP has been widely applied to evolve dispatching rules (DRs) for various scheduling problems in manufacturing and producing including machine scheduling [32–34], job shop scheduling [31,35] and flexible job shop scheduling [36–38]. Especially for the dynamic flexible job shop scheduling (DFJSS) [36,37,39–41], the routing and sequencing rules evolved by Genetic Programming Hyper-Heuristic (GPHH) have achieved better performance than the manually designed DRs, for both single-objective and multi-objective DFJSS. In the field of satellite scheduling, Zhang et al. [42] attempted to apply the GP to a static AEOSSP, evolving constructive heuristics by a GP-based evolutionary approach (GPEA). The obtained heuristic algorithms exhibit fine performance and can even outperform published sophisticated meta-heuristic algorithms.

Inspired by the successful application of the multi-objective hyper-heuristic, we propose a knowledge-transfer based multi-objective GP algorithm (KT-MOGP) to solve the MO-DAEOSSP. We first design a heuristic-based simulation, which constructs discrete event simulations and generates feasible schedules for fitness evaluation of HRs. Then a Pareto-based multi-objective GPHH is developed to evolve the HRs. During the training of the KT-MOGP, knowledge-transfer methods including elite tree transfer and feature importance transfer are adopted to enhance the efficiency. The initial problem that has the smallest number of candidate requests is taken as the initial source problem. The non-dominated solutions of the trained source problem are extracted and transferred to guide the initialization of the next to be trained problem together with their feature importance. After that, the newly trained problem is used as the source of the next problem to-be-trained for feature extraction and knowledge transfer, and so on. To increase the diversity of knowledge, a duplicate removal mechanism is proposed.

For the heuristic-based simulation, two statuses for a working AEOS are defined, the idle monitoring status and the request execution status. During the idle monitoring, a candidate request to be scheduled next is first selected among the arrived requests by an HR and the start time for the request is determined. After a request is selected, it needs to wait until the system time reaches the determined start time. During the waiting period, once a new request arrives, the previous steps should be repeated to reselect the request to process next and update the start time. That is, the to-be-scheduled request may be replaced by new requests. The AEOS switches to the request execution status when the time reaches the start time of the selected request, updates the scheduling plan and the current system state, and then turns to the idle monitoring status again.

More specifically, the main contributions of this work are as follows:

- The proposal of a specific MO-AEOSSP considering dynamic arrival observation requests, which is the first attempt at this problem under a new practical scenario.
- The mathematical model and two practical optimization objectives are described. Based on them, the problem is formulated as a timeline-based scheduling process.
- A GPHH-based multi-objective algorithm (KT-MOGP) is proposed, equipped with a two-status heuristic-based simulation and a knowledge transfer mechanism.
- The KT-MOGP and its evolved HRs are evaluated on various instances and show remarkable performance regarding scenarios of both multi-objective and single-objective.

The rest of this article is organized as follows. The MO-DAEOSSP is given and the mathematical model is presented in Section 2. Section 3 describes the proposed KT-MOGP and the specific heuristic-based simulation. Computational results and analysis are provided in Section 4. In Section 5, the conclusion and future work are drawn.

2. Multi-objective dynamic agile earth observation satellites scheduling problem (MO-DAEOSSP)

2.1. Problem description and assumptions

The AEOSSP is an oversubscribed planning and scheduling problem, aiming to generate feasible and reasonable working plans for an on-orbit satellite. In this work, a dynamic version of AEOSSP is considered, which needs to decide the observation targets to be scheduled and determine the start time for each target while the observation requests are randomly arriving. During the optimization process, two objective functions are taken into account to guarantee the total profit and the average image quality of the obtained working plan.

Some reasonable simplifications and assumptions are proposed as below:

- The data download process and charging process are neglected in this study.
- All the requests considered in this paper are targets which can be observed in one pass. Polygons and large area requirements are not discussed, nor are task merging and decomposition.
- The satellite can only image one target at a time and once the observation starts it cannot be preempted by any other event.
- The dynamic event studied in this paper is the dynamic arrival of observation requests, thus other events such as the breakdown of satellites or the cloud obscuration are not discussed.

2.2. Notations and variables

The observation requests which arrive in real time are taken as the input, denoted by $R = \{r_1, \dots, r_N\}$. Note that the requests are unknown before arrived. In other words, information including both the arrival time and the attributes are not known until requests arrive. Thus, the dynamic request in this paper cannot be precalculated as many previous studies [9,10] do. For each request $r_i \in R$, several parameters are defined:

- i: the identifier for r_i
- p_i: the profit earned if r_i is observed
- a_i: the arrival time of r_i
- du_i: the duration for observing r_i. Only if r_i has been observed for du_i can it be regarded successfully executed;
- $TW_i = \{tw_{i1}, \dots, tw_{ij}, \dots, tw_{in_i}\}$: the set of visible time windows (VTW) for r_i, where tw_{ij} means the jth VTW of r_i and n_i is the number of VTWs. Note that the same request may be served from two orbits, but there is at most one VTW for the request in the same orbit.

Variables for a VTW tw_{ij} are given as below:

- ts_{ij}: the start time of tw_{ij}
- te_{ij}: the end time of tw_{ij}
- o_{ij}: the index of orbit in which tw_{ij} is located
- tbq_{ij}: the time at which the best image quality of r_i can be obtained, which is the nadir point of the tw_{ij} as discussed in [9, 11]
- twq_{ij}: the time at which the worst image quality of r_i can be obtained
- $A_{ij} = \{(\gamma_t, \pi_t, \phi_t) | ts_{ij} \le t \le te_{ij}\}$: the corresponding attitude angles (pitch, roll, yaw) at any time t in the time window tw_{ij}

The attributes of the satellite and the scheduling environment are defined:

- $[ST, ET]$: the scheduling horizon of the AEOS
- $O = \{o_1, \dots, o_m\}$: the set of orbits, which is calculated by the satellite orbit trajectory prediction
- $o_k = [ot_{ls}, ot_{le}]$: the start and end time of the lth orbit

Here we define the decision variables in this study:

- $x_{ij} = \begin{cases} 1, \textit{the request } r_i \textit{ is selected to be observed in its } j \text{ th } VTW \\ 0, \textit{otherwise} \end{cases}$
- h_i: start time for observing r_i if r_i is selected.

2.3. Objective functions

$$\max f_1 = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij} \cdot p_i \Big/ \sum_{i=1}^{N} p_i \tag{1}$$

$$\max f_2 = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij} \cdot Q(r_i) \Big/ \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij} \tag{2}$$

Eqs. (1) and (2) represent the objective functions in this study, maximizing the total profit and maximizing the average image quality of scheduled requests. $Q(r_i)$ represents the image quality of request r_i:

$$Q(r_i) = 10 - 9 \cdot \frac{|h_i - tbq_{ij}|}{|twq_{ij} - tbq_{ij}|} \tag{3}$$

The image quality $Q(r_i)$ for r_i is within the interval $[1, 10]$ and is calculated depending on hi, tbq_{ij} and twq_{ij} if r_i is executed within of tw_{ij}.

2.4. Constraints

The following constraints are given to define feasible sequence of MO-DAEOSSP:

- A request r_i is not considered in the scheduling process or the candidate request queue until it arrives (the system time reaches a_i).
- The VTWs for any request is not visible until it arrives.

$$ts_{ij} \le a_i, \forall tw_{ij} \in TW_i \tag{4}$$

- Each request can be observed at most once during the scheduling horizon.

$$\sum_{j=1}^{n_i} x_{ij} \le 1, \forall r_i \in R \tag{5}$$

- The observation of one request must be inside one of it corresponding VTWs.

$$h_i \ge ts_{ij}, \ if \sum_{j=1}^{n_i} x_{ij} = 1 \tag{6}$$

$$h_i + du_i \le te_{ij}, \ if \sum_{j=1}^{n_i} x_{ij} = 1 \tag{7}$$

- Any two requests cannot overlap with each other and there must exist enough time for the AEOS to adjust the maneuvering attitude.

$$h_i + du_i + Tr_{ik} \le h_k, \ if \sum_{j=1}^{n_i} x_{ij}, \sum_{j=1}^{n_k} x_{ik} = 1 \tag{8}$$

- The satellite attitude transition time Tr_{ik} from request r_i to request r_k can be calculated by the following piecewise linear function:

$$Tr_{ik} = \begin{cases} 11.66, & \Delta\theta \le 10 \\ 5 + \Delta\theta / v_1, & 10 \le \Delta\theta \le 30 \\ 10 + \Delta\theta / v_2, & 30 \le \Delta\theta \le 60 \\ 16 + \Delta\theta / v_3, & 60 \le \Delta\theta \le 90 \\ 22 + \Delta\theta / v_4, & \Delta\theta \ge 90 \end{cases} \tag{9}$$

where v_1, v_2, v_3, v_4 are angular transition velocities and $\Delta\theta = |\Delta\gamma| + |\Delta\pi| + |\Delta\phi|$.

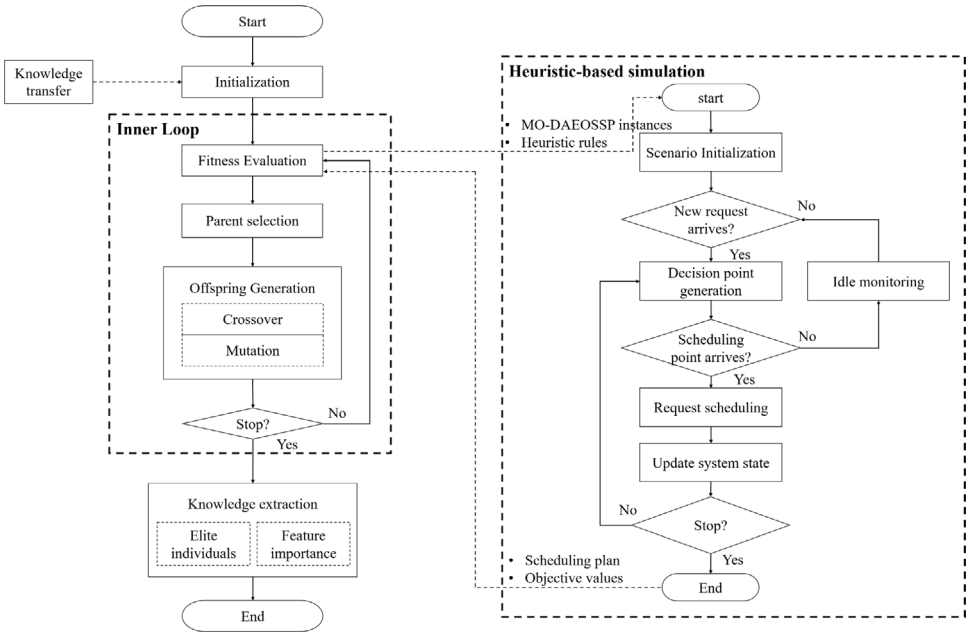


Fig. 2. The overall framework. The KT-MOGP consists of two parts, namely GP based hyper heuristic approach for evolving HRs, and fitness evaluation for each HR, which simulates the scheduling process of the dynamic AEOSSP and returns scheduling plan and objective values based on a given heuristic.

3. Knowledge transfer based multi-objective genetic programming algorithm (KT-MOGP) for MO-DAEOSSP

3.1. The overall framework of KT-MOGP

Fig. 2 shows the framework of KT-MOGP. Similar to GPHH-based methods for addressing other scheduling problems, the proposed algorithm in this study also includes a simulation stage and an evolution stage. In the simulation stage, a meta-algorithm is designed to evaluate the fitness of HRs. The right part of Fig. 2 shows the scheduling decision making process of the MO-DAEOSSP. The individuals obtained from evolution stage are first decoded into HRs. Then feasible scheduling plans and corresponding objective values are output based on the HRs and instances. The detailed scheduling decision making process is elaborated in Section 3.2.

In the evolution stage, a GPHH with knowledge transfer mechanism is adopted to evolve HRs for the MO-DAEOSSP. Since this study considers a multi-objective problem, the non-dominated sorting scheme of the NSGA-II is used in the population selection. Based on the fitness value returned from the heuristic-based simulation, the individuals are evolved through several offspring generating and reproducing operations. At the end of the evolution stage, a knowledge extraction module collects the knowledge of elite individuals and feature contributions from the output population, then applies them as transferred knowledge to guide the population generation of evolution for other scenarios. To prevent overfitting, the training instances used in the simulation stage are changed every few iterations. The utilized GPHH and knowledge transfer method is illustrated in Section 3.3.

3.2. The Heuristic-based simulation in KT-MOGP

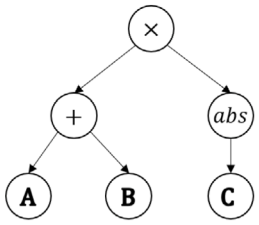
The heuristic-based simulation in KT-MOGP is a timeline-based decision process, which generates feasible schedules and returns fitness values. Since the information of a request cannot be acquired before it arrives, the decision points of the scheduling process cannot be generated in advance as Zhang et al. [42] did. Instead, the decision-making process or updating operation is triggered related to newly arrived requests.

Given a DAEOSSP instance and a heuristic rule HR, the decision process can be implemented as below. At the beginning of the scheduling horizon, there may exist several requests. The HR is first applied to calculate the priority of each request then select the request with the highest priority as the next request to be scheduled r_{next}. Based on current system state $S_t = \{t, o_t, \theta_t\}$, the start moment of r_{next} is decided as h_{next}. Then, system stays in a monitoring status before time t reaches h_{next}, that is, if a new request arrives during this period, HR will be called again to determine whether to replace the current r_{next} with the newly arrived request. Note that the feasibility of r_{next} and h_{next} has been validated during the implementation of HR. If the newly arrived request has a higher priority than r_{next}, the new request will be set to r_{next} and h_{next} should be updated according to the new r_{next}. The system switches to scheduling status when t reaches h_{next}. In this status, the scheduling plan Sch, system state S and the set of arrived candidate requests $R_{arrival}$ are updated while r_{next} and h_{next} are reset.

Since the problem in this study is a multi-orbit DAEOSSP, the scheduling horizon of the decision process is divided into several stages according to the predicted satellite orbit trajectory, and orbit switching is performed as the system time t advances. Specifically, when r_{next} cannot be selected under the current S_t and $R_{arrival}$, or t reaches the end of current orbit ot_{le} without newly arrived requests, the system time t is switched to the start time of next orbit $ot_{(l+1)s}$ with S_t and $R_{arrival}$ are then updated.

In this heuristic-based simulation, each output schedule Sch contains a series of request execution events σ, i.e. $Sch = (\sigma_1, \sigma_2 ...)$, where each $\sigma = \langle r_\sigma, h_\sigma, du_\sigma, \theta_\sigma, p_\sigma, Q_\sigma \rangle$ is characterized by its request r_σ, start time h_σ, duration du_σ, attitude angle $\theta_\sigma = (\gamma_\sigma, \delta_\sigma, \phi_\sigma)$, profit p_σ and image quality Q_σ. System state is a tuple $S_t = \langle t, o_t, \theta_t \rangle$ consist of current time t, current orbit o_t and attitude angle θ_t. The heuristic rules HRs are depicted by individuals in the evolution stage of the KT-MOGP, whose representation will be fully described later.

Algorithm 1 indicates the process for evaluating heuristic rule. For a given instance and heuristic rules, feasible scheduling solutions and corresponding objective values can be output by applying Algorithm 1. Lines 3 to 5 are the initialization of the whole scheduling. Note that orbits in this study can be regarded as independent scheduling horizons. Each orbit can be divided into sunshine interval and shade interval, where only in the sunshine can the satellite accomplish observation.



(a) Tree representation of HR

Index	Node
1	×
2	+
3	*abs*
4	**A**
5	**B**
6	**C**

(b) Nodes of HR

	Node	Floor	Parent	Left	Right
1	1	1	0	2	3
2	2	2	1	4	5
3	3	2	1	6	-1
4	4	3	2	-1	-1
5	5	3	2	-1	-1
6	6	3	3	-1	-1

(c) Matrix representation of HR

Fig. 3. An example of the heuristic rule for DAEOSSP. Based on the given index of nodes in 3(b), the matrix representation of HR is shown in 3(c). "Floor" attribute represents the level of the tree where the node is located, "parent" denotes the upper-floor node connected to the current node, "left" and "right" is the next-floor nodes connected. Specifically, when the current node has no successor nodes, "left" and "right" are set to −1.

Algorithm 1 The heuristic-based simulation of DAEOSSP

Input: A DAEOSSP instance, a heuristic rule HR
Output: A feasible schedule Sch

1: All requests are not completed, the schedule Sch is empty
2: **while** $t \le ET$ **do**
3: $l \leftarrow 1$ // Start from the first orbit
4: $t \leftarrow ot_{ls}$
5: Update the set of arrived requests $R_{arrival}$ and current system state S_t
6: $r_{next} \leftarrow \emptyset$ // The next request to be scheduled
7: **if** $r_{next} = \emptyset$ **then**
8: **while** $R_{arrival} = \emptyset$ **do**
9: Extract the next arrived request r'
10: **if** $a' \ge ot_{le}$ or $r' = \emptyset$ **then**
11: // Switch to the next orbit
12: $l \leftarrow l + 1$
13: $t \leftarrow ot_{ls}$
14: Update S_t, $R_{arrival}$
15: **else**
16: $t \leftarrow a^i$
17: $R_{arrival} \leftarrow r'$
18: Update S_t
19: **end if**
20: **end while**
21: $[r_{next}, h_{next}] \leftarrow HR(R_{arrival}, S_t, Sch)$
22: **else**
23: **while** $t < h_{next}$ **do**
24: Extract the next arrived request r^i
25: **if** $a' < h_{next}$ **then**
26: // Monitoring
27: $R_{arrival} \leftarrow R_{arrival} \cup r'$
28: $[r_{next}, h_{next}] \leftarrow HR(R_{arrival}, S_t, Sch)$
29: **else**
30: // Scheduling
31: $Sch \leftarrow Sch \cup r_{next}$
32: $t \leftarrow h_{next} + du_{next}$
33: Update S_t, $R_{arrival}$
34: $r_{next} \leftarrow \emptyset$
35: **end if**
36: **end while**
37: **end if**
38: **end while**

In the shade interval, the satellite releases the occupied resource to prepare for the next orbit. Because of the above reasons, the orbit in this study refers specifically to the sunshine interval. The satellite switches to the start time of the next orbit after the observation in the sunshine of one orbit, which is stated in Lines 10 to 14.

In Algorithm 1, $R_{arrival}$ and S_t is updated as below:

- Regarding $R_{arrival}$, first find all requests that have already arrived and are in the current orbit, then delete or clip the conflicting VTWs whose current time t exceeds its end time or conflicts with existing scheduled requests in Sch.
- Based on the last scheduled request in Sch and current system time t, the current attitude angle θ_t and orbit o_t for S_t can be determined.

3.3. The genetic programming hyper Heuristic (GPHH) in KT-MOGP

In the GPHH of the KT-MOGP, the HRs to be evolved for fitness evaluation are represented as list trees. Different from standard GPHH for single-objective, this study considers heuristic rule training in multi-objective scenarios, and the DAEOSSP is complex and difficult to solve. Therefore, to improve the training efficiency, some knowledge transfer strategies for extracting elite individuals and the feature contribution are introduced. Besides, to enhance the diversity of population so that new regions which have not been explored can be better searched, a duplicate removal mechanism is introduced by utilizing the characteristics of list tree to delete duplicate individuals.

3.3.1. Individual representation and initialization

As mentioned above, individuals in the evolving population of KT-MOGP are heuristic rules for decision-making, which are used to calculate the priority of each candidate request. These individuals are described as tree representations in this study. Fig. 3 depicts a simple individual example.

The HR in Fig. 3(a) is $(A+B)\times|C|$, three feature terminals and three operators are adopted as drawn in Fig. 3(b). The tree representation can also be transferred into a matrix formation in Fig. 3(c), the specific position of each node can be described by establishing links.

A growth method is implemented to generate the initial population of KT-MOGP without any prior transferred knowledge. With the limited maximum depth of the tree representation, each node of the tree is randomly selected among the sets of features and operators.

When there exists an output population of HR from a previous scenario, we apply the knowledge transfer mechanism to extract the elite individuals and calculate the feature contribution of each feature terminal, then set them to guide the training of HR in other scenarios. This part will be explained in detail later.

3.3.2. Genetic operations

The genetic operations of the KT-MOGP include selection, crossover, mutation and population reproduction.

- **Selection**: Before selecting individuals, the non-dominated sorting and crowding distance calculation of NSGA-II [43] is adopted to sort the given population. Then a tournament selection is used to select individuals to form the mating pool for offspring generation.

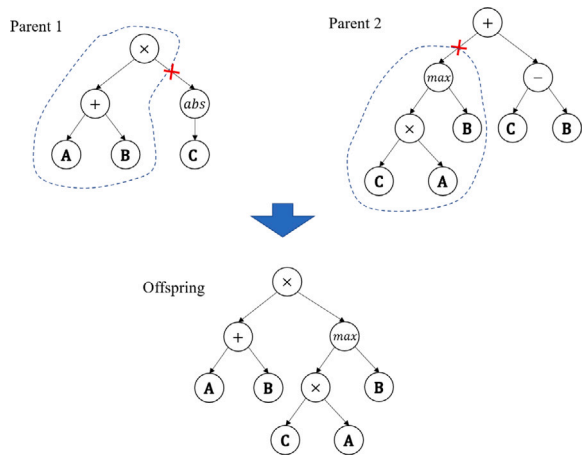


Fig. 4. An example of crossover operator in KT-MOGP.

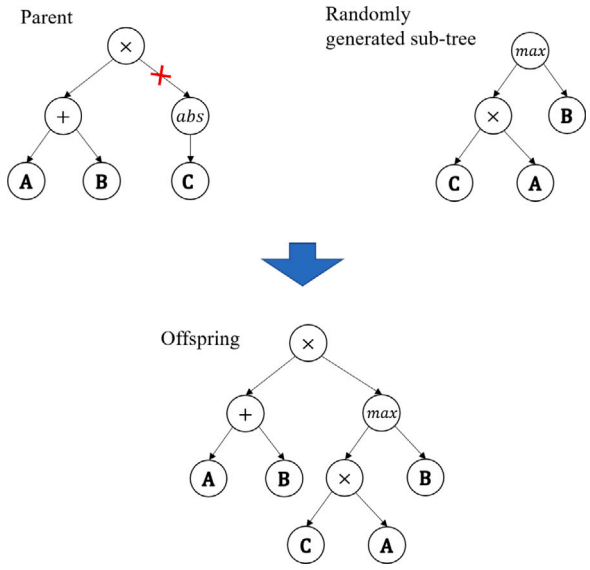


Fig. 5. An example of mutation operator in KT-MOGP.

- **Crossover**: Sub-tree crossover is used in this study. As shown in Fig. 4, two parents are selected from the mating pool, a crossover point is randomly sampled on each tree. Then the sub-tree of parent 1 containing root node and the sub-tree of parent 2 containing external points are combined to form a new individual.

- **Mutation**: The single-point mutation is used in this study. As shown in Fig. 5, a parent is selected and a mutation point is randomly sampled. Then the new individual is produced by re-combining the parent tree and another randomly generated new sub-tree at the mutation point.
- **Population reproduction**: At the end of each iteration, the parent population and new offspring individuals generated by the above genetic operation recombine to form a new population. When performing reproduction, non-dominated sorting and crowding distance are applied to select individuals.

3.4. Knowledge transfer-based initialization

As illustrated in Fig. 2, after completion of the evolution stage, we perform some knowledge extraction operations on the final output population. Specifically, the elite individuals in the final population are extracted, and the contribution of feature terminals is measured. This information will be used later in the initialization process of the KT-MOGP for a new scenario.

3.4.1. Elite-preserving operation

The effectiveness of biased subtree transfer in GPHH has been verified in [44], transferring elite individuals from a source problem can help the algorithm to converge faster on related target problems.

In KT-MOGP, since the evolution stage are conducted among multi-objective scenarios, the set of transferred individuals preserves the non-dominated solutions of the final population. Note that there may exist several duplicates in the set of non-dominated solutions. To enhance the diversity of transferred knowledge and to avoid repeatedly computing the fitness of the same individuals, a duplicate removal operation is involved as presented in Algorithm 2.

Given the final population of a completed KT-MOGP, the non-dominated individuals with unique fitness values are first preserved. Then for the remaining individuals, the phenotypic duplicates of them are deleted by the following step. Given an individual $\Omega^*[i]$, if there exists no individual which has the same fitness value as it does, $\Omega^*[i]$ is directly put into the set of preserved individuals Ω'. Otherwise, a rough method is first applied to compare the matrix representation of $\Omega^*[i]$ and $\Omega'[j]$ (the individual in Ω' with the same fitness value as $\Omega^*[i]$)). If they have the same matrix representation, discard $\Omega^*[i]$. The phenotypic distance of these two individuals should be measured as shown in line 7 if they have a different matrix representation. In this study, we adopt the method successfully utilized in [45–47]. Specifically, given a specific system state and a series of candidate requests, use different individuals to calculate their priorities and sort the requests to generate two sorting vectors. Determine whether two individuals have the same phenotype by calculating the Euclidean distance between the vectors.

By the use of matrix representation, the time cost for duplicate removal can be greatly decreased. Different from the direct application of phenotypic characterization in previous work [45,48], the MO-DAEOSSP is a problem of complicated features and constraints, which makes the phenotypic characterization more time-consuming. Instead, by comparing the matrix representations of individuals, the output heuristic rules can be roughly filtered first without the phenotypic characterization, which can greatly save the whole computation cost. Taking the output individuals of KT-MOGP in 150-request scenario as an example, it takes 83.34 s to remove the duplicates based on phenotypic characterization, while only 27.99 s are needed adopting the method in Algorithm 2, saving the computation time by 66.41%.

3.4.2. Feature contribution extraction

By fixing a feature terminal to a constant, its contribution can be figured out [49]. Thus, the feature contribution is calculated by the following steps. First adopt the elite-preserving operator to the set Ω' of individuals. Then replace the to be removed feature terminal x with a constant of one. The contribution of the specific feature can be obtained by calculating the change of fitness value. The feature contribution of x is the mean gap $\overline{G}_x$ of all the individuals:

$$\overline{G}_x = \frac{1}{|\Omega'|}\sum_{i=1}^{|\Omega'|} |F(\Omega'[i]) - F(\Omega'[i]|x=1)| \tag{10}$$

Note that only the feature terminals are extracted, for the contribution of the operators is relatively abstract and hard to extract.

3.4.3. Guided initialization strategy

Based on the above two knowledge extraction methods, we design a guided initial population generation strategy as shown in Algorithm 3.

The new population consists of three parts: the preserved elite individuals from completed source problem, individuals generated using

Algorithm 2 Duplicate removal

Input: A set of non-dominated solutions Ω, their corresponding fitness values F_Ω
Output: A set of non-dominated solutions without phenotypic duplicates Ω'

$\Omega' \leftarrow Unique(\Omega, F_\Omega)$ // Select the individuals with unique fitness values
$\Omega^* = \Omega - \Omega'$
for $i = 1 \rightarrow |\Omega^*|$ **do**
  if $F(\Omega^*[i]) = F(\Omega'[j] \in \Omega')$ **then**
    Get the matrix representation of $\Omega^*[i]$ and $\Omega'[j]$ as $M_{\Omega^*[i]}$, $M_{\Omega'[j]}$
    if $M_{\Omega^*[i]} \neq M_{\Omega'[j]}$ **then**
      $D(i,j) \leftarrow PhenotypicDistance(\Omega^*[i], \Omega'[j])$
      if $D(i,j) \neq 0$ **then**
        $\Omega' \leftarrow \Omega' \cup \Omega^*[i]$
      end if
    end if
  else
    $\Omega' \leftarrow \Omega' \cup \Omega^*[i]$
  end if
end for

the feature contribution knowledge and randomly generated individuals. Noticeably, when generating individuals using the feature contribution, a roulette wheel selection based on the feature contribution is adopted to sample the feature terminal during the growth method of individual.

Algorithm 3 Knowledge transfer-based initialization

Input: Ω', the preserved elite non-dominated HRs from completed problem; $\overline{G}$, the set of contribution for feature terminals; N, population size
Output: Pop, the knowledge transfer-based initial population

if $\Omega' < \lfloor \frac{N}{3} \rfloor$ **then**
  $Pop \leftarrow \Omega'$
else
  $Pop \leftarrow RandomSelect(\Omega', \lfloor \frac{N}{3} \rfloor)$ // Randomly select $\lfloor \frac{N}{3} \rfloor$ individuals from Ω'
end if
for $i = 1 \leftarrow \lfloor \frac{N}{3} \rfloor$ **do**
  $Pop \leftarrow Pop \cup GuidedGeneration(\overline{G})$ // Generate individuals using the feature contribution knowledge
end for
for $i = 1 \leftarrow N - |Pop|$ **do**
  $Pop \leftarrow Pop \cup RandGeneration()$ // Randomly generate remaining individuals
end for

4. Computational experiment

In this section, the experiment design including instance generator, parameter settings and performance metrics is first introduced. Then the computational results are presented and some analysis and discussions are given.

4.1. Instance generation

In the training process of KT-MOGP, six scenarios whose candidate observation requests range from 100 to 350, with an increment step of 50, are constructed. For each scenario, 50 instances are generated independently to form the training test. The average performance of 5 randomly selected instances is used for each fitness evaluation in KT-MOGP. To prevent overfitting, the instances for the simulation stage of KT-MOGP are reselected in each iteration. To validate the test performance, six test sets are generated, each of which contains 30 independent instances under the same scenario.

Table 1
Orbital parameters of the satellite: a, the length of semi-major axis; e, eccentricity; i, inclination; ω, argument of perigee; Ω, right ascension of the ascending node; and m the mean anomaly.

Satellite	α	e	i	ω	Ω	m
AS-01	7141701.7	0.000627	98.5964	95.5069	342.307	125.2658

For the generation of each instance, we adopt the strategy used by Liu et al. [11] and He et al. [10]. Each observation request is defined by its geographical position (latitude, longitude, and altitude). Given the number of total candidate requests, a uniform random distribution is utilized to spread the targets uniformly in a given region. For each request, the VTWs are generated based on the geographical position, satellite capability and ephemerides which created through trajectory prediction by the orbital parameters shown in Table 1. In this study, the AS-01 satellite is used for our experiments. The maximum maneuvering angles of the satellite in pitch, roll and yaw axes are 45°, 45°, and 90°, respectively. Two kinds of regions are considered: (1) the Chinese region (3°N to 53°N and 74°E to 133°E) and (2) worldwide region. The scheduling horizon is from 2013/4/20 00:00:00 to 2013/4/20 24:00:00. Other factors for each request are as follows: identifier, priority (in the range of [1,10]), duration (in the range of [15,30] in seconds), arrival time, all the factors above have also been stated in Section 2.2. The arrival time for each request is randomly sampled between the start time of the scheduling horizon and the start time of the last observation opportunity for this request. That is, before the arrival time, this request and all the attributes are unknown for the system.

Three test sets are adopted in this study:

- Set I: An instance set containing 30 test instances. The number of observation requests ranges from 100 to 350, with an increment step of 50. In each instance, requests are uniformly distributed worldwide using the method proposed in [11]. All requests in this tested set arrive dynamically.
- Set II: An instance set containing 30 test instances. In this set, scenarios with more conflicting requests are tested. This test set includes 30 instances with request numbers ranging from 100 to 350, where the requests of each scenario are uniformly distributed in the area of 3°N–53°N and 74°E–133°E (the China area). All requests in this test set arrive dynamically.
- Set III: All settings of Set III are the same as Set II, except that the requests in this test set are static tasks.

The dataset, which contains various-size instances for training and testing, is publicly available at https://github.com/wlnelysion.

4.2. Parameter settings

All experiments in this paper are conducted using an i7-9700K CPU, 64 GB of RAM, Windows 10 operating system, and MATLAB 2020b is used for coding. Some of the comparison methods are programmed and implemented through the Evolutionary Multi-objective Optimization Platform (PlatEMO) [50].

Table 2 shows the feature terminals used in this study. Three types of feature terminals are included: the features for request, the features for VTW and the feature for system state. Among them, SL is the remaining length of the VTW tw_{ij} at system time t, and it is normalized by dividing the length of the current orbit. WT is the gap from the start time of VTW tw_{ij} to current system time t. If t has entered the range of tw_{ij}, its WT is set to 10e−6. All feature terminals are normalized into

Table 2
The feature terminal set in this study.

Terminal	Description
P	The profit of request r_i
DU	The duration of request r_i
OTWN	The number of VTWs for r_i
OTWL	The overall length of VTWs for r_i
TIS	Current system time t
REPT	The remaining time of the current orbit o_l
APR	The average profit of the currently scheduled requests
AQU	The average image quality of the currently scheduled requests
OTWS	The start time of the VTW tw_{ij}
OTWA	The attitude transition angle when inserting request r_i at its earliest execution opportunity
CONTWN	The number of VTWs that conflict with the VTW tw_{ij}
CONTWL	The overall length of VTWs that conflict with the VTW tw_{ij}
SL	The slack degree of the VTW tw_{ij}
WT	The waiting time of the VTW tw_{ij}
BQOT	The time at which the best image quality can be obtained
WQOA	The time at which the worst image quality can be obtained

Table 3
The Parameter settings used in KT-MOGP and MO-GPEA.

Parameter	Value
Initialization	Growth
Population size	200
Number for iterations	51
Maximum depth of the initial population	5
Maximum depth during evolution	10
Crossover probability	0.85
Mutation probability	0.15
Selection	Tournament selection with size 2

the range [0,1] during the fitness evaluation procedure. The function set is set to $\{+, -, \times, \div, \max, \min, abs\}$, which is widely used in other GPHH research. Note that the division ÷ is a protected division which returns 1 if divided by 0.

Table 3 shows the parameter settings of KT-MOGP.

To evaluate both the training and test performance, experiments are designed to compare the proposed algorithm with other methods as follows:

- To verify the effectiveness of the proposed heuristic template, KT-MOGP is compared with GPEA [42], which is the only existing GPHH-based algorithm for AEOSSP. Since GPEA is a method targeting the single-objective AEOSSP, for a fair comparison, we embedded the heuristic template of GPEA into the multi-objective framework of KT-MOGP (denoted as MO-GPEA), then compared their training and test performance under the same training and test instances. Note that the setting of MO-GPEA is exactly the same as that of KT-MOGP, except the heuristic template. Table 3 shows the parameter settings of KT-MOGP and MO-GPEA. Besides, to demonstrate the superiority of KT-MOGP among static scenarios, simulations are conducted to compare KT-MOGP with MO-GPEA and several classical MOEAs on an independent test set of MO-AEOSSP.
- Likewise, to further verify the effectiveness among single-objective scenarios, we compare the proposed KT-MOGP with GPEA on the same single-objective training and testing instances. It is worth mentioning that to verify the performance of KT-MOGP on a single-objective scenario, only the selection operator in the algorithm needs to be modified. The KT-MOGP and GPEA are then compared with 5 other classical heuristics rules on an independent test set.

4.3. Experimental results

4.3.1. Multi-objective scenarios

As described in Section 4.1, for each training scenario we use 50 different instances for simulation. In the training process of KT-MOGP and MO-GPEA, we ensure the fairness of the training process by using the same random seed for the two algorithms so that they select the same training instances during fitness evaluation. The 100-request scenario is regarded as the initial source problem for the knowledge transfer mechanism. That is, the initialization of 150-request scenario uses the extracted knowledge from 100-request scenario, the 200-request scenario uses the output knowledge of 150-request scenario, and so on. The final non-dominated solutions of different training scenario are depicted in Fig. 6, where f_1 is the total profit and $y - axis$ denotes the average image quality.

It is obvious that in all scenarios, KT-MOGP outperforms MO-GPEA according to the obtained final Pareto fronts (PFs). From Fig. 6, we can see that with the increase in the total observation profit, the average imaging quality of scheduled requests correspondingly decreases, which is in line with our previous analysis of the two conflicting optimization objectives. It is also obvious from the figure that KT-MOGP can generally obtain solutions with higher average imaging quality in each training scenario than MO-GPEA.

To test the effect of the HR sets generated through the evolution of the proposed method, the test is first performed on Set I, which is a separate test set generated worldwide. The HR sets obtained in Fig. 6 are directly used rather than re-evolving HRs by KT-MOGP and MO-GPEA. The results are shown in Table 4, where HV, f_1 and f_2 denote the hypervolume [51], best objective value of f_1 (total obtained profit) and best objective value of f_2 (average image quality), respectively. HV is the main performance indicator used in this study for comparison and analysis of results of different methods, since it is a strictly monotonic unary indicator [52–55] which can easily compare different PFs and capture order relations between them [56]. In addition, for practical problem with complex constraints, HV is easy to apply and can reflect the dominance and distribution properties of nonlinear PFs [56,57].

From the table, one can see that HRs evolved by both KT-MOGP and MO-GPEA (denoted as KTGPHRs and GPEAHRs) can achieve good performance in instances in Set I, but obviously KTGPHRs perform better than GPEAHRs in most instances. In terms of HV, which reflects the overall performance of the algorithm, there is only one (350_1) among all 30 instances in Set I where the HV value of KTGPHR is worse than that of GPEAHR. By comparing the objective values, similar conclusions can be drawn. The f_1 obtained by KTGPHRs is inferior to GPEAHR in only one instance (250_3). Although GPEAHRs can get better f_2 in some instances, the objective values are not significantly better than those of KTGPHRss, with worse performance in f_1 and HV, indicating that the comprehensive performance of MO-GPEA is inferior.

To verify the performance of the HR sets in scenarios with more conflicting requests and to measure the generalization of the algorithm among instances generated by different distributions, a set of instances generated in China (Set II) is tested. The results are shown in Table 5. It can be seen that KTGPHRs still have a very obvious advantage over GPEAHRs in these scenarios. Only in very few instances is the f_2 obtained by GPEAHRs slightly better than that of KTGPHRs, but at the same time KTGPHRs still maintain better HV and f_1 values. Compared with Table 4, the completion rate of the proposed requests is greatly reduced, and with the increase in the number of requests, the completion rate (or total profit f_1) decreases more significantly.

The multi-problem Wilcoxon rank-sum test [58] at a 5% significance level is conducted to summarize the HV performance among the instances in Set I and Set II. According to the p-values presented in Table 6, KT-MOGP is significantly better than MO-GPEA for addressing the MO-DAEOSSP.

Although the proposed algorithm is designed for dynamic multi-objective satellite scheduling problems, we are still curious about

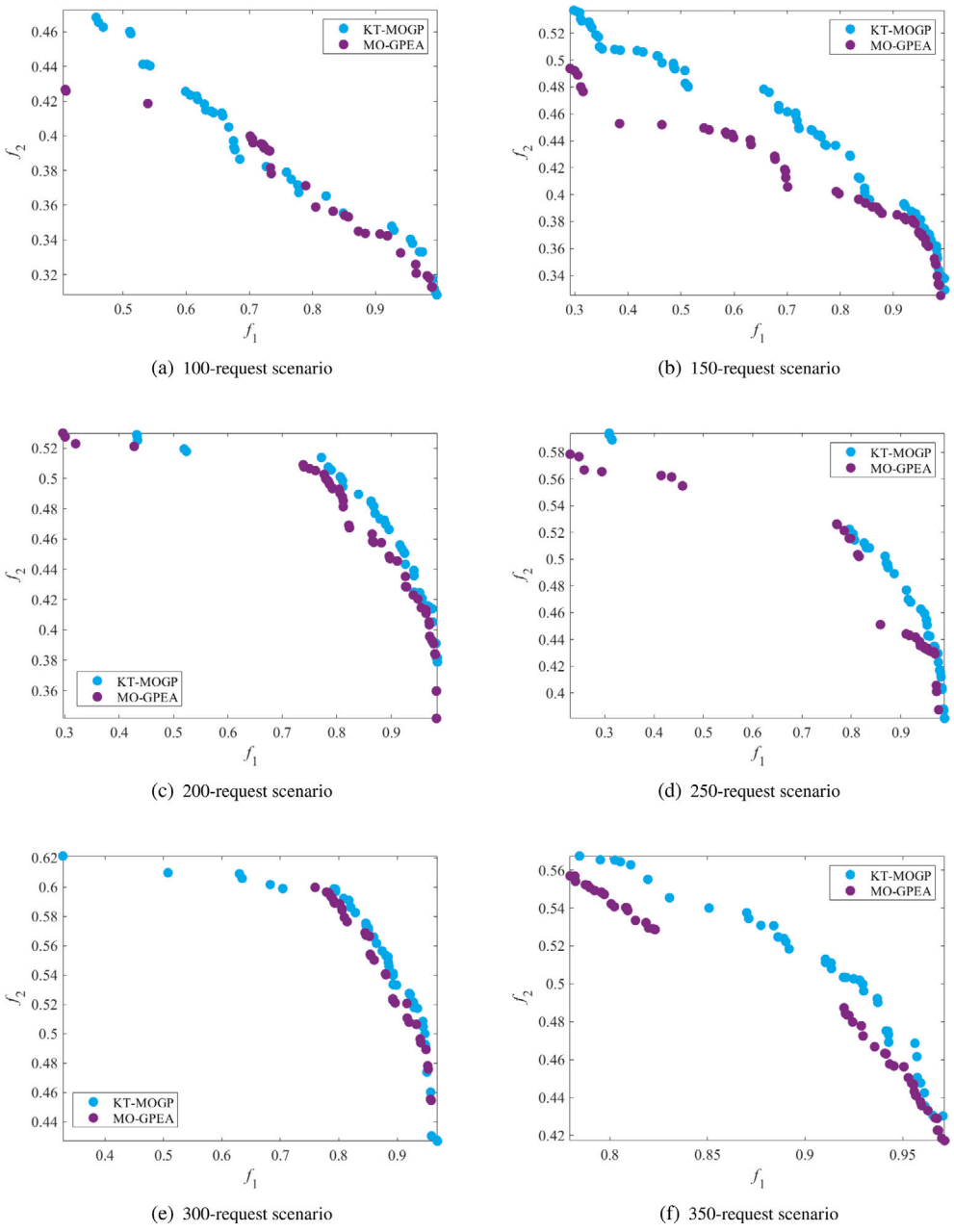


Fig. 6. Non-dominated solutions reserved in different scenarios of MO-DAEOSSP using KT-MOGP and MO-GPEA.

whether it can maintain its superiority in static scenarios. Therefore, we compared KTGPHRs and MO-GEPArules with NSGA-II [43] and MOEA/D [59] in Set III. NSGA-II and MOEA/D are classical multi-objective evolutionary algorithms (MOEAs) which have already been successfully applied to MO-AEOSSP, and the parameter settings and problem-targeted operators used in [22] are adopted. For each instance, we present the average values of 10 independent runs for NSGA-II and MOEA/D. The results are shown in Table 7.

We find some interesting phenomena through the experimental results. First, among the four participating algorithms, NSGA-II performs the best in most instances, followed by KTGPHRs and GPEAHRs, while MOEA/D performs the worst in all 30 test instances. However, as the number of requests to be selected in the scenario increases, the advantage of NSGA-II is weakened accordingly. In instances with a request size of 350 (350_1-5), we surprisingly find that the performance of NSGA-II is no longer better than that of KTGPHRs, and even worse than that of GPEAHRs in some instances. As the number of candidate requests and conflicting requests increases, NSGA-II needs more iterations and evaluations to enhance its performance while the performance of KTGPHRs is rather stable. Besides, KTGPHRs and GPEAHRs can obtain solutions with higher values of f_2, which shows that they are able to generate scheduling plans with higher average imaging quality, and also indicates that they have a superior ability to preserve extreme solutions in the f_2 direction. In summary, although the overall performance of the proposed method is inferior to NSGA-II on the static MO-AEOSSP, it can provide better non-dominated solutions than MOEA/D in a very short period without iteration, ensuring a high level of imaging quality. Thus, KT-MOGP is still a quite competitive method for static MO-AEOSSP.

Table 8 presents the results of the Wilcoxon rank-sum test conducted between KTGPHR and other comparative methods on Set III. According to the obtained p-values, there is a statistically significant difference between KTGPHR and MOEA/D, but the performances of KTGPHR and NSGA-II are quite similar, indicating that NSGA-II is not significantly better than the proposed method. In addition, KTGPHR and GPEAHR

Table 4
HV and objective values of the non-dominated solutions obtained by KTGPHR and GEPAHR on Set I. The "f_1" and "f_2" in this table are the best objective values for each instance obtained by different methods. The best results in each instance are bolded.

Instance	KTGPHR			GPEAHR			Instance	KTGPHR			GPEAHR		
	HV	f_1	f_2	HV	f_1	f_2		HV	f_1	f_2	HV	f_1	f_2
100_1	**0.963**	**1.000**	**0.417**	0.947	0.973	0.374	250_1	**0.958**	**0.971**	0.629	0.951	0.971	**0.635**
100_2	**0.986**	**0.998**	**0.427**	0.979	0.980	0.378	250_2	**0.950**	**0.975**	0.559	0.932	0.957	**0.572**
100_3	**0.982**	**1.000**	**0.393**	0.956	0.978	0.379	250_3	**0.952**	0.979	0.601	0.948	**0.983**	**0.611**
100_4	**0.976**	**0.998**	**0.416**	0.973	0.992	0.385	250_4	**0.972**	**0.975**	**0.547**	0.970	0.974	0.547
100_5	**0.965**	**1.000**	**0.435**	0.963	0.986	0.400	250_5	**0.977**	**0.997**	0.565	0.974	0.996	**0.567**
150_1	**0.965**	**0.996**	**0.540**	0.950	0.979	0.520	300_1	**0.981**	**0.984**	**0.577**	0.975	0.978	0.569
150_2	**0.976**	**1.000**	**0.528**	0.963	0.973	0.486	300_2	**0.977**	**0.980**	**0.579**	0.967	0.970	0.554
150_3	**0.972**	**1.000**	**0.547**	0.965	0.988	0.504	300_3	**0.971**	**0.973**	**0.584**	0.963	0.969	0.568
150_4	**0.956**	**0.989**	**0.569**	0.954	0.984	0.530	300_4	**0.976**	**0.978**	0.548	0.967	0.970	**0.551**
150_5	**0.978**	**0.987**	**0.512**	0.977	0.987	0.487	300_5	**0.966**	**0.968**	**0.564**	0.963	0.966	0.532
200_1	**0.988**	**0.996**	**0.514**	0.966	0.975	0.495	350_1	**0.956**	**0.966**	**0.627**	0.949	0.953	0.602
200_2	**0.969**	**0.983**	**0.547**	0.968	0.975	0.509	350_2	**0.963**	**0.974**	**0.626**	0.942	0.940	0.591
200_3	**0.991**	**1.000**	0.485	0.972	0.993	**0.516**	350_3	**0.955**	**0.965**	**0.627**	0.941	0.941	0.601
200_4	**0.984**	**1.000**	0.538	0.968	0.986	**0.541**	350_4	**0.959**	**0.967**	**0.610**	0.952	0.953	0.578
200_5	**0.980**	**0.985**	0.505	0.969	0.974	**0.507**	350_5	**0.940**	**0.946**	**0.656**	0.930	0.929	0.625

Table 5
HV and objective values of the non-dominated solutions obtained by KTGPHR and GEPAHR on Set II. The "f_1" and "f_2" in this table are the best objective values for each instance obtained by different methods. The best results in each instance are bolded.

Instance	KTGPHR			GPEAHR			Instance	KTGPHR			GPEAHR		
	HV	f_1	f_2	HV	f_1	f_2		HV	f_1	f_2	HV	f_1	f_2
100_1	**0.730**	**0.703**	0.560	0.691	0.667	**0.600**	250_1	**0.462**	**0.409**	**0.712**	0.412	0.356	0.642
100_2	**0.675**	**0.651**	**0.583**	0.653	0.625	0.555	250_2	**0.467**	**0.415**	**0.714**	0.430	0.379	0.663
100_3	**0.739**	**0.716**	0.582	0.721	0.699	**0.608**	250_3	**0.476**	**0.424**	**0.718**	0.428	0.376	0.708
100_4	**0.659**	**0.629**	**0.617**	0.578	0.540	0.575	250_4	**0.461**	**0.410**	**0.735**	0.439	0.388	0.702
100_5	**0.682**	**0.654**	**0.567**	0.679	0.648	0.560	250_5	**0.489**	**0.439**	**0.722**	0.402	0.349	0.721
150_1	**0.573**	**0.536**	**0.673**	0.526	0.493	0.645	300_1	**0.431**	**0.375**	**0.778**	0.389	0.330	0.745
150_2	**0.596**	**0.563**	**0.723**	0.536	0.504	0.677	300_2	**0.462**	**0.410**	**0.761**	0.412	0.356	0.730
150_3	**0.621**	**0.587**	**0.739**	0.537	0.507	0.687	300_3	**0.432**	**0.376**	**0.756**	0.359	0.297	0.727
150_4	**0.551**	**0.510**	**0.677**	0.516	0.480	0.587	300_4	**0.449**	**0.395**	**0.743**	0.388	0.328	0.708
150_5	**0.609**	**0.577**	**0.704**	0.570	0.541	0.647	300_5	**0.452**	**0.399**	**0.748**	0.371	0.310	0.728
200_1	**0.501**	**0.453**	**0.739**	0.446	0.398	0.724	350_1	**0.398**	**0.343**	**0.806**	0.366	0.309	0.773
200_2	**0.527**	**0.485**	**0.723**	0.457	0.405	0.718	350_2	**0.402**	**0.345**	**0.774**	0.377	0.319	0.768
200_3	**0.527**	**0.481**	**0.713**	0.490	0.445	0.711	350_3	**0.416**	**0.362**	**0.778**	0.379	0.321	0.762
200_4	**0.485**	**0.436**	**0.735**	0.433	0.380	0.731	350_4	**0.413**	**0.359**	**0.775**	0.393	0.338	0.758
200_5	**0.490**	**0.441**	**0.747**	0.452	0.404	0.694	350_5	**0.413**	**0.356**	**0.783**	0.384	0.328	0.771

Table 6
p-values obtained by Wilcoxon rank-sum test for KTGPHR and GPEAHR on Set I and Set II.

Test Set	Set I	Set II
KTGPHR v.s. GPEAHR	7.566E−03	2.461E−02

perform relatively similarly in Set III. This is mainly because the idle monitoring mode of KT-MOGP is not working since there are no dynamically arrived requests in instances of Set III. The above results further demonstrate that KT-MOGP is a useful method for solving static MO-AEOSSP.

Normalization can map the range of different features to similar scales, which helps to avoid the excessive influence of certain features and keep balance among features. Besides, feature normalization can balance the scale of search space which benefits the convergence of the proposed method. To reveal the effectiveness of feature normalization, the HRs output by KT-MOGP with and without normalized feature terminals are compared (denoted as the normalization experiment). The normalization experiments are conducted on Set II. The output results are further statistically tested through the multi-problem Wilcoxon rank-sum test at the 5% significance level, as shown in Table 9. It is clear from the table that there exists a statistical difference between HRs generated with and without normalized feature terminals in terms of HV, f_1 and f_2, indicating that the feature normalization can obviously help improve the performance of the proposed GP method. Since the normalization may incur some time cost, we calculate the time consumption of outputting scheduling solutions with and without feature normalization. The results show that normalization increases the average cost of feature calculation by 17.6%, but the time of a single decision step still remains at the millisecond level. In other words, the scheduling processes adopting the above two feature calculating methods can be both regarded as instantaneous.

4.3.2. Single-objective scenarios

In the previous analysis, we compared the performance of the proposed algorithm with existing research in various multi-objective scenarios. In this subsection, we verify whether the proposed algorithm can continue to maintain its superiority in single-objective DAEOSSP scenarios through a series of comparative experiments. We choose to maximize the total profit (f_1) as the optimization objective, then evolve the modified KT-MOGP to obtain a set of HRs (denoted as GPHRs). The obtained GPHRs are compared with the HRs evolved by modified MO-GPEA (denoted as EAHRs) and several existing classical heuristic rules on Set I and Set II. Specifically, the algorithms used to evolve HRs in single-objective scenarios adopt the same experimental parameters and settings as mentioned above, the only difference is the selection operation. In this section, the GPHH-based algorithms only refer to the objective value (f_1 value) when selecting individuals, that is, the non-dominated sorting method is no longer adopted.

Tables 10 and 11 state the comparative results of the 6 heuristic rules on Set I and Set II respectively. Except for the GPHR and EAHR mentioned before, other comparison methods are introduced below.

Table 7
HV and objective values of the non-dominated solutions obtained by KTGPHR and GEPAHR on Set II. The "f_1" and "f_2" in this table are the best objective values for each instance obtained by different methods. The best results in each instance are bolded.

Instance	KTGPHR			GPEAHR			MOEA/D			NSGA-II		
	HV	f_1	f_2	HV	f_1	f_2	HV	f_1	f_2	HV	f_1	f_2
100_1	0.872	0.863	**0.666**	0.885	0.880	0.600	0.855	0.869	0.455	**0.937**	**0.939**	0.567
100_2	0.899	0.892	**0.692**	0.911	0.910	0.622	0.883	0.923	0.476	**0.960**	**0.966**	0.597
100_3	0.939	0.939	**0.685**	0.943	0.942	0.607	0.893	0.892	0.467	**0.951**	**0.955**	0.573
100_4	0.937	0.932	**0.643**	0.954	0.958	0.624	0.896	0.914	0.485	**0.968**	**0.976**	0.581
100_5	0.943	0.945	**0.696**	0.916	0.907	0.622	0.902	0.917	0.454	**0.963**	**0.979**	0.619
150_1	0.775	0.769	0.718	0.656	0.654	**0.729**	0.735	0.732	0.474	**0.811**	**0.816**	0.553
150_2	0.794	0.797	0.732	0.723	0.730	**0.764**	0.761	0.779	0.482	**0.848**	**0.835**	0.548
150_3	0.806	0.796	0.721	0.751	0.764	**0.751**	0.752	0.744	0.454	**0.824**	**0.809**	0.543
150_4	0.756	0.753	**0.736**	0.699	0.694	0.726	0.744	0.738	0.474	**0.809**	**0.791**	0.549
150_5	0.762	0.753	0.701	0.747	0.748	**0.712**	0.715	0.701	0.455	**0.793**	**0.786**	0.579
200_1	0.689	**0.671**	0.713	0.632	0.604	**0.728**	0.626	0.610	0.488	**0.696**	0.667	0.585
200_2	**0.695**	**0.671**	**0.754**	0.681	0.659	0.719	0.618	0.603	0.508	0.688	0.666	0.599
200_3	0.671	**0.651**	**0.727**	0.618	0.585	0.715	0.628	0.616	0.487	**0.679**	0.650	0.559
200_4	0.694	0.678	**0.745**	0.644	0.618	0.712	0.640	0.623	0.464	**0.732**	**0.723**	0.574
200_5	0.696	0.673	0.673	0.666	0.646	**0.702**	0.643	0.634	0.452	**0.744**	**0.735**	0.565
250_1	0.615	0.592	**0.763**	0.598	0.571	0.730	0.563	0.539	0.463	**0.642**	**0.618**	0.568
250_2	0.613	0.578	0.751	0.621	0.602	**0.778**	0.583	0.546	0.511	**0.659**	**0.638**	0.604
250_3	0.602	0.571	0.785	0.603	0.583	**0.814**	0.575	0.548	0.477	**0.663**	**0.645**	0.597
250_4	0.607	0.573	**0.760**	0.618	0.599	0.760	0.575	0.550	0.470	**0.661**	**0.642**	0.568
250_5	0.606	0.573	**0.769**	0.606	0.582	0.748	0.569	0.547	0.473	**0.665**	**0.648**	0.570
300_1	0.580	0.539	0.725	0.555	0.515	**0.743**	0.527	0.497	0.473	**0.605**	**0.577**	0.581
300_2	0.576	0.535	**0.761**	0.580	0.542	0.714	0.538	0.513	0.493	**0.595**	**0.569**	0.567
300_3	0.584	0.544	**0.759**	0.564	0.522	0.722	0.516	0.482	0.465	**0.610**	**0.588**	0.570
300_4	0.574	0.537	**0.736**	0.562	0.521	0.713	0.540	0.509	0.480	**0.606**	**0.573**	0.557
300_5	0.581	0.542	**0.747**	0.549	0.509	0.721	0.512	0.498	0.500	**0.587**	**0.551**	0.570
350_1	**0.578**	**0.548**	**0.788**	0.516	0.474	0.782	0.473	0.435	0.501	0.541	0.506	0.566
350_2	**0.583**	**0.556**	0.808	0.531	0.498	**0.819**	0.468	0.418	0.499	0.550	0.519	0.597
350_3	**0.598**	**0.573**	**0.773**	0.530	0.490	0.734	0.475	0.433	0.486	0.549	0.519	0.591
350_4	**0.591**	**0.564**	**0.780**	0.541	0.506	0.772	0.477	0.435	0.472	0.534	0.492	0.593
350_5	**0.580**	**0.548**	0.776	0.532	0.496	**0.788**	0.468	0.419	0.467	0.541	0.503	0.586

Table 8
p-values obtained by Wilcoxon rank-sum test for KTGPHR and comparative methods on Set III.

Method	NSGA-II	MOEA/D	GPEAHR
KTGPHR v.s.	0.5642	4.593E−02	0.2804

Table 9
p-values obtained by Wilcoxon rank-sum test for normalization experiment.

	HV	f_1	f_2
p-value	3.137E−03	1.899E−03	3.022E−15

- PF [60]: Sort the candidate requests in descending order of profit, then attempt to insert them in order.
- RF [60]: Randomly sort the candidate requests, then attempt to insert them in order. We take the average results of 10 independent runs for each instance.
- PWF [61]: Sort the candidate requests in descending order of profit. For the requests have the same profit, they are sorted from early to late by their start time of VTW in the current orbit.
- The FIFO rule proposed by Bianchessi and Righini [62].

From the given two tables, we can see that the performance of GPHR can outperform all comparison methods, followed by EAHR and FIFO, while PWF, PF and RF perform worst. By analyzing the experimental results, it can be easily figured out that the start time of the OTW (OTWS) is an important feature for single-objective DAEOSSP, which is why methods considering OTWS (GPHR, EAHR, and FIFO) perform better than others. To provide a more intuitive view of the performance difference, we perform the Wilcoxon rank-sum test on the obtained results, as shown in Table 12. According to the obtained *p*-value, there are statistically significant differences between GPHR and other manually designed heuristics. For EAHR, GPHR performs significantly better than it on Set I, but among Set II they perform quite similarly, indicating that EAHR is also competitive for dealing with conflicting requests.

4.4. Effectiveness of knowledge transfer-based initialization

To validate the effectiveness of the proposed knowledge transfer-based initialization, we pick two representative scenario scales, 150-request scenario and 350-request scenario, and test the training and testing performance of the algorithm with and without knowledge transfer based-initialization. The algorithm without knowledge transfer-based initialization is denoted as NKT-MOGP, other setups remain the same as that of KT-MOGP.

The final non-dominated solutions of different training scenarios are depicted in Fig. 7. Obviously, the non-dominated solutions obtained by KT-MOGP can dominate most of the non-dominated individuals generated by NKT-MOGP, which indicates that the knowledge transfer-based initialization has a significant effect on improving the convergence of the algorithm. In addition, the PFs obtained by KT-MOGP also have a better distribution in the solution space than those obtained by NKT-MOGP, which reveals the better quality of spreading and diversity.

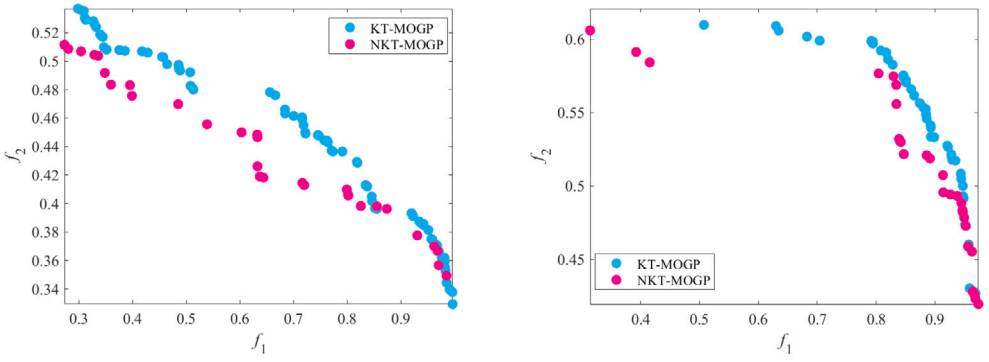
Table 13 states the HV values of HR sets generated by KT-MOGP and NKT-MOGP (denoted as KTHRs and NKTHRs) among the extended Set of Set I (denoted as Eset), which generate ten different instances for each scenario size using the same instance generating method as Set I adopts. As KTHRs always get better HVs which imply better overall performance, the results illustrate that KTHRs generally outperform NKTHRs, further demonstrating the effectiveness of the proposed knowledge transfer-based initialization. Besides, we can also figure out that although knowledge transfer-based initialization method can further improve the quality of the obtained HRs, the degree of improvement is not very significant in some instances. Combined with the

Table 10
The f_1 values obtained by GPHRs and other comparative methods on Set I. The best result for each instance are bolded.

Instance	PF	RF	PWF	FIFO	EAHR	GPHR	Instance	PF	RF	PWF	FIFO	EAHR	GPHR
100_1	0.578	0.426	0.627	0.955	0.976	**1.000**	250_1	0.316	0.252	0.439	0.962	0.970	**0.974**
100_2	0.477	0.409	0.556	0.985	0.977	**0.998**	250_2	0.352	0.250	0.389	0.967	0.975	**0.977**
100_3	0.540	0.436	0.584	0.978	0.978	**0.996**	250_3	0.296	0.235	0.353	0.967	0.976	**0.989**
100_4	0.488	0.476	0.563	0.986	0.986	**1.000**	250_4	0.423	0.234	0.414	0.990	0.984	**0.991**
100_5	0.484	0.429	0.540	0.991	0.980	**0.995**	250_5	0.417	0.248	0.362	0.954	0.957	**0.970**
150_1	0.467	0.344	0.541	0.990	0.999	**1.000**	300_1	0.257	0.207	0.354	0.950	0.976	**0.983**
150_2	0.357	0.332	0.462	0.984	0.987	**0.989**	300_2	0.302	0.203	0.400	0.977	0.987	**0.995**
150_3	0.406	0.307	0.456	0.967	0.976	**0.988**	300_3	0.251	0.221	0.303	0.949	0.962	**0.964**
150_4	0.478	0.366	0.506	0.994	0.994	**1.000**	300_4	0.317	0.210	0.375	0.965	0.975	**0.978**
150_5	0.424	0.342	0.524	0.981	0.981	**0.981**	300_5	0.271	0.215	0.368	0.967	0.974	**0.975**
200_1	0.358	0.287	0.426	0.977	0.992	**1.000**	350_1	0.225	0.178	0.348	0.916	0.936	**0.936**
200_2	0.352	0.281	0.434	0.996	**1.000**	**1.000**	350_2	0.266	0.188	0.411	0.924	0.942	**0.950**
200_3	0.406	0.295	0.450	0.986	**0.991**	**0.991**	350_3	0.252	0.170	0.324	0.940	0.954	**0.961**
200_4	0.423	0.280	0.460	0.992	0.991	**1.000**	350_4	0.272	0.187	0.341	0.967	0.973	**0.974**
200_5	0.417	0.299	0.499	0.977	0.965	**0.991**	350_5	0.228	0.211	0.310	0.943	0.961	**0.967**

Table 11
The f_1 values obtained by GPHRs and other comparative methods on Set II. The best result for each instance are bolded.

Instance	PF	RF	PWF	FIFO	EAHR	GPHR	Instance	PF	RF	PWF	FIFO	EAHR	GPHR
100_1	0.431	0.320	0.506	0.676	0.773	**0.781**	250_1	0.234	0.164	0.298	0.358	0.449	**0.485**
100_2	0.433	0.304	0.474	0.596	0.713	**0.754**	250_2	0.240	0.195	0.319	0.364	0.504	**0.525**
100_3	0.471	0.308	0.561	0.735	0.809	**0.816**	250_3	0.312	0.177	0.366	0.372	0.467	**0.519**
100_4	0.419	0.331	0.509	0.733	0.820	**0.826**	250_4	0.294	0.176	0.354	0.382	0.482	**0.521**
100_5	0.501	0.267	0.563	0.676	0.757	**0.763**	250_5	0.216	0.180	0.337	0.381	0.496	**0.510**
150_1	0.307	0.235	0.376	0.456	0.596	**0.600**	300_1	0.231	0.149	0.231	0.317	0.440	**0.461**
150_2	0.327	0.253	0.451	0.558	0.648	**0.674**	300_2	0.195	0.155	0.195	0.321	0.421	**0.430**
150_3	0.354	0.262	0.434	0.491	0.603	**0.651**	300_3	0.260	0.139	0.260	0.324	0.426	**0.450**
150_4	0.356	0.236	0.398	0.558	0.627	**0.713**	300_4	0.236	0.152	0.236	0.313	0.454	**0.472**
150_5	0.312	0.211	0.357	0.540	0.638	**0.663**	300_5	0.190	0.160	0.190	0.309	0.430	**0.460**
200_1	0.443	0.225	0.343	0.443	0.578	**0.621**	350_1	0.197	0.127	0.273	0.304	0.395	**0.411**
200_2	0.420	0.203	0.395	0.420	0.518	**0.544**	350_2	0.184	0.130	0.263	0.298	0.383	**0.385**
200_3	0.423	0.199	0.370	0.423	0.536	**0.548**	350_3	0.216	0.133	0.277	0.293	0.391	**0.404**
200_4	0.404	0.189	0.379	0.404	0.546	**0.575**	350_4	0.214	0.135	0.340	0.283	0.368	**0.370**
200_5	0.426	0.188	0.356	0.426	0.502	**0.570**	350_5	0.183	0.128	0.304	0.282	0.377	**0.382**



(a) 150-request scenario (b) 350-request scenario

Fig. 7. Non-dominated solutions reserved in different scenarios of MO-DAEOSSP using KT-MOGP and NKT-MOGP.

Table 12
p-values obtained by Wilcoxon rank-sum test for GPHR and comparative methods on Set I and Set II.

GPHR v.s.	PF	RF	PWF	FIFO	EAHR
Set I	2.895E−11	2.902E−11	2.902E−11	3.058E−3	3.058E−2
Set II	9.660E−09	3.020E−11	8.697E−08	4.46E−4	0.412

training performance, the reason is obvious. The knowledge transfer-based initialization can help accelerate convergence in the early stage of KT-MOGP, whereas NKT-MOGP can also obtain relatively competitive HRs due to its superior operators and heuristic-based simulation stage as the number of iterations increases.

To further illustrate the above statement, and to figure out the extent of improvement brought by the transfer mechanism, the additional experiments are conducted as follows: A set of HRs outputted by KT-MOGP based on Set II are obtained first. Note that HRs are generated taking the 100-request scenario as the source domain for extraction, thus for all training scenarios the transfer knowledge is extracted based on 100-request output solutions. The newly obtained HRs (denoted as KTHR_100) are then compared with KTHR and NKTHR on Set II.

Table 14 shows the HV values generated by KTHR, NKTHR and KTHR_100 on Set II. It is obvious that KTHR still performs best in most of the instances, indicating the effectiveness of the proposed transfer mechanism. However, when comparing the results of the remaining two methods, we find that in most instances KTHR_100 performs even worse than NKTHR. The aforementioned phenomenon indicates that a

Table 13
The HV values obtained by KTHRs and NKTHRs on Eset. The best result for each instance are bolded.

Instance	KTHR	NKTHR	Instance	KTHR	NKTHR
150_1	**0.965**	0.949	350_1	**0.938**	0.935
150_2	**0.978**	0.973	350_2	**0.957**	0.951
150_3	**0.983**	0.979	350_3	**0.963**	0.955
150_4	**0.976**	0.974	350_4	**0.940**	0.928
150_5	**0.972**	0.956	350_5	**0.949**	0.945
150_6	**0.952**	0.950	350_6	**0.960**	0.951
150_7	**0.972**	0.965	350_7	**0.945**	0.930
150_8	**0.989**	0.982	350_8	**0.952**	0.952
150_9	**0.966**	0.957	350_9	0.955	**0.958**
150_10	**0.962**	0.947	350_10	**0.959**	0.949

Table 14
The HV values obtained by KTHRs, NKTHRs and KTHR_100s on Set II. The best result for each instance are bolded.

Instance	KTHR	NKTHR	KTHR_100
150_1	**0.506**	0.462	**0.506**
150_2	**0.536**	0.534	**0.536**
150_3	**0.554**	0.532	**0.554**
150_4	**0.522**	0.498	**0.522**
150_5	**0.552**	0.538	**0.552**
200_1	**0.587**	0.498	0.490
200_2	**0.587**	0.498	0.490
200_3	**0.558**	0.528	0.488
200_4	**0.577**	0.496	0.469
200_5	**0.582**	0.474	0.478
250_1	0.437	**0.438**	0.425
250_2	0.413	0.405	**0.422**
250_3	**0.515**	0.468	0.445
250_4	0.449	**0.455**	0.413
250_5	**0.430**	0.424	0.411
300_1	**0.454**	0.382	0.380
300_2	**0.466**	0.374	0.391
300_3	**0.434**	0.377	0.367
300_4	**0.474**	0.385	0.409
300_5	**0.473**	0.395	0.406
350_1	**0.426**	0.382	0.330
350_2	**0.426**	0.382	0.330
350_3	**0.439**	0.377	0.336
350_4	**0.426**	0.385	0.351
350_5	**0.447**	0.395	0.387

Table 15
p-values obtained by Wilcoxon rank-sum test for KTHR, NKTHR and KTHR_100 on Set II.

	KTHR	NKTHR	KTHR_100
KTHR	/	1.197E−02	5.685E−03
KTHR_100	1.197E−02	9.925E−01	

reasonable transfer and extraction of suitable knowledge can help improve the performance of output HRs, while unreasonable knowledge extraction and utilization may even lead to negative transfer which causes performance degradation. To provide an intuitive view of the performance difference, the Wilcoxon rank-sum test is conducted and shown in Table 15. According to the *p*-value, there exist statistically significant differences between KTHR and other methods, but KTHR_100 is not significantly better than NKTHR, further verifying our analysis.

4.5. Efficiency of the proposed method

In this section, the efficiency of the proposed method in this work is analyzed from two aspects, the training procedure and the testing procedure.

The efficiency of the training procedure relies on two main components: the heuristic template and the knowledge transfer. To verify the effectiveness of the heuristic template, we first exclude the knowledge transfer initialization and generate the HRs. The average runtimes for one iteration under various-size instances are shown in Fig. 8. It is obvious from the figure that the computing cost of KT-MOGP has an apparent tendency related to the scale of the instance. Most of the runtime is spent on the evaluation of HRs. Although the scheduling at each decision step is real-time, the evaluation of an HR needs to output the corresponding solution and fitness values of a given instance, thus the entire simulation process needs to be completed. We also calculate the time cost of HR generation, and the average time spent generating an HR is 0.124 ms. Regarding knowledge transfer, it has been mentioned before that the design of matrix representation and rough removal of individuals can help improve the efficiency a lot. The average time of rough removal is 0.0245 s while the average time cost of removal based on phenotypic characterization is 1.33 s. For feature contribution extraction, calculating one feature contribution cost average 0.5 s based on one individual.

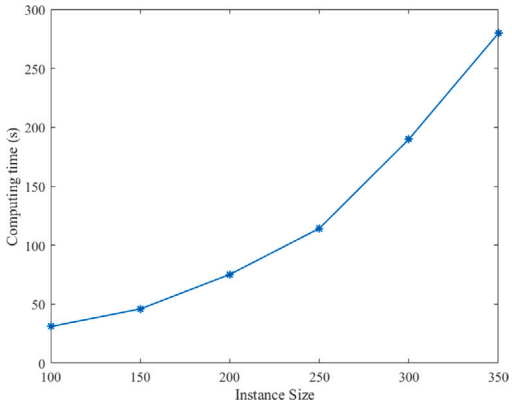


Fig. 8. The computing time (s) for one iteration on instance of 100–350 requests using KT-MOGP.

Table 16
The average number of nodes in HRs obtained by instances of various sizes.

Instance size	100	150	200	250	300	350
Tree Size	56.06	61.24	67.29	56.06	77.34	69.02

The efficiency of the testing procedure is mainly related to the average time spent by each decision when scheduling requests. According to the experimental results, the proposed method makes each decision in 0.025 ms, indicating the decision-making in scheduling process is almost instantaneous.

4.6. Analysis of evolved rules

In this section, the evolved HRs are analyzed from several different aspects to help better understand the proposed method.

Table 16 shows the average number of nodes in the output HRs generated by instances of various sizes. Note that the node contains both the feature terminal and the operator. We find from the table that the size of HRs does not show a clear linear correlation with the scenario size, and there is no excessive expansion of tree structure, which indicates the effectiveness of the individual generation and evolution operations.

We select the 150-request scenario and sort the obtained HRs by f_1 and f_2, respectively. The individuals who perform best in f_1 and f_2 are selected and shown in Table 17. Among the output individuals, HR1 can get scheduling solution with the best f_1 value. By analyzing the features in HR1, it can be found that CONTWL and P are frequently used and play important roles, revealing that HR1 tends to schedule requests with higher priority and less conflict. OTWS, SL and WT are also important terminals in HR1. Thus HR1 is more likely to schedule requests early in the timeline. HR2 focuses more on f_2 and makes a

Table 17
Examples of evolved individuals.

HR1 : $\frac{1}{\mathrm{SL}\times\mathrm{OTWN}} \times (\mathrm{P} \times (1 + \mathrm{COTWL}) + \mathrm{COTWL} + \mathrm{OTWS} - \frac{\mathrm{WT}}{(\mathrm{COTWL}-\mathrm{OTWL})\times\mathrm{BQOT}\times\mathrm{SL}})$
HR2 : $\frac{1}{\mathrm{OTWN}} \div (\mathrm{OTWS} \times \mathrm{SL} \times (\mathrm{BQOT} + \mathrm{REPT}) \times \max(\mathrm{APR} + \mathrm{SL}, \mathrm{BQOT}) \times \mathrm{BQOT} \times \max(\mathrm{OTWS}, \mathrm{WQOA}))$

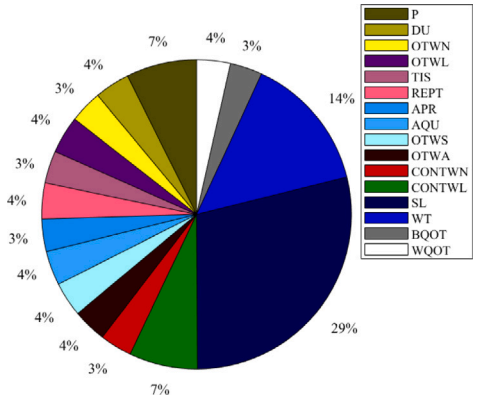


Fig. 9. The contribution of each feature terminal in the HRs obtained by 150-request scenario.

trade-off between scheduling profit and observation quality, which is verified by its structure. BQOT is the most frequently used terminal in HR2, indicating that HR2 tends to consider the obtained image quality more. APR and WQOA, two terminals about image quality, are adopted in HR2. Besides, SL, OTWS and OTWN are also important in HR2, demonstrating that HR2 is still likely to schedule requests early in the timeline because of the dynamic characteristics of MO-DAEOSSP.

As mentioned in Section 3.4, feature contribution is another important factor for knowledge transfer since it decides the probability of each node when generating new individuals. Fig. 9 shows the overall contribution of each feature terminal in the obtained set of 150-request HRs. SL and WT are features which contribute most (29% and 14%) and are more likely to be chosen when generating new individuals by knowledge-transfer based initialization method. The contributions of P and CONTWL are both 7%, while the remaining feature terminals are basically at the same level (3% to 4%). The above phenomenon indicates that even if HRs have different preferences when scheduling requests, request timeliness is always an important component which influences the output solutions.

5. Conclusion and future work

A multi-objective agile earth observation satellite scheduling problem considering dynamic arrival observation requests (MO-DAEOSSP) is studied in this paper, where the total profit and the average image quality of scheduled requests are optimized simultaneously. Manually designed heuristics used for previous research highly rely on expert knowledge while iterative based methods can get high-quality solutions but are always quite time-consuming and unacceptable for real-time scheduling. Inspired by the successful application of genetic programming based hyper-heuristics in manufacturing and producing domain, we propose a knowledge-transfer based multi-objective genetic programming algorithm (KT-MOGP) in this work. A heuristic-based simulation equipped with idle monitoring and request execution modules is first designed to form the simulation procedure for KT-MOGP, evaluating the fitness of heuristic rules (HRs). Then a multi-objective genetic programming hyper heuristic (GPHH) framework is proposed to evolve the HRs. KT-MOGP applies knowledge-transfer methods to enhance the efficiency and accelerate convergence, utilizing elite individuals and knowledge extracted from non-dominated solutions.

We conduct extensive experiments on the proposed algorithm in multi-objective and single-objective scenarios, including comparisons with existing GPHH-based methods, comparisons with other multi-objective evolutionary algorithms (MOEAs) in static scenarios, and comparisons with other construction heuristics in single-objective DAEOSSP scenarios. Experimental results state that KT-MOGP is able to evolve HRs with competitive performance for both multi-objective scenarios and single-objective scenarios, and the evolved HRs are competitive for both static and dynamic scenarios. Besides, experiments also verify the effectiveness of the proposed knowledge transfer-based initialization.

How to generate or select representative instances for the training of the GPHH based algorithm is a problem worth researching, since it can greatly improve the efficiency for the evolution stage of the algorithm. Besides, we will consider the address for more complex MO-DAEOSSP, especially for the multi-satellite problem in the future. Different from the single satellite scenarios, several satellites should first adopt a collaborative planning strategy to allocate requests for each satellite. The proposed method should be modified to adapt to the multi-satellite scenario. By combining the collaborative planning strategy and the GPHH based method, a framework that can address large-scale satellite scheduling problem can be proposed. Furthermore, hierarchy mechanisms for multi-satellite optimization, such as adaptive strategies and parameter tuning can be another focus of this study.

CRediT authorship contribution statement

Luona Wei: Conceptualization, Methodology, Software, Writing – original draft. **Ming Chen:** Writing – review & editing, Validation. **Lining Xing:** Formal analysis, Investigation. **Qian Wan:** Analysis. **Yanjie Song:** Investigation. **Yuning Chen:** Resources. **Yingwu Chen:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work is partially supported by the National Natural Science Foundation of China [Grant Number 62307015, 71701203, 72001212], Hubei Provincial Natural Science Foundation of China [Grant Number: 2023AFB295], Knowledge Innovation Program of Wuhan-Shugung Project, China [Grant Number: 2023010201020390], Hunan Postgraduate Research Innovation Project, China [Grant Number: CX20210034] and China Postdoctoral Science Foundation [Grant Number: 2023M741304].

References

[1] X. Wang, G. Wu, L. Xing, W. Pedrycz, Agile Earth observation satellite scheduling over 20 years: formulations, methods and future directions, 2020, arXiv preprint arXiv:2003.06169.

[2] J. Zhang, L. Xing, An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem, Comput. Oper. Res. 139 (2022) 105626.

[3] D. Habet, M. Vasquez, Saturated and consistent neighborhood for selecting and scheduling photographs of agile earth observing satellite, in: Proc. 5th Metaheuristics Int. Conf, Citeseer, 2003, pp. 1–6.

[4] D. Habet, M. Vasquez, Y. Vimont, Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite, Comput. Optim. Appl. 47 (2) (2010) 307–333.

[5] C. Han, X. Wang, G. Song, R. Leus, Scheduling multiple agile Earth observation satellites with multiple observations, 2018, arXiv preprint arXiv:1812.00203.

[6] J.-F. Cordeau, G. Laporte, Maximizing the value of an earth observation satellite orbit, J. Oper. Res. Soc. 56 (8) (2005) 962–968.

[7] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, N. Bataille, How to manage the new generation of agile earth observation satellites, in: Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, Citeseer, 2000, pp. 1–10.

[8] S. Peng, H. Chen, C. Du, J. Li, N. Jing, Onboard observation task planning for an autonomous earth observation satellite using long short-term memory, IEEE Access 6 (2018) 65118–65129.

[9] G. Peng, R. Dewil, C. Verbeeck, A. Gunawan, L. Xing, P. Vansteenwegen, Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times, Comput. Oper. Res. 111 (2019) 84–98.

[10] L. He, X. Liu, G. Laporte, Y. Chen, Y. Chen, An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling, Comput. Oper. Res. 100 (2018) 12–25.

[11] X. Liu, G. Laporte, Y. Chen, R. He, An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time, Comput. Oper. Res. 86 (2017) 41–53.

[12] M. Chen, J. Wen, Y.-J. Song, L.-n. Xing, Y.-w. Chen, A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem, Swarm Evol. Comput. 65 (2021) 100912.

[13] G. Wu, M. Ma, J. Zhu, D. Qiu, Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks, J. Syst. Eng. Electron. 23 (5) (2012) 723–733.

[14] Y.-J. Song, Z.-S. Zhang, B.-Y. Song, Y.-W. Chen, Improved genetic algorithm with local search for satellite range scheduling system and its application in environmental monitoring, Sustain. Comput. Inf. Syst. 21 (2019) 19–27.

[15] L. He, X.-L. Liu, Y.-W. Chen, L.-N. Xing, K. Liu, Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment, Adv. Space Res. 63 (2) (2019) 897–912.

[16] P. Wang, G. Reinelt, P. Gao, Y. Tan, A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation, Comput. Ind. Eng. 61 (2) (2011) 322–335.

[17] R. Xu, H. Chen, X. Liang, H. Wang, Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization, Expert Syst. Appl. 51 (2016) 195–206.

[18] P. Tangpattanakul, N. Jozefowiez, P. Lopez, A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite, European J. Oper. Res. 245 (2) (2015) 542–554.

[19] L. Li, F. Yao, N. Jing, M. Emmerich, Preference incorporation to solve multi-objective mission planning of agile earth observation satellites, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017, pp. 1366–1373.

[20] L. Li, H. Chen, J. Li, N. Jing, M. Emmerich, Preference-based evolutionary many-objective optimization for agile satellite mission planning, IEEE Access 6 (2018) 40963–40978.

[21] J. Wang, N. Jing, J. Li, Z.H. Chen, A multi-objective imaging scheduling approach for earth observing satellites, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, 2007, pp. 2211–2218.

[22] L. Wei, L. Xing, Q. Wan, Y. Song, Y. Chen, A multi-objective memetic approach for time-dependent agile earth observation satellite scheduling problem, Comput. Ind. Eng. 159 (2021) 107530.

[23] X. Zhai, X. Niu, H. Tang, L. Wu, Y. Shen, Robust satellite scheduling approach for dynamic emergency tasks, Math. Probl. Eng. 2015 (2015).

[24] J. Cui, X. Zhang, Application of a multi-satellite dynamic mission scheduling model based on mission priority in emergency response, Sensors 19 (6) (2019) 1430.

[25] G. Povéda, O. Regnier-Coudert, F. Teichteil-Königsbuch, G. Dupont, A. Arnold, J. Guerra, M. Picard, Evolutionary approaches to dynamic earth observation satellites mission planning under uncertainty, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 1302–1310.

[26] W. Yang, L. He, X. Liu, Y. Chen, Onboard coordination and scheduling of multiple autonomous satellites in an uncertain environment, Adv. Space Res. 68 (11) (2021) 4505–4524.

[27] Y. Li, Planning and Scheduling Spacecraft Observation Under Uncertamties in Dynamic Environment, Harbin Institute of Technology, 2008.

[28] L.A. Kramer, S.F. Smith, Task swapping: Making space in schedules for space, in: Proc. Fourth International Workshop on Planning and Scheduling for Space, IWPSS-04, Citeseer, 2004.

[29] J. Wang, X. Zhu, L.T. Yang, J. Zhu, M. Ma, Towards dynamic real-time scheduling for multiple earth observation satellites, J. Comput. System Sci. 81 (1) (2015) 110–124.

[30] H. Fan, H. Xiong, M. Goh, Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints, Comput. Oper. Res. 134 (2021) 105401.

[31] S. Nguyen, Y. Mei, M. Zhang, Genetic programming for production scheduling: a survey with a unified framework, Complex Intell. Syst. 3 (1) (2017) 41–66.

[32] H.R. Topcuoglu, A. Ucar, L. Altin, A hyper-heuristic based framework for dynamic optimization problems, Appl. Soft Comput. 19 (2014) 236–251.

[33] D. Jakobović, K. Marasović, Evolving priority scheduling heuristics with genetic programming, Appl. Soft Comput. 12 (9) (2012) 2781–2789.

[34] M. Đurasević, D. Jakobović, K. Knežević, Adaptive scheduling on unrelated machines with genetic programming, Appl. Soft Comput. 48 (2016) 419–430.

[35] S. Nguyen, Y. Mei, B. Xue, M. Zhang, A hybrid genetic programming algorithm for automated design of dispatching rules, Evol. Comput. 27 (3) (2019) 467–496.

[36] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling, IEEE Trans. Cybern. 51 (4) (2020) 1797–1811.

[37] B. Xu, Y. Mei, Y. Wang, Z. Ji, M. Zhang, Genetic programming with delayed routing for multiobjective dynamic flexible job shop scheduling, Evol. Comput. 29 (1) (2021) 75–105.

[38] J.C. Tay, N.B. Ho, Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems, Comput. Ind. Eng. 54 (3) (2008) 453–473.

[39] D. Yska, Y. Mei, M. Zhang, Genetic programming hyper-heuristic with co-operative coevolution for dynamic flexible job shop scheduling, in: European Conference on Genetic Programming, Springer, 2018, pp. 306–321.

[40] F. Zhang, Y. Mei, M. Zhang, Evolving dispatching rules for multi-objective dynamic flexible job shop scheduling via genetic programming hyper-heuristics, in: 2019 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2019, pp. 1366–1373.

[41] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job-shop scheduling, IEEE Trans. Cybern. (2022).

[42] F. Zhang, Y. Chen, Y. Chen, Evolving constructive heuristics for agile earth observing satellite scheduling problem with genetic programming, in: 2018 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2018, pp. 1–7.

[43] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[44] T.T.H. Dinh, T.H. Chu, Q.U. Nguyen, Transfer learning in genetic programming, in: 2015 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2015, pp. 1145–1151.

[45] T. Hildebrandt, J. Branke, On using surrogates with genetic programming, Evol. Comput. 23 (3) (2015) 343–367.

[46] Y. Mei, S. Nguyen, B. Xue, M. Zhang, An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming, IEEE Trans. Emerg. Top. Comput. Intell. 1 (5) (2017) 339–353.

[47] M.A. Ardeh, Y. Mei, M. Zhang, Genetic programming with knowledge transfer and guided search for uncertain capacitated arc routing problem, IEEE Trans. Evol. Comput. (2021).

[48] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, K.C. Tan, Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling, IEEE Trans. Evol. Comput. PP (99) (2021) 1.

[49] Y. Mei, M. Zhang, S. Nyugen, Feature selection in evolving job shop dispatching rules with genetic programming, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, 2016, pp. 365–372.

[50] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Comput. Intell. Mag. 12 (4) (2017) 73–87.

[51] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.

[52] J.G. Falcón-Cardona, M.T. Emmerich, C.A.C. Coello, On the construction of Pareto-compliant quality indicators, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp. 2024–2027.

[53] T. Friedrich, K. Bringmann, T. Voß, C. Igel, The logarithmic hypervolume indicator, in: Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms, 2011, pp. 81–92.

[54] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration, in: Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings, Vol. 4, Springer, 2007, pp. 862–876.

[55] E. Zitzler, J. Knowles, L. Thiele, Quality assessment of pareto set approximations, Multiobjective Optim. Interact. Evol. Approaches (2008) 373–404.

[56] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, L. Salomon, Performance indicators in multiobjective optimization, European J. Oper. Res. 292 (2) (2021) 397–422.

[57] N. Riquelme, C. Von Lücken, B. Baran, Performance metrics in multi-objective optimization, in: 2015 Latin American Computing Conference, CLEI, IEEE, 2015, pp. 1–11.

[58] F. Wilcoxon, Individual comparisons by ranking methods, in: Breakthroughs in Statistics, Springer, 1992, pp. 196–202.
[59] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[60] J. Chen, M. Chen, J. Wen, L. He, X. Liu, A heuristic construction neural network method for the time-dependent agile earth observation satellite scheduling problem, Mathematics 10 (19) (2022) 3498.
[61] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, N. Bataille, Selecting and scheduling observations of agile satellites, Aerosp. Sci. Technol. 6 (5) (2002) 367–381.
[62] N. Bianchessi, G. Righini, Planning and scheduling algorithms for the COSMO-SkyMed constellation, Aerosp. Sci. Technol. 12 (7) (2008) 535–544.