A Dual-Layer Clustering Guided Genetic Algorithm for Multiobjective Multitype Satellite Observation Scheduling

[1]Yong-Chao Chen, [1]Xiao-Fang Liu*, [1]Zhi-Hui Zhan, [1,2]Jun Zhang
College of Artificial Intelligence
[1]*Nankai University*, Tianjin, China
[2]*Hanyang University*, ERICA, Ansan, South Korea
liuxiaofang@nankai.edu.cn

***Abstract*—Earth observation satellites have been widely applied and become indispensable in various domains. The collaboration of optical, synthetic aperture radar, and electromagnetic satellites enables diverse mission requirements through complementary capabilities. However, scheduling observation tasks across multiple satellite types to satisfy diverse objectives still remains a critical challenge, especially on large-scale instances. This paper focuses on two optimization objectives, i.e., maximizing the total profit of scheduled tasks and maximizing the average profit efficiency across satellites. To solve this problem, a dual-layer clustering guided genetic algorithm is proposed to obtain high-quality and well-distributed solution sets. The algorithm decouples the scheduling process into two layers, i.e., task sequence construction and time window assignment. The two layers dynamically switch based on the performance improvement to generate new task sequences or new time-window assignments. At both of task and time-window layers, clustering is performed based on characteristics and historical selection frequencies to guide the crossover and mutation for improving profits and resource utilization. In addition, a time-window allocation strategy is adopted in both layers to improve solution quality by utilizing historical information and current satellite resource states. Experimental results on fifteen instances with up to 1300 tasks demonstrate that the proposed algorithm outperforms state-of-the-art algorithms in terms of convergence and diversity.**

***Index Terms*—satellites observation scheduling, multiobjective optimization, genetic algorithm, clustering, resource allocation**

I. Introduction

The rapid development of aerospace technology has propelled the deployment and application of Earth observation satellites (EOSs). With the expanding application of satellite observation technology in domains such as disaster emergency response, resource and environmental monitoring, and military intelligence reconnaissance, the demand for observation tasks has experienced explosive growth [1]. However, spaceborne resources remain extremely limited relative to the ever increasing observation demands, rendering the efficient scheduling of satellite resources to maximize system efficiency a critical problem that urgently requires resolution.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62476141, in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.RS-2025-00555463, RS-2025-25456394 and RS-2025-25460542), in part by the Tianjin Top Scientist Studio Project under Grant 24JRRCRC00030, and in part by the Tianjin Belt and Road Joint Laboratory under Grant 24PTLYHZ00250. (*Corresponding author: Xiao-Fang Liu*)

EOSs can observe ground targets only during specific visible time windows (VTWs) when they pass overhead. Each target typically has multiple discrete VTWs corresponding to different satellite overpasses. Since observation requests usually exceed satellite capacity, the Earth observation satellite scheduling problem (EOSSP) aims to select tasks and assign appropriate VTWs to maximize overall benefit while satisfying operational constraints. Currently operational EOSs can be classified into three categories based on their payload types [2]. Optical satellites provide high-resolution images but are weather-dependent, SAR satellites enable all-weather observation but face geometric distortion, and EM satellites detect signals but lack high-resolution imagery. An observation system comprising multiple satellite types can accommodate diverse mission requirements through complementary payloads. Existing research has demonstrated that the EOSSP is NP-hard [3].

Evolutionary algorithms [4], including evolutionary algorithm based on decomposition (MOEA/D) [5], genetic algorithm (GA) [6], and particle swarm optimization (PSO) [7], have been widely applied to combinatorial optimization problems due to their capability of exploring complex solution spaces [8]. However, current research exhibits limitations. Existing studies primarily focus on single-objective optimization for profit maximization, lacking systematic mechanisms for multi-objective optimization in practical scenarios. Although existing algorithms have advanced in optimizing task execution priority sequences [9], they rely on simple greedy heuristics when generating scheduling plans for the task sequences (i.e., allocating time windows to tasks), neglecting correlations among task characteristics, window attributes, and satellite states. This results in suboptimal resource allocation and limited solution quality, particularly for large-scale instances.

To address the aforementioned issues, this paper proposes a dual-layer clustering guided genetic algorithm (DLCGA). The problem is formulated as a multi-objective optimization model to maximize the total task profit and average satellite profit per unit time. Since tasks possess diverse attributes, similar tasks may have similar assignment patterns. Thus, clustering techniques are adopted to group similar tasks and time windows for guiding the evolutionary process. The scheduling process is decoupled into a task sequence construction layer and a time-window allocation layer. The two optimization layers dynamically switch during the iterative process and adaptively adjust the cluster size according to optimization performance for enhancing exploration.

II. Problem Model

Given a satellite observation system [10], satellite resource states, including memory and energy among others, are reset at the beginning of each orbital period. Each observation task can be completed within a single visible time window, ensuring complete coverage of the target area during one satellite pass. Each satellite executes at most one task at any given time.

Let $\mathcal{S} = \{S_1, \ldots, S_M\}$ and $\mathcal{T} = \{T_1, \ldots, T_N\}$ denote the sets of satellites and tasks, respectively. For satellite S_i, let τ_i denote its type (optical, SAR, or EM), ρ_i its spatial resolution, ϕ_i its polarization mode, E_i its energy capacity per orbital period, and C_i its storage capacity. Let $\mathcal{W}_{ij} = \{w_{ij}^1, \ldots, w_{ij}^{K_{ij}}\}$ represent the set of visible time windows for satellite S_i to observe task T_j, where $K_{ij} = |\mathcal{W}_{ij}|$ is the number of available windows. For task T_j, let κ_j^r, r_j^r and ϕ_j^r denote the required satellite type, resolution, and polarization mode, respectively. Let e_j, c_j, d_j, and p_j represent the energy consumption, storage requirement, execution duration, and profit of task T_j, respectively. The transition time between tasks T_{j_1} and T_{j_2} on the same satellite is denoted as $\Delta_{j_1 j_2}$. Binary decision variable $x_{ijk} \in \{0,1\}$ indicates whether satellite S_i executes task T_j during time window w_{ij}^k, i.e., $w_{ij}^k \in \mathcal{W}_{ij}$. Let t_{ijk}^s and t_{ijk}^e denote its start time and end time, respectively.

1) Constraints: :

$$x_{ijk} \leq \mathbb{I}(\tau_i = \kappa_j^r), \quad \forall S_i \in \mathcal{S}, \forall T_j \in \mathcal{T}, \forall k \in [K_{ij}] \tag{1}$$

$$x_{ijk} \leq \mathbb{I}(\rho_i \leq \rho_j^r), \quad \forall S_i \in \mathcal{S}, \forall T_j \in \mathcal{T}, \forall k \in [K_{ij}] \tag{2}$$

$$x_{ijk} \leq \mathbb{I}(\phi_i = \phi_j^r), \quad \forall S_i \in \mathcal{S}, \forall T_j \in \mathcal{T}, \forall k \in [K_{ij}] \tag{3}$$

$$\sum_{T_j \in \mathcal{T}} \sum_{k=1}^{K_{ij}} e_j x_{ijk} \leq E_i, \quad \forall S_i \in \mathcal{S} \tag{4}$$

$$\sum_{T_j \in \mathcal{T}} \sum_{k=1}^{K_{ij}} c_j x_{ijk} \leq C_i, \quad \forall S_i \in \mathcal{S} \tag{5}$$

$$\sum_{S_i \in \mathcal{S}} \sum_{k=1}^{K_{ij}} x_{ijk} \leq 1, \quad \forall T_j \in \mathcal{T} \tag{6}$$

$$t_{ij_2k_2}^s - t_{ij_1k_1}^e \geq \Delta_{j_1 j_2}, \quad j_1 < j_2, (j_\ell, k_\ell) \in \Pi_i \tag{7}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition is true and 0 otherwise, and Π_i denotes the sequence of task-window pairs assigned to satellite S_i (i.e., tasks with their visible time windows on the satellite), with $j_1 < j_2$ indicating that task T_{j_1} precedes T_{j_2} in Π_i. Eqs. (1), (2) and (3) ensure that the observing satellite matches task requirements regarding type, resolution, and polarization mode. Eqs. (4) and (5) limit energy consumption and storage capacity per orbital period for each satellite. Eq. (6) ensures each task is executed at most once. Eq. (7) guarantees sufficient transition time between consecutive tasks on the same satellite.

2) Objective Functions: : This paper aims to optimize two objectives, i.e., the total profit f_1 and the average ratio of task execution profit to execution time across satellites f_2:

$$\max F(S) = (f_1(S), f_2(S))$$

$$f_1(S) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{K_{ij}} p_j x_{ijk} \tag{8}$$

$$f_2(S) = \frac{1}{M} \sum_{i=1}^{M} \left(\frac{\sum_{j=1}^{N} \sum_{k=1}^{K_{ij}} p_j x_{ijk}}{\sum_{j=1}^{N} \sum_{k=1}^{K_{ij}} d_j x_{ijk}} \right) \tag{9}$$

where f_1 directly reflects the overall economic benefit of task execution, and f_2 promotes satellites to accommodate more tasks (or achieve higher total profit) within limited observation windows by encouraging the selection of high-value tasks with shorter execution duration.

III. Proposed Algorithm

DLCGA adopts a dual-layer evolution scheme with task and time-window layers. Each individual contains a task sequence and corresponding time-window assignments. An external archive $\mathcal{F}$ stores nondominated solutions. The algorithm alternates between layers based on offspring quality counters (count_1, count_2) reaching thresholds (θ_1, θ_2), as shown in Fig. 1.

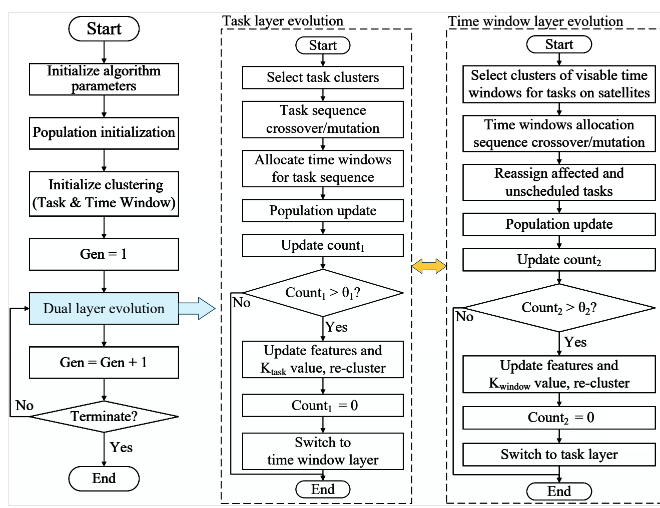


Fig. 1: The flowchart of DLCGA.

A. Clustering Strategy

The address the large-scale tasks, K-means clustering is employed at both tasks and time windows to divide the elements into multiple clusters, guiding the algorithm toward promising areas. Z-score normalization is performed to avoid the influence of the scales of different features. Denote the number of clusters for tasks and time windows as K_{task} and K_{window}, respectively. K-means clustering is performed on the feature vectors of tasks or time windows to obtain $K(= K_{\text{task}}$ or $K_{\text{window}})$ clusters. Each cluster is represented by the mean feature vectors over all tasks (or time windows) inside. The feature vectors of all clusters are ranked using nondominated sorting. The selection probability for cluster C_k with nondominated rank r_k is calculated as:

$$p_k = \frac{\frac{1}{r_k+1}}{\sum_{l=1}^{K} \frac{1}{r_l+1}} \tag{10}$$

Tasks with similar characteristics are clustered to explore task combinations that share compatible resource requirements and profitability. For each task T_j, we construct a feature vector combining static and dynamic attributes:

$$\mathbf{v}_j^{\text{task}} = [p_j, d_j, p_j/d_j, c_j, e_j, \alpha_j] \tag{11}$$

where p_j/d_j indicates profit per unit time of task T_j, c_j and e_j are defined in Section II. α_j represents the task selection probability defined as:

$$\alpha_j = \frac{\sum_{s\in\mathcal{F}} \mathbb{I}(T_j \in s)}{\sum_{s\in\mathcal{F}} |s|} \tag{12}$$

where $\mathcal{F}$ denotes the current nondominated solution set, $\mathbb{I}(T_j \in s)$ is an indicator function that returns 1 if task T_j appears in solution s and 0 otherwise, and $|s|$ denotes the number of tasks scheduled in solution s. The frequency α_j is initialized to 0 and updated iteratively as the algorithm progresses. Tasks with higher α_j values have historically contributed more frequently to high-quality solutions and are emphasized in subsequent genetic operations.

Time windows are grouped based on their temporal properties and historical selection patterns. Each triplet (task T_j, satellite S_i, time window $w_{ij}^k \in \mathcal{W}_{ij}$) is taken as a distinct sample with a feature vector:

$$\mathbf{v}_{ijk}^{\text{window}} = [t_{ijk}^s, t_{ijk}^e, t_{ijk}^e - t_{ijk}^s, \alpha_j, \beta_{ijk}] \tag{13}$$

where $t_{ijk}^e - t_{ijk}^s$ is the window duration, α_j is the task selection probability as defined in Eq. (12), and β_{ijk} represents the window selection probability defined as:

$$\beta_{ijk} = \frac{\sum_{s\in\mathcal{F}} \mathbb{I}((T_j, S_i, w_{ij}^k) \in s)}{\sum_{s\in\mathcal{F}} \mathbb{I}((T_j, S_i) \in s)} \tag{14}$$

where $\mathbb{I}((T_j, S_i, w_{ij}^k) \in s)$ indicates whether solution s assigns task T_j to satellite S_i using time window w_{ij}^k, and $\mathbb{I}((T_j, S_i) \in s)$ indicates whether solution s assigns task T_j to satellite S_i regardless of the specific time window chosen. The frequency β_{ijk} is initialized to 0 and updated as the algorithm iterates. Time windows with higher β_{ijk} values are prioritized.

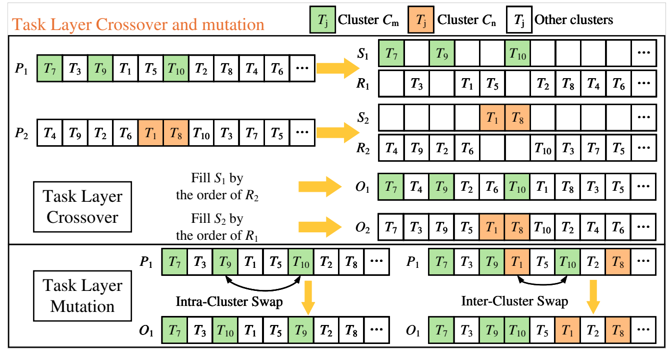


Fig. 2: Illustration of crossover and mutation operations at the task layer.

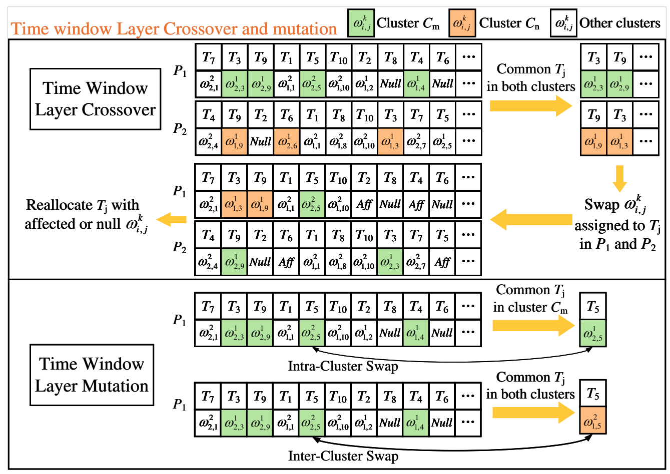


Fig. 3: Illustration of crossover and mutation operations at the time-window layer.

B. Cluster Guided Genetic Operations

The clustering strategy enables genetic operations to exploit structural patterns in the search space. Genetic operations adjust the task sequences and the time-window assignments for tasks.

1) Crossover: Crossover operations exchange gene segments between parents based on cluster membership. As illustrated in Fig. 2, task layer crossover selects two clusters C_m and C_n based on Eq. (10), extracts corresponding task segments from parents P_1 and P_2, and generates offspring by preserving segment positions while filling remaining slots with tasks from the other parent. Time-window layer crossover (Fig. 3) exchanges satellite and window assignments for common tasks between selected clusters, then reconstructs affected satellite schedules.

2) Mutation: Mutation operations modify solutions through strategic position swaps within or across clusters. At the task layer, intra-cluster mutation selects one cluster according to Eq. (10) and randomly swaps the positions of two tasks within that cluster. Inter-cluster mutation selects two clusters and swaps the positions of one randomly selected task from each cluster. At the time-window layer, intra-cluster mutation selects a task within the cluster that has alternative feasible satellite observation options and reassigns its allocation. Inter-cluster mutation selects two clusters according to Eq. (10), identifies the same task that appears in both clusters with different feasible satellite observation options, and reassigns its allocation. Following mutation, affected satellites reconstruct their schedules to maintain feasibility.

3) Time-Window Selection Strategy: For each task T_j, all feasible satellite orbit and time window pairs (S_i, w_{ij}^k) are chosen from the candidate set to satisfy temporal and resource constraints. If there is no feasible options for task T_j, it is discarded and the selection proceeds to the next task; otherwise, each feasible pair (S_i, w_{ij}^k) is evaluated using three criteria, i.e., the window selection frequency β_{ijk} derived from nondominated solutions as Eq. (14), the storage abundance ratio and the energy abundance ratio, which are calculated as:

$$R_{ijk}^{\text{storage}} = \frac{C_i^{\text{r}}(w_{ij}^k)}{c_j}, \; R_{ijk}^{\text{energy}} = \frac{E_i^{\text{r}}(w_{ij}^k)}{e_j} \tag{15}$$

where $C_i^{\text{r}}(w_{ij}^k)$ denotes the residual storage capacity of satellite S_i in the orbital period containing time window w_{ij}^k, $E_i^{\text{r}}(w_{ij}^k)$ denotes the residual energy of satellite S_i in the orbital period containing time window w_{ij}^k. The first criterion is maximized to favor historically preferred windows, while the second and third criteria are minimized to promote efficient resource utilization. All feasible options are ranked using nondominated sorting based on the three criteria. The selection probability for option (S_i, w_{ij}^k) with nondominated rank r_{ijk} is computed as:

$$p_{ijk} = \frac{\frac{1}{r_{ijk}+1}}{\sum\limits_{(S_{i'}, w_{i'j}^{k'}) \in \mathcal{SW}_j} \frac{1}{r_{i'jk'}+1}} \tag{16}$$

where $\mathcal{SW}_j$ denotes the set of all feasible (S_i, w_{ij}^k) pairs for task T_j. A satellite and one of its visible time windows for the task are selected based on the probabilities calculated by Eq. (16). This strategy enables the algorithm to construct feasible solutions while learning from historical search patterns and adapting to dynamic resource availability.

C. Population Initialization and Update

1) Population Initialization: The initial population comprises five strategies, each generating 20% of solutions: the window scarcity rule prioritizes tasks with fewer feasible time windows across satellites; the profit priority rule prioritizes tasks with higher profit values; the duration priority rule prioritizes tasks with shorter execution durations; the profit per time rule prioritizes tasks with higher efficiency ratios p_j/d_j; and the random strategy selects tasks uniformly to enhance diversity. To enhance population diversity, for each individual generated by the heuristic rules, random swapping of two tasks is performed on 5% tasks. After that, satellite and time windows are assigned through a greedy procedure based on the task sequence. For each task, time windows are examined chronologically starting from the earliest available window, and the first window satisfying constraints is selected. Then the algorithm begins at the task layer evolution.

TABLE I: Algorithm Parameters

Algorithm	Parameters
DLCGA	$N_p = 40$, $\theta_1 = 10$, $\theta_2 = 15$, $p_c^{\text{task}} = p_c^{\text{window}} = 0.8$, $p_m^{\text{task}} = 0.05$, $p_m^{\text{window}} = 0.25$, $K_{\text{task}}^{\text{init}} = K_{\text{window}}^{\text{init}} = 2N/100$
BEA	$N_p = 30$, crossover probability 1.0, mutation probability $1/N$
CGA	$N_p = 30$, crossover probability 0.9, mutation probability 0.05, initial $k = 8$ (500 tasks), $k = 12$ (700 tasks), $k = 16$ (900 tasks), $k = 20$ (1100 tasks), $k = 24$ (1300 tasks), $k_{\min} = k/2$
RL-MA	$N_p = 30$, $L = 2$, $L_p = 20$, $\varepsilon_s = 1 - t/10000$, crossover probability $\alpha = 0.9$, mutation probability $\beta = 0.1$, $\alpha_L = 0.01$, $\gamma = 0.95$
SLCGA	$N_p = 30$, $\theta_1 = 30$, $p_c^{\text{task}} = 0.8$, $p_m^{\text{task}} = 0.05$, $K_{\text{task}}^{\text{init}} = K_{\text{window}}^{\text{init}} = 2N/100$

SLCGA is a DLCGA variant with the task layer only

2) Population Update: At each generation, the parent population P and offsprings Q are combined to form a temporary population $R = P \cup Q$ for nondominated sorting based on the two optimization objectives. The crowding distance mechanism [11] is adopted to select the next population P' for maintaining solution diversity.

3) Switch between Layers: At the task layer, if count_1 reaches a predefined threshold θ_1, then the task cluster number K_{task} decreases until reaching lower bound $\lfloor K_{\text{task}}^{\text{init}}/2 \rfloor$, where $K_{\text{task}}^{\text{init}}$ is the initial value. Tasks are reclustered. The algorithm then switches to time-window layer operations for several generations, modifying only satellite and time window assignments. In time-window layer, if count_2 reaches a predefined threshold θ_2, the time-window cluster number K_{window} decreases until reaching $\lfloor K_{\text{window}}^{\text{init}}/2 \rfloor$, where $K_{\text{window}}^{\text{init}}$ is the initial value. Dynamic features of satellite and time window pairs are recalculated for clustering. The algorithm then returns to task layer operations. This adaptive mechanism refines clustering structure dynamically in response to search progress.

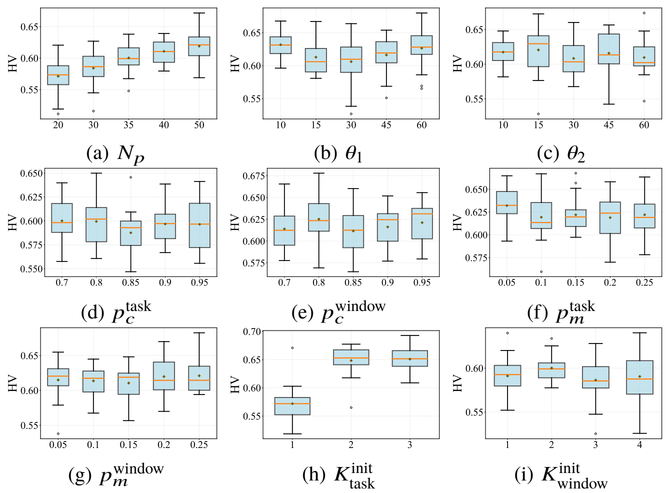
D. Worst Computational Complexity of DLCGA

Let N denote the number of tasks, M the number of satellites, W the maximum number of feasible time windows for task-satellite pairs, N_p the population size, K_{task} the number of task clusters, K_{window} the number of time window clusters, and I the total number of iterations. The worst-case complexity over I iterations is $O(I \times (N_p \times N \times M \times W + N \times M^2 \times W^2 + N_p^2))$.

IV. Experiments

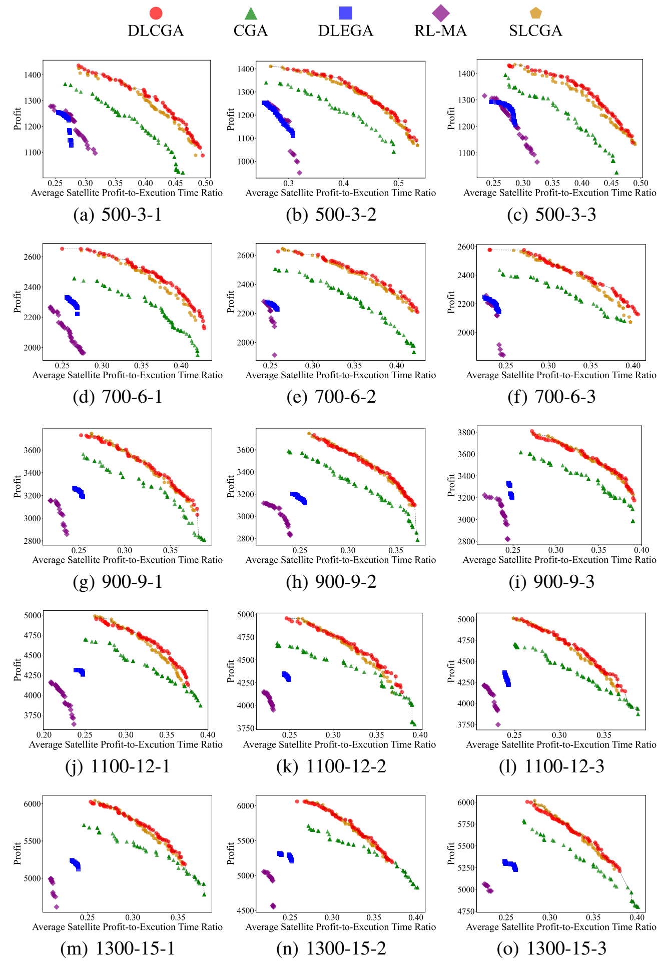
A. Experimental Settings

1) Test Instances: Fifteen test instances with varying numbers of tasks and satellites are generated using Systems Tool Kit (STK) 11.6. Each instance contains three task types corresponding to three satellite types. The number of tasks ranges from 500 to 1300 with increments of 200. Three instances are created for each task scale. Each instance follows the naming convention "$A-B-C$", where A denotes the total number of tasks, B represents the number of satellites, and C indicates the instance serial number under the same task scale. In each



(a) N_p (b) θ_1 (c) θ_2 (d) p_c^{task} (e) p_c^{window} (f) p_m^{task} (g) p_m^{window} (h) $K_{\text{task}}^{\text{init}}$ (i) $K_{\text{window}}^{\text{init}}$

Fig. 4: Box plots of HV values obtained by DLCGA with different parameters.



(a) 500-3-1 (b) 500-3-2 (c) 500-3-3 (d) 700-6-1 (e) 700-6-2 (f) 700-6-3 (g) 900-9-1 (h) 900-9-2 (i) 900-9-3 (j) 1100-12-1 (k) 1100-12-2 (l) 1100-12-3 (m) 1300-15-1 (n) 1300-15-2 (o) 1300-15-3

Fig. 5: Nondominated solutions obtained by each algorithm in all runs on each instance with (a)-(c) 500 tasks, (d)-(f) 700 tasks, (g)-(i) 900 tasks, (j)-(l) 1100 tasks, and (m)-(o) 1300 tasks.

instance, task locations are randomly distributed across global latitudes and longitudes, and the visible time windows are computed for each satellite observing each task.

2) Competing Algorithms: To validate the effectiveness of the proposed DLCGA algorithm, three recent and representative evolutionary algorithms commonly used for satellite scheduling problems are adopted for comparison, namely the bilevel evolutionary algorithm (BEA) [6], the cluster based genetic algorithm (CGA) [12], and the reinforcement learning based memetic algorithm (RL-MA) [13]. Additionally, a variant of DLCGA called single layer clustering guided genetic algorithm (SLCGA) is compared.

3) Performance Measure: Hypervolume (HV) [14] is employed to evaluate algorithm performance by measuring the volume of objective space dominated by the obtained solution set relative to a reference point. The reference point is set to (1, 1). Each nondominated solution is normalized by the extreme objective values across all nondominated solutions. To ensure fair comparison, the maximum function evaluation times is set as 10,000. The Wilcoxon rank sum test is applied with a significance level of 0.05 to determine statistical differences between DLCGA and each competing algorithm. The notations "+", "=", and "-" denote that DLCGA performs significantly better than, statistically equivalent to, or worse than the competing algorithm, respectively.

B. Parameter Investigation

The parameters of DLCGA are investigated using instance 500-3-1, including population size N_p, layer transition thresholds θ_1 and θ_2, crossover probabilities p_c^{task} and p_c^{window}, mutation probabilities p_m^{task} and p_m^{window}, and initial cluster numbers $K_{\text{task}}^{\text{init}}$ and $K_{\text{window}}^{\text{init}}$ (computed as $c \times (N/100)$ where N denotes the task number and the scaling coefficient c is investigated). Fig. 4 shows the HV values under different parameter settings.

C. Compared with Other Algorithms

1) Results: Table II presents the HV values obtained by all algorithms across all test instances. The results demonstrate that DLCGA consistently outperforms all competing algorithms, achieving the highest HV values on 14 instances. This indicates that the solution sets obtained by DLCGA exhibit superior performance in terms of diversity across the objective space. The variant of DLCGA, namely SLCGA, achieves the second best ranking, and obtains results close to or equal to those of DLCGA on five medium and large scale instances. Additionally, the CGA algorithm ranks third in the overall performance evaluation. The comparison between CGA and DLCGA indirectly reflects the significant influence of the time window allocation strategy on the final scheduling results.

2) Solution Quality: To intuitively visualize the solution quality, Fig. 5 presents the nondominated solutions acquired by each algorithm across all runs on all instances within the objective space. For small-scale instances with 500 tasks, the solution distributions of all algorithms exhibit relatively

TABLE II: HV Results of DLCGA, CGA, DLEGA, RL-MA, and SLCGA

Instance	CGA		DLEGA		RL-MA		SLCGA		DLCGA
	Mean(Std)		Mean(Std)		Mean(Std)		Mean(Std)		Mean(Std)
500-3-1	4.08e-01(2.72e-02)	+	8.90e-02(1.58e-02)	+	1.03e-01(2.44e-02)	+	6.54e-01(1.84e-02)	+	**7.09e-01(2.28e-02)**
500-3-2	4.46e-01(2.86e-02)	+	1.06e-01(1.51e-02)	+	9.96e-02(2.54e-02)	+	7.11e-01(2.60e-02)	+	**7.38e-01(2.19e-02)**
500-3-3	4.20e-01(2.43e-02)	+	1.18e-01(2.19e-02)	+	1.06e-01(3.70e-02)	+	7.10e-01(1.91e-02)	+	**7.47e-01(2.35e-02)**
700-6-1	5.24e-01(1.65e-02)	+	1.38e-01(1.45e-02)	+	7.59e-02(2.09e-02)	+	7.45e-01(2.49e-02)	+	**7.90e-01(2.34e-02)**
700-6-2	5.13e-01(2.00e-02)	+	7.24e-02(1.35e-02)	+	6.04e-02(1.76e-02)	+	7.51e-01(1.88e-02)	+	**7.76e-01(2.39e-02)**
700-6-3	4.80e-01(2.55e-02)	+	9.59e-02(1.20e-02)	+	6.90e-02(2.32e-02)	+	7.23e-01(2.69e-02)	+	**7.56e-01(2.68e-02)**
900-9-1	5.50e-01(1.83e-02)	+	1.32e-01(1.73e-02)	+	5.58e-02(1.18e-02)	+	7.25e-01(2.33e-02)	=	**7.35e-01(1.79e-02)**
900-9-2	5.28e-01(1.87e-02)	+	1.09e-01(1.49e-02)	+	4.10e-02(9.81e-03)	+	7.40e-01(1.79e-02)	=	**7.43e-01(2.80e-02)**
900-9-3	6.15e-01(1.19e-02)	+	1.17e-01(1.42e-02)	+	6.65e-02(1.28e-02)	+	7.97e-01(2.44e-02)	=	**8.00e-01(1.85e-02)**
1100-12-1	5.11e-01(1.21e-01)	+	9.56e-02(1.54e-02)	+	3.61e-02(6.31e-03)	+	7.04e-01(2.09e-02)	=	**7.15e-01(1.93e-02)**
1100-12-2	5.15e-01(2.22e-02)	+	9.53e-02(1.31e-02)	+	2.96e-02(8.57e-03)	+	6.80e-01(1.60e-02)	+	**6.97e-01(1.79e-02)**
1100-12-3	5.01e-01(1.34e-02)	+	9.11e-02(1.29e-02)	+	3.45e-02(8.09e-03)	+	6.64e-01(1.73e-02)	+	**6.81e-01(1.89e-02)**
1300-15-1	5.28e-01(1.70e-02)	+	9.23e-02(1.08e-02)	+	1.96e-02(4.36e-03)	+	6.89e-01(1.28e-02)	+	**7.05e-01(1.46e-02)**
1300-15-2	5.10e-01(1.87e-02)	+	8.51e-02(1.29e-02)	+	2.35e-02(4.89e-03)	+	6.53e-01(1.31e-02)	+	**6.63e-01(1.48e-02)**
1300-15-3	5.42e-01(1.70e-02)	+	9.28e-02(1.62e-02)	+	1.97e-02(7.93e-03)	+	**6.83e-01(1.66e-02)**	=	6.79e-01(2.47e-02)
Summary (+/=/-)	15/0/0		15/0/0		15/0/0		10/5/0		

concentrated. Specifically, DLCGA and SLCGA tend to generate solution sets with higher profits, whereas CGA shows competitive performance in achieving better average satellite profit to execution time ratio. As the instance scale grows, the disparities in solution distribution among different algorithms become increasingly evident. Notably, DLEGA and RL-MA demonstrate significant degradation in the average satellite profit to execution time ratio objective. Meanwhile, CGA produces solution sets with limited performance in profit but exhibits advantages in the second objective. As the task scale expands, the solution distributions of DLEGA and RL-MA become increasingly dense with diminishing solution quantities, indicating insufficient exploration capability and diversity in the objective space. In contrast, the solution sets obtained by the proposed DLCGA and its variant SLCGA continue to demonstrate excellent coverage and distribution characteristics. DLCGA exhibits superior performance compared to SLCGA in terms of the average satellite profit to execution time ratio, whereas the solution sets of SLCGA tend to favor higher profit. Thus, DLCGA consistently achieves solution sets with uniform distribution in the objective space. This demonstrates that the dual-layer clustering mechanism effectively enhances both exploration and exploitation capabilities, enabling DLCGA to maintain robust performance in discovering diverse and high-quality solutions.

V. Conclusion

This study investigates the multiobjective multitype satellite observation scheduling problem and develops DLCGA to generate diversified and high-quality solution sets. DLCGA adopts a bilevel optimization framework that decouples the scheduling process into task sequence construction and time-window allocation. Clustering-based genetic operators are adopted to guide the search towards high-quality solutions based on features of tasks and time windows. Experimental results on 15 instances with varying scales demonstrate that DLCGA significantly outperforms competing algorithms in terms of convergence and diversity. The dual-layer clustering mechanism effectively enhances exploration and exploitation capabilities, enabling DLCGA to obtain high-quality solution sets.

References

[1] S. L. Ustin and E. M. Middleton, “Current and near-term earth-observing environmental satellites, their missions, characteristics, instruments, and applications,” *Sensors*, vol. 24, no. 11, p. 3488, 2024.

[2] G. Gao, L. Yao, W. Li, L. Zhang, and M. Zhang, “Onboard information fusion for multisatellite collaborative observation: Summary, challenges, and perspectives,” *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 2, pp. 40–59, 2023.

[3] X. Zhai, X. Niu, H. Tang, L. Wu, and Y. Shen, “Robust satellite scheduling approach for dynamic emergency tasks,” *Math. Probl. Eng.*, vol. 2015, no. 1, p. 482923, 2015.

[4] X.-F. Liu, Y. Fang, Z.-H. Zhan, Y.-L. Jiang, and J. Zhang, “A cooperative evolutionary computation algorithm for dynamic multiobjective multi-auv path planning,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 669–680, 2024.

[5] X.-F. Liu, X.-X. Xu, Z.-H. Zhan, Y. Fang, and J. Zhang, “Interaction-based prediction for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1881–1895, 2023.

[6] F. Yao, Y. Chen, L. Wang, Z. Chang, P.-Q. Huang, and Y. Wang, “A bilevel evolutionary algorithm for large-scale multiobjective task scheduling in multiagile earth observation satellite systems,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 6, pp. 3512–3524, 2024.

[7] X.-F. Liu, Y. Fang, Z.-H. Zhan, and J. Zhang, “Strength learning particle swarm optimization for multiobjective multirobot task scheduling,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 7, pp. 4052–4063, 2023.

[8] X.-F. Liu, M. Gao, Y. Fang, Z.-H. Zhan, and J. Zhang, “Contrastive-learning-based decision making for dynamic time-linkage optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 11, pp. 8661–8674, 2025.

[9] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, and G. Wu, “A generic markov decision process model and reinforcement learning method for scheduling agile earth observation satellites,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 3, pp. 1463–1474, 2020.

[10] J. Berger, N. Lo, and M. Barkaoui, “Quest–a new quadratic decision model for the multi-satellite scheduling problem,” *Comput. Oper. Res.*, vol. 115, p. 104822, 2020.

[11] C. R. Raquel and P. C. Naval Jr, “An effective use of crowding distance in multiobjective particle swarm optimization,” in *Proc. Genet. Evol. Comput. Conf.*, 2005, pp. 257–264.

[12] Y. Song, J. Ou, J. Wu, Y. Wu, L. Xing, and Y. Chen, “A cluster-based genetic optimization method for satellite range scheduling system,” *Swarm Evol. Comput.*, vol. 79, p. 101316, 2023.

[13] Y. Song, P. N. Suganthan, W. Pedrycz, R. Yan, D. Fan, and Y. Zhang, “Energy-efficient satellite range scheduling using a reinforcement learning-based memetic algorithm,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 4, pp. 4073–4087, 2024.

[14] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, “Hyperplane assisted evolutionary algorithm for many-objective optimization problems,” *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3367–3380, 2019.