Hierarchical Resource Scheduling Method for Satellites Combining Particle Swarm Optimization and Heuristic Factors

Yang Li
College of the Intelligence Science and Technology
National University of Defense Technology
Changsha, China
liyang_17@nudt.edu.cn

Xiye Guo*
College of the Intelligence Science and Technology
National University of Defense Technology
Changsha, China
13574832047@163.com

Xiaotian Ma
College of the Intelligence Science and Technology
National University of Defense Technology
Changsha, China
mxt2020@nudt.edu.cn

Sichuang Ren
College of the Intelligence Science and Technology
National University of Defense Technology
Changsha, China
rensichuang12@nudt.edu.cn

Jun Yang
College of the Intelligence Science and Technology
National University of Defense Technology
Changsha, China
john323@163.com

2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI) | 979-8-3503-2714-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEMI59194.2023.10270089

***Abstract*—Satellites are crucial in meteorological remote sensing, national defence, and economic development. However, resources on satellites are limited, and satellite mission execution is also subject to various time and space constraints. Proper scheduling of satellite resources is essential to complete as many users' tasks as possible. Nevertheless, it is challenging and has been proven to be an NP-hard problem. To address this problem, we propose a hierarchical scheduling framework that combines an improved particle swarm algorithm with heuristic factors. The resource scheduling problem is divided into task assignment and resource scheduling in the timeline. This approach effectively reduces the complexity of the resource scheduling problem and promotes search performance. It also achieves the goals of load balance, ensuring stable satellite system operation and meeting users' requirements.**

Keywords—Particle swarm optimization, satellite resource scheduling, heuristic factor, load balance

I. INTRODUCTION

Earth observation satellites have been widely used in agriculture, meteorological monitoring, disaster relief and other fields. In national defence and economic development, they play a significant role in the acquisition of information on the earth's surface. Although the number of satellites has increased recently, the observation demand has also grown greatly. The resource scheduling problem of satellites is still an NP-hard problem [1]. Therefore an efficient resource scheduling algorithm is significant for making full use of satellite resources and completing various tasks efficiently.

Earth observation satellites are in resource-limited environments. They operate under strict time and space restrictions. The core of satellite operation control is ensuring the satellite meets various resource and space-time constraints, operates smoothly, and meets user needs by allocating corresponding resources to relevant tasks. Therefore, resource optimization configuration and scheduling are fundamental but challenging issues. Numerous researchers have considered the earth observation satellite resource scheduling problem as a single-objective optimization problem with total mission profit. However, multi-objective optimization is more appropriate to the actual satellite operation environment. It is necessary to meet the needs of the satellite itself and consider the system's global balance [2]. Wei L. [3] studied the multi-orbit scenario of the agile earth observation satellite problem with two objectives, which considered the task profit and load balance of tasks completed in different orbits. From the perspective of resource allocation, maintaining the working balance can retional the utilization of satellites and prolong their service life. In satellite edge computing, computing resources are limited and should ensure that tasks are not overly concentrated on a node to ensure system stability [4]. Load balancing of network traffic is performed in network communication to avoid over-congestion of nodes causing network paralysis [5]. Load balance should also be considered in the process of resource scheduling for multiple Earth observation satellites because balanced resources can reserve sufficient resources for possible unexpected tasks to respond to users' needs. In this way, the resource scheduling plan can be minimally changed. In this paper, the satellite's total task profit to satisfy users' observation demand is the first objective, and the load balance of resources between satellites is considered the second optimization objective.

Efficient scheduling algorithms are necessary to solve such problems. The work of the scholars will be described in terms of model building and algorithms. Standard scheduling models include the graph theory model [4], constraint satisfaction problems [7], linear integer programming model [8], etc. While the graph theory model has a simple form, it can be challenging to represent problems and constraints in the graph. Additionally, the constraint satisfaction problem model has strict restrictions and requires long search times. The linear integer programming model regards constraints as linear, ignoring nonlinear constraints. As a result, there is still a gap between model establishment and actual problems.

The algorithm for solving problem models is the key to solving satellite scheduling problems. Typically, it includes genetic algorithms [9], ant colony optimization algorithms[10] and their combination methods, etc. Liu W. et al. [11] proposed an improved whale optimization algorithm

979-8-3503-2714-4/23/$31.00 ©2023 IEEE

for communication satellite resource scheduling, which avoids falling into the local optimal solution through cross-mutation operators and achieves high scheduling quality and stability. Chen M. et al. [9] proposed a genetic algorithm based on population disturbance and elimination strategy to solve the scheduling problem of multi-satellite measurement and control resources, which performs well in total profits and task completion rate. Zhang Z. et al. [10] conducted research on satellite control problems and proposed a simple ant colony algorithm that introduced an optimization model based on visible arcs and work cycles. The effectiveness of the algorithm was demonstrated through simulation and comparative experiments. However, the satellite resource scheduling problem has numerous nonlinear constraints, diverse optimization objectives, and high problem complexity. Therefore, it is hard to find an algorithm suitable for all scenarios. Extensive studies have demonstrated that intelligent optimization algorithms exhibit broad applications and good comprehensive performance.

The particle swarm optimization (PSO), proposed by James Kennedy and Russell Eberhart [12] in 1995, is an intelligent optimization algorithm inspired by birds foraging. Its parameters are simple but powerful. It starts from the random solution to find the optimal solution iteratively and evaluates the quality of the solution through the fitness function. There is no need for PSO to design complex rules. For its adaptability, PSO has been widely applied in many optimization problems [13]-[14]. Based on existing research results and the actual needs of scheduling issues, this paper adopts an improved particle swarm optimization (IPSO) algorithm combining heuristic factors to solve the multi-satellite resource scheduling problem.

A method that combines PSO and heuristic factors for satellite resource scheduling is proposed. The problem is decomposed into task allocation and resource scheduling in the timeline. Compared with the non-layered approach, the problem representation is simplified, the constraint checking is reduced, and the computational complexity is declined.

This paper is organized as follows. In section II, the model for satellite resource scheduling is designed, and the optimization objectives and constraints of the model under this scheme are analyzed. In section III, this work put forward a method combining PSO and heuristic factors. In Section IV, the proposed scheme and optimization algorithm is simulated. Finally, the conclusion and future work are elaborated.

II. Model Construction

A. Problem description

The satellite resource scheduling problem is a multi-objective optimization problem with multiple constraints. The user sends a request to the scheduling center. Tasks are described in terms of priority, resources required, task duration, etc. The task profit is evaluated by the priority of the task and the resources consumed by the task. The scheduling center allocates resources and satellites for tasks based on the satellites' resource situation and provides a reasonable resource scheduling plan.

The resource scheduling system mainly includes satellites, users, and the scheduling center, and its system composition is shown in Fig. 1.

Firstly, the resource scheduling problem is modeled, and the model is elaborated from the perspectives of problem variables, constraint conditions, and optimization objectives.

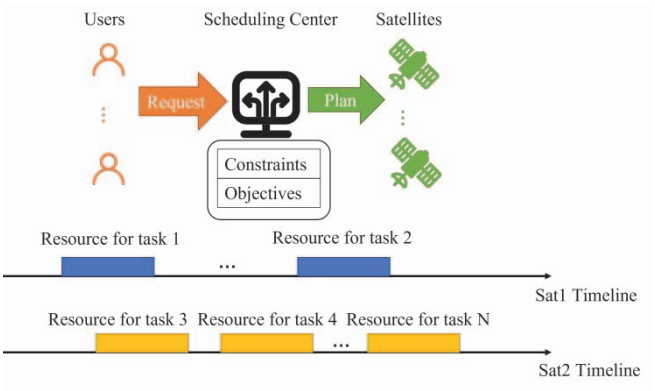

Fig. 1. Components of the resource scheduling center.

The problem variables include the satellite set S_{set}={SAT_1, SAT_2, SAT_3,...}, where SAT_i={MEM_i, Ene_i} represents the current resource state of the satellite i. Task set T_{set}={$Task_1$, $Task_2$, $Task_3$,...}, where $Task_j$={Pro_j, m_j, e_j, [(VW_{s1}, VW_{e1}, SAT_1), (VW_{s2}, VW_{e2}, SAT_2),...]}. The symbol representations involved are shown in TABLE I.

TABLE I. The Definition of the Symbol

Symbol	Definition
Nt	The total number of tasks
Ns	The total number of satellites
S_{set}	The set of satellites to schedule
T_{set}	The set of tasks to execute
e_i	The energy consumed by the ith task
m_i	The memory consumed by the ith task
Pro_i	The profit of the ith task
x_i	A variable indicating whether a task has been completed or not.
MEM_i	The total memory space of the ith satellite
Ene_i	The total energy of the ith satellite
VW_{s1}	The start time of the executable time window for task i
VW_{e1}	The end time of the executable time window for task i
EW_{s1}	The start time for task i to execute
EW_{e1}	The end time for task i to execute
t_{trans}	The switching time between two adjacent tasks

B. Constraints

Satellite resource scheduling faces numerous time and resource constraints, with the following constraints:

(1)Task execution time constraint: Due to the relative rotation between the earth and satellites, the tasks must be executed within the time window between the target and the satellite. The start time of execution must precede the beginning of the assigned time window, and the end time of execution must be prior to the end of the same time window, which meets the constraint conditions.

$$\begin{cases} EWs_i \geq VWs_i \\ EWe_i \leq VWe_i \end{cases} \tag{1}$$

(2)Task switching constraint: After executing a task on the same satellite, it needs some time to switch to the next task. Meet constraint conditions:

$$EWe_i + t_{trans} \le EWs_{i+1} \tag{2}$$

(3) Task storage and energy resource constraints: The storage capacity of each satellite is restricted, which can limit the number of completed tasks. Satellite task execution requires the consumption of electrical energy. Therefore, the scheduling process must consider the constraints imposed by satellite energy. The storage and energy resources consumed to complete a task must not exceed the satellite's total storage and energy resources.

$$\sum_{j=1}^{N_i} e_j \le Ene_i \tag{3}$$

$$\sum_{j=1}^{N_i} m_j \le MEM_i \tag{4}$$

C. Objectives

This work considers two optimization objectives: task completion rate and satellite resource load balance. During the task execution process, the first step is to ensure that the user needs are met as much as possible and to improve the total rewards obtained from task completion, which is reflected in (5). In satellite edge computing, resource load balance [15] is an essential factor. It is calculated as (7). Load balance can improve the system's efficiency and reliability. It can prevent failures caused by excessive scheduling of single satellite tasks and heavy loads. Therefore, the system's robustness is enhanced. A constellation with balanced resource status can effectively respond to potential mission requests in the future. In this situation, there is no need to disrupt existing resource scheduling plans on a large scale. The objective function formulas are as follows:

$$f_1 = Max \sum_{j=1}^{N_{task}} x_j pro_j \tag{5}$$

$$x_j = \begin{cases} 0, \ task\ is\ scheduled \\ 1, else \end{cases} \tag{6}$$

$$f_2 = Min \sum_{i=1}^{N_{sat}} (R_i - R_{ave})^2 / (n-1) \tag{7}$$

$$R_{ave} = \sum_{i=1}^{N_{sat}} R_i / N_{sat} \tag{8}$$

III. Problem Solving Process

The problem of satellite resource scheduling involves allocating computing, storage, energy, and other resources within a satellite to execute tasks. The goal is to achieve load balance across the constellation while meeting the constraints of time and resources. In this work, load balance is achieved through IPSO to allocate resources to tasks. Then, related resources are assigned tasks based on heuristic factors at specific times. Following this, both parts are elaborated on separately in the following sections.

A. Task allocation by IPSO

The users send requests to the scheduling center. Tasks' attributes include the task's type, priority, and target latitude and longitude. The scheduling center analyzes the tasks submitted by the user and calculates the resource utilization of the tasks. The IPSO algorithm implements task allocation. The parameters in the IPSO algorithm are shown in TABLE II.

TABLE II. Parameters in IPSO

Parameter	Definition
N	Number of particles
P_0^i	The initial position of the ith iteration
V_0^i	The initial velocity of the ith iteration
P_{best}	The best position of the particle swarm
F_{best}	The best fitness of the particle swarm
p_{best}^i	The best position of the ith iteration
ω	Inertia weight for velocity
C_1	The individual cognition factor of the particle
C_2	The swarm cognition factor of the particles

The overall process of the algorithm is as follows: first, initialize the position and velocity of the particle swarm, the best position p_{best} currently searched by the particle swarm, the best position P_{best} searched by the particle swarm, and the optimal fitness F_{best} of the particle swarm. Then calculate the fitness of the particles and judge whether the population reaches the maximum number of iterations. If so, terminate the iteration and output the optimal solution. Otherwise, update the five parameters above, calculate the fitness function, and perform the next iteration.

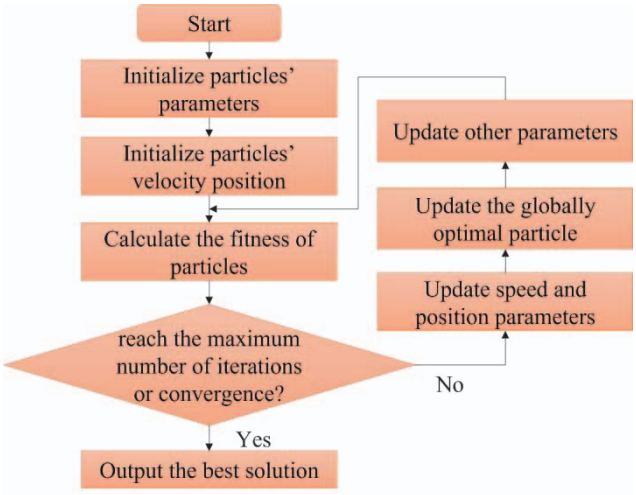

Fig. 2. The process of the IPSO algorithm

A representation method for task allocation is designed, as shown in Fig. 3. The specific time is not considered in the mission assignment phase, but the satellite's ability to perform the mission is under consideration. And then schedule the exact time after the task assignment is completed. In this way, the dimensionality of the position and velocity of each particle can be reduced sharply. Because each task has multiple time windows on each satellite, considering both in the particle swarm search phase will significantly increase the complexity of the search. Constraint checking will be correspondingly more difficult. Each particle in the swarm has position parameters {P_1, P_2,...,

$P_M\}$ and velocity parameters $\{V_1, V_2,..., V_m\}$, and both parameters have M dimensions. M represents the number of tasks.

Where $P_i \in [0, N_s - 1]$, and $V_i \in [-V_{max}, +V_{max}]$.

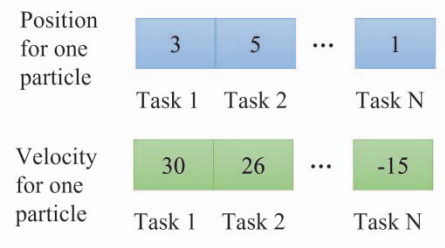

Fig. 3. Representation for task allocation

Next, the update method for particle position and velocity is designed. The particle velocity is updated in each update process according to (9). The particles' speed update consists of three parts: the inertia part, the particle's individual cognition, and the particle's swarm cognition. They correspond to the three sections on the right side of (9).

$$V_i^{'} \leftarrow \omega \bullet V_i + C_1 \bullet r_1 \bullet (p_{best} - P_i) + C_2 \bullet r_2 \bullet (P_{best} - P_i) \quad (9)$$

The position of particles is updated according to (10).

$$P_i' \leftarrow [P_i + V_i / 10] \quad (10)$$

Since the velocity value is continuous, the position value is an integer value. This update method is adopted to maintain the perceptual properties of the particle swarm algorithm and ensure that updated position values are integers. The following improvements are made. The first step of the update method is scaling the speed value and adding it to the position value to obtain a new position value. The updated position value is then rounded and judged whether it is in the range of position values. If it exceeds this range, it will be replaced by the boundary value. The update of particle positions is completed.

Fitness function design. In task allocation, load balance should be considered. Therefore, the variance of satellite resource status is used as the evaluation function. It is calculated as (11). Subtract the resources consumed by the task from the current satellite resource state, calculate the variance of the remaining satellite resources, and evaluate the effectiveness of the task allocation.

$$fit = \sum_{i=1}^{N} (R_i - R_{ave})^2 / (n-1) \quad (11)$$

B. Resource arrangement by heuristic factors

The resource scheduling process consists of two parts. The first part is task allocation using the IPSO algorithm to allocate required resources to the corresponding satellite. The second part involves determining the specific time to provide the required resources for the task. Although IPSO enables each satellite to obtain a list of tasks to execute, there may be more than one execution time window during the resource scheduling period, and conflicts may arise between tasks. Therefore, it is crucial to determine the specific time to provide resources for each task to avoid conflicts and prevent task execution failure. They are sorted according to their executable windows' start time to schedule specific times for executing tasks and then arrange the time following this order. The time arrangement process for tasks is shown in Fig. 4.

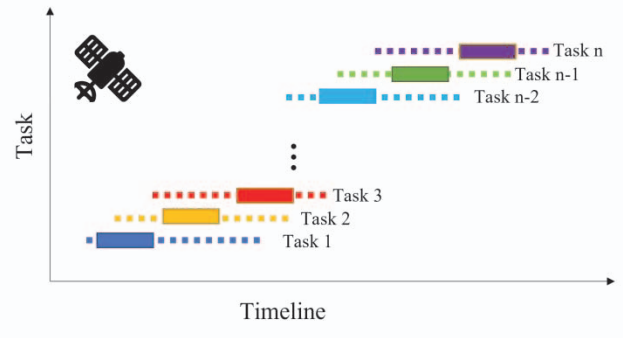

Fig. 4. The time arrangement process for tasks

IV. Experiment and Results

A. Experiment parameters

This section introduces the experimental process and simulation results. STK (Systems Tool Kit) is used to simulate the position relationship and visibility between satellites and task target points and analyze the executable time window of the tasks.

The satellites' orbit is determined by orbital elements. The orbit's semi-major axis is denoted by ***a***, while ***e*** represents the orbital eccentricity. ***i*** refers to the orbital inclination and ω denotes the true anomaly. ***RAAN*** stands for the right ascension of ascending node. M_0 represents the mean anomaly. Referring to the experimental part of the article [16], a constellation to schedule resources consisting of six satellites is designed. The satellites' orbit parameters are presented in TABLE III. The scheduling period is from 2022/11/09 00:00:00.00 to 2022/11/09 24:00:00.00.

TABLE III. Orbital Parameters of the Satellites

a/km	*e*	*i*	ω	*RAAN*	M_0
7200	0	96.576	0	175.5	0
7200	0	96.576	0	145.5	30
7200	0	96.576	0	115.5	60
7200	0	96.576	0	85.5	90
7200	0	96.576	0	55.5	120
7200	0	96.576	0	25.5	150

Each satellite collects and transmits its initial resource state back to the control center during operation. A random number between 160 and 180 is generated for simulation purposes. TABLE IV. displays the satellite resource states used in this experiment. The task's resource requirements are created using random numbers between 2 and 7.

Tasks are randomly generated from 5 ° N-55 ° N and 80 ° E-130 ° E, which can verify the ability of resource scheduling algorithms in satellite systems under dense demand. The task time is set to 12s, including 10s task execution time and 2s task switching time. Simulations of five scenarios comprising 50, 100, 150, 200, and 250 tasks are conducted to validate the algorithm's efficacy.

TABLE IV. INITIAL RESOURCES LEFT IN SATELLITES

Satellite	Initial resource left
SAT01	180
SAT02	176
SAT03	162
SAT04	166
SAT05	177
SAT06	180

The experiments were conducted using a laptop with an Intel Core i7-127000CPU, Nvidia GeForce RTX 3060 GPU, and 16GB of RAM. The programming language utilized was Python. The initial parameter settings of the IPSO algorithm are presented in TABLE V.

TABLE V. INITIAL PARAMETERS FOR IPSO

Parameter	Value	Parameter	Value
ω	1	N_{SAT}	5
C_1	2	$Iter_{Num}$	2000
C_2	2	$V_{\max}$	30

B. Results and discussion

This section presents an evaluation of the effectiveness of the proposed resource scheduling algorithm. The performance of the algorithm is estimated by analyzing the experimental results. Fig. 5 shows the convergence of the algorithm and the variation of resource variance during the scheduling process. In this scenario, 100 tasks are assigned using the IPSO algorithm. It can be observed that the algorithm converges after approximately 142 iterations. At this point, the resource variance is low at about 0.5, indicating that the satellites have achieved load balance.

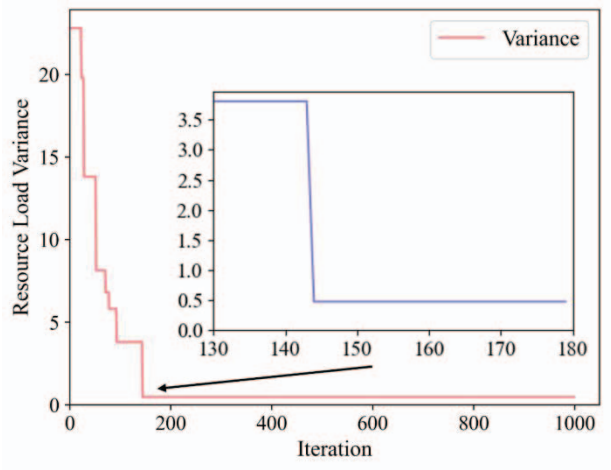

Fig. 5. Algorithm convergence in the 100-task scenario.

The resource scheduling method was tested 20 times in scenarios with 50, 100, 150, 200, and 250 tasks. The results are shown in Fig. 6. The test results for average satellite resource variance were 0.5722, 2.3889, 5.5, 14.1333, and 31.2167. The resource variance between satellites is very low when the number of tasks is less than 200. As the number of tasks increases, the algorithm's stability gradually becomes less stable, and an unbalanced state of resources appears. Nevertheless, the objective of load balance is achieved when the task number is lower than 250.

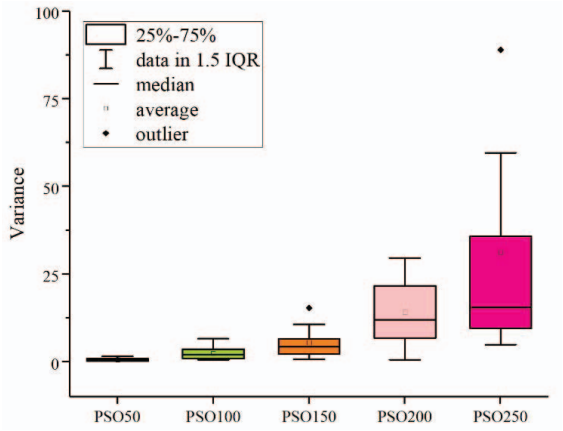

Fig. 6. Stability for the proposed method in five scenarios

Each corresponding satellite obtains the assigned tasks after using IPSO. Then the corresponding resources are assigned to the task in the timeline through a heuristic factor. In this work, tasks are sorted according to the start time of the task execution window, which is the heuristic factor. Furthermore, tasks with an early start time will be first arranged. The resource arrangement results in the timeline in the scenario of 100 tasks are shown in the following TABLE VI.

TABLE VI. RESULTS OF TIME ARRANGEMENT FOR TASKS IN THE SCENARIO OF 100 TASKS.

taskID	*satID*	*VWs*	*VW$_e$*	*EW$_s$*	*EW$_e$*
0	SAT01	101	948	101	111
1	SAT03	0	24	0	10
2	SAT02	0	339	0	10
3	SAT02	0	613	156	166
…	…	…	…	…	…
98	SAT03	31708	32501	31708	31718
99	SAT05	18301	19099	18301	18311

The second phase resolves the time conflict of scheduling resources for the tasks. With a small number of tasks, it is better able to provide the required resources for the tasks. The task completion rate is high, and the first objective of the article is better achieved.

V. CONCLUSION

This paper studied the satellite resource scheduling problem, which was modelled regarding resources and constraints. The resource scheduling was then divided into two parts: task assignment and resource timing arrangement. The task assignment was conducted using an improved particle swarm algorithm, and resource timing arrangement was performed through a heuristic factor. Experiments were conducted in scenarios involving 50, 100, 150, 200, and 250 tasks for six satellites to verify the feasibility of the proposed algorithm. Results showed that the proposed method could achieve load balance in scenarios with average resource variance. This approach is suitable for satellite resource scheduling scenarios and simplifies the representation of task

assignment thus reducing the search complexity and improving the search efficiency.

The future work will be dedicated to improving the algorithm in two directions: (1) Further enhance the algorithm's universality to enable it to adapt to highly complex satellite resource scheduling scenarios in the future. (2) Improve the algorithm's effectiveness to provide optimal load balancing in scenarios with thousands of tasks.

ACKNOWLEDGMENT

The authors would like to show great gratitude to the members of the iSAT team for their supervision and help on this paper.

REFERENCES

[1] W. J. Wolfe and S. E. Sorensen, "Three Scheduling Algorithms Applied to the Earth Observing Systems Domain," vol. 46, no. 1 Management Science, pp. 148–166, 2000.

[2] Z. Zhang, Y. Chen, L. He, L. Xing, and Y. Tan, "Learning-driven many-objective evolutionary algorithms for satellite-ground time synchronization task planning problem," Swarm and Evolutionary Computation, vol. 47, pp. 72-79, 2019.

[3] L. Wei, L. Xing, Q. Wan, Y. Song, and Y. Chen, "A Multi-objective Memetic Approach for Time-dependent Agile Earth Observation Satellite Scheduling Problem," Computers & Industrial Engineering, vol. 159, pp. 107530, 2021.

[4] H. Zhang, R. Liu, A. Kaushik, and X. Gao, "Satellite Edge Computing With Collaborative Computation Offloading: An Intelligent Deep Deterministic Policy Gradient Approach," IEEE Internet of Things Journal, vol. 10, No. 10, pp. 9092-9107, 2023.

[5] E. Gures, I. Shayea, S. Saad, M. Ergen, A. El-Saleh, N.Ahmed, and M. Alnakhli, "Load balancing in 5G heterogeneous networks based on automatic weight function," ICT Express, 2023.

[6] S. Ding, H. Du, N. Xia, S. Li, and Y. Yu, "Study on Gibbs Optimization-Based Resource Scheduling Algorithm in Data Aggregation Networks," Electronics, vol. 11, No. 11, pp. 1695, 2022.

[7] D. Zhou, Y. Wang, M. Sheng, C. Tang, and J. Li, "An integrated image task planning in satellite networks: From instruction release and observation perspective," China Communications, vol. 20, No. 5, pp. 288-301, 2023.

[8] X. Chen, G. Reinelt, G. Dai, and A. Spitz, "A mixed integer linear programming model for multi-satellite scheduling," European Journal of Operational Research, vol. 275, No. 2, pp. 694-707, 2019.

[9] M. Chen, J. Wen, Y. Song, L. Xing, and Y. Chen, "A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem," Swarm and Evolutionary Computation, vol. 65, pp. 100912, 2021.

[10] Z. Zhang, F. Hu, and N. Zhang, "Ant colony algorithm for satellite control resource scheduling problem," Applied Intelligence, vol. 48, No. 10, pp. 3295-3305, 2018.

[11] W. Liu, X. Wei, and C. Han, "Communication satellite resource scheduling based on improved whale optimization algorithm," MATEC Web of Conferences, vol. 355, pp. 03001, 2022.

[12] R. Poli, J. Kennedy, and T. J. I. Blackwell, "Particle swarm optimization," 2007.

[13] K. P. Rani, P. Sreedevi, E. Poornima, and T. S. Sri, "FTOR-Mod PSO: A fault tolerance and an optimal relay node selection algorithm for wireless sensor networks using modified PSO," Knowledge-Based Systems, vol. 272, pp. 110583, 2023.

[14] J. Bi, M. Zhao, G. Yao, H. Cao, Y. Feng, H. Jiang and D. Chai, "PSOSVRPos: WiFi indoor positioning using SVR optimized by PSO," Expert Systems with Applications, vol. 222, pp. 119778, 2023.

[15] P. Kumar and R. Kumar, "Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey," ACM Computing Surveys, vol. 51, No. 6, 2019.

[16] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen, "An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling," Computers & Operations Research, vol. 100, pp. 12-25, 2018