

## Research Article

# Exact and Heuristic Methods for Observing Task-Oriented Satellite Cluster Agent Team Formation

Hao Chen , Shu Yang, Jun Li, and Ning Jing

*College of Electronic Science, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Hao Chen; [hchen@nudt.edu.cn](mailto:hchen@nudt.edu.cn)

Received 20 May 2018; Revised 29 June 2018; Accepted 15 July 2018; Published 6 August 2018

Academic Editor: Piotr Jędrzejowicz

Copyright © 2018 Hao Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of aerospace science and technology, Earth Observation Satellite cluster which consists of heterogeneous satellites with many kinds of payloads appears gradually. Compared with the traditional satellite systems, satellite cluster has some particular characteristics, such as large-scale, heterogeneous satellite platforms, various payloads, and the capacity of performing all the observation tasks. How to select a subset from satellite cluster to perform all observation tasks effectively with low cost is a new challenge arising in the field of aerospace resource scheduling. This is the agent team formation problem for observation task-oriented satellite cluster. A mathematical scheduling model is built. Three novel algorithms, i.e., complete search algorithm, heuristic search algorithm, and swarm intelligence optimization algorithm, are proposed to solve the problem in different scales. Finally, some experiments are conducted to validate the effectiveness and practicability of our algorithms.

## 1. Introduction

With the widespread use of spatial information in modern society, the Earth Observation Satellite (EOS) and its related technologies play an important role in the agriculture, industry, science detection, and other fields [1, 2].

EOSs circle on fixed orbits to acquire images of the Earth's surface (for certain targets), which are with respect to many constraints, such as orbit constraints, energy constraints, and payload constraints. How to make full use of the precious EOSs resource, resolve the satellite conflicts effectively, and maximize the satisfaction of users has aroused the concern of scholars all over the world.

Currently, the object of satellite observation task scheduling is the whole EOSs set. The aim of scheduling is to generate the observation activities for each satellite, which should be complied with all the constraints of the satellite and meet the user observation requirements as much as possible. The current research work can be divided into two categories. They are centralized observation task scheduling and distributed observation task scheduling.

In centralized observation task scheduling methods, Constraint Satisfaction Problem (CSP) model [3], Mixed Integer Programming (MIP) model [4] and directed acyclic

graph model [5] are employed to formulate the problem. And then, greedy algorithm [6, 7], heuristic search algorithms [4, 8–10], and intelligent optimization methods [11–13] are adopted to solve the problem. The satellites under scheduling are usually homogeneous; they have the same constraints not only in satellite platform but also in payloads.

The second category is distributed observation task scheduling methods. In these methods, EOS is formulated as a cognitive agent, which is autonomous, cooperative, and socialized. So homogeneous, even heterogeneous satellite agents can form a complicated multiagent system (MAS). Satellite agents can coordinate with each other autonomously and assign earth observation task to the suitable satellite agent in MAS. Coordinating and cooperating methods in these research works are usually based on the auction method.

S. Peng et al. [14] presented a 4-satellite agent task coordination method based on contract network protocol with heuristic scheme fusion strategy. C. Wang et al. [15] propose a cooperative coevolutionary method with transfer learning operator to solve this problem in a dynamic situation.

In order to handle the larger scale satellite agents task scheduling problem, P. Feng et al. [16] bring forward a heuristic hybrid contract network protocol method. In the bid stage, cluster method is used to decrease the bid number in each

worker agent. In the evaluation stage, genetic algorithm is employed to search for high benefit of multiagent bids in manager agent. The number of satellite agents needs to be under 10.

Compared with centralized observation task scheduling methods, distributed observation task scheduling methods are more flexible. They can formulate heterogeneous EOSs to heterogeneous cognitive agents and can solve the heterogeneous satellite observation task scheduling problem more effectively and efficiently. Furthermore, they are instinct parallel algorithm. Different satellite agent can be deployed to different computation node in the same network. So computation efficiency can be improved with parallel computing technology.

But with the increasing of satellite agents' number, satellite agents coordinating and collaborating cost grow significantly [16].

In current research works, the satellite task scheduling problem is always classified to the oversubscribed problem [17], which means that satellite resource cannot meet all the requirements in one scheduling cycle. So the processing of current satellite task scheduling is to determine a subset of observing tasks. When satellites perform the subset of observing tasks, all constraints (both satellite platform and payloads) are satisfied and users have the optimal or near-optimal benefit. High benefit means more number of high priority tasks to be performed or less energy satellite consumption for performing such observation activities. Moreover, the number of satellites involved in scheduling is usually in one to a dozen or so.

With the development of aerospace science and technology, satellite cluster which consists of many kinds of heterogeneous satellites appears gradually [18]. Such as SMARA satellite cluster for Jupiter atmospheres detection [19], ANTS satellite cluster for asteroid detection [20], OLFAR satellite cluster for astronomical observation [21], and Dove satellite cluster for real-time observation of the earth [22]. The scale of these satellite clusters is between dozens to hundreds.

The Chinese government is making efforts to integrate earth observation resources belonging to different department in order to form a unified aerospace observation network. In the future, large-scale satellite cluster (Figure 1) will be an important form of earth observation resources [23].

Compared with the conventional EOSs system, EOSs cluster has some significant characteristics as follows:

(a) The scale of EOSs cluster is much larger than the conventional EOSs system. The number of member satellite in EOSs cluster is usually between dozens to hundreds, while the satellite number of the conventional EOSs system is usually one to a dozen or so.

(b) The member of satellites in EOSs cluster is highly heterogeneous. The platform, payload, and orbit of satellites in EOSs cluster may be much different from each other.

(c) The satellite cluster is usually designed to perform some specific mission such as scientific observation in some specific area [24]. In other words, the satellite cluster is specific task oriented. So for a given observation task set, the satellite cluster can perform all the tasks in one scheduling circle usually.



FIGURE 1: The illustration of large-scale satellite cluster.

So for a satellite cluster, the aim of scheduling is to find a subset from the whole satellite set which can perform all the given earth observation tasks without any constraints violation and minimize the satellite usage cost of the satellite cluster.

From the perspective of distributed observation task scheduling, we can consider each satellite as an agent, and the whole satellite cluster is a multiagent system. So the subset of satellite selected to perform the given observation tasks is called the task-oriented satellite cluster agent team. This problem is called the agent team formation for observation task-oriented satellite cluster.

The scheduling object, the aim of scheduling, and the problem scale of satellite cluster agent team formation is different from the traditional satellite task scheduling problems. Although we can learn from the ideas of traditional satellite task scheduling approaches, it is hard to adopt existing satellite task scheduling method to our problem effectively.

In a satellite cluster, different satellite has different platform and payload, so their skills and capabilities are different from each other. For a given observation task set, a satellite agent team usually has different cost to perform the tasks if the satellite agent team has different members. How to form a rational task-oriented satellite agent team which has no extra observation ability and minimal usage cost has become a new and urgent problem in the field of Earth Observation Satellite task scheduling.

How to form a team for achieving a given set of tasks is also an important issue in multiagent systems [25]. Okimoto et al. focus on how to form a mission oriented robust team. Robustness means the ability to reach the goal even if some agents break down in the task performing process [26]. A branch-and-bound search-based algorithm, an approximate algorithm with local search operator [27], some greedy algorithms, and a multiobjective evolution algorithm based method [28] are employed to solve the problem. But it is unlikely that a satellite breaks down in the process of performing observation tasks. The robustness has been taken into account when the EOS is designed.

Marcolino et al. [29] focus on the diversity of an agent team and prove that a diverse team can overcome a uniform team. In the context of robot team formation, Liemhetcharat et al. consider the probability of module failures of robots and study how to form a multirobot team that is robust to failures [30]. Farhangian et al. [31] and Rochlin et al. [32] take personality, interest, and mind into account in the agent team formation process. They maintain that agent team performance is significantly influenced by the makeup of participant personalities and temperaments and goes beyond the analysis of individual skills. But in our problem, we make a hypothesis that the goal of all the satellite agents in a cluster is to complete tasks; they do not have any minds to maximize their self-benefit.

There are also some scholars who study the agent team formation problem in various application fields, such as RoboCup [33], rescue problem in fire field [27, 34], unmanned aerial vehicles coalition formation strategies [35], agent team formation in social networks [36–38], business projects [39], engineering projects [40], and design problem [41]. In each specific field, there are some particular goal and constraints needing to be considered. The methods cannot be applied to our problem effectively.

To the best of our knowledge, no previous studies have investigated satellite cluster observing task scheduling problem from the perspective of agent team formation.

The contributions of this paper can be summarized as follows:

- (i) We study the observing task-oriented satellite cluster agent team formation problem, which is a new challenge in the field of aerospace resource scheduling and aims to propose the reasonable scheduling model and objective function according to the actual usage of satellite cluster.
- (ii) Three novel algorithms which are a complete search algorithm, a heuristic search algorithm, and a swarm intelligence optimization algorithm are designed to solve the problem. The time complexity of the algorithms is analyzed.
- (iii) After analyzing computational results in experiment, we can make the conclusion that the three algorithms are scalable for problems with different scale.

The rest of this paper is organized as follows. Section 2 describes the mathematical formulation on the agent team formation problem for observation task-oriented satellite cluster. In Section 3, a complete algorithm based on depth-first branch-and-bound search is proposed. Then the time complexity of the complete algorithm is analyzed. In order to conquer the high time complexity of the complete algorithm, we design an approximation algorithm with a heuristic pruning strategy in Section 4. And then the time complexity of the heuristic approximation algorithm is analyzed. In Section 5 a heuristic algorithm based on Fireworks Algorithm (FWA) [42] is proposed, which is a novel swarm intelligence optimization method. Section 6 presents the numerical experiments to demonstrate the ability and suitable scenario of the proposed algorithms. The last section concludes the

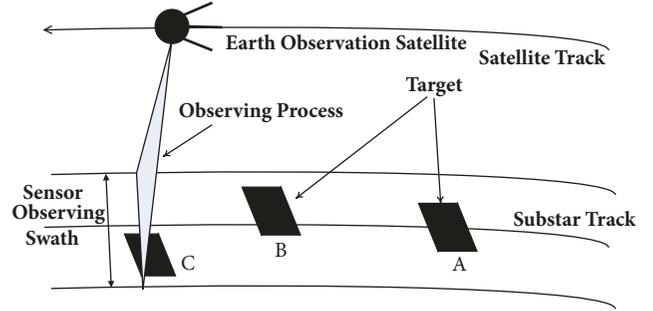


FIGURE 2: The working process of an EOS.

paper and provides several future extensions of our algorithms.

## 2. Problem Formulation

EOSs circle the earth to take images of the interested area (observation target, we use target for short in the remaining sections) in the vicinity of its substar track with certain work mode pattern (rolling angle of a satellite to point to target). Since the orbit is fixed, satellite can only access a certain target in certain time windows which are called target access windows (TAWs).

The process is illustrated in Figure 2.

Agent team formation for observation task-oriented satellite cluster is to select a satellite agent team from the heterogeneous satellite cluster, which can perform all the given earth observation tasks without any constraints violation and minimize the satellite usage cost of the whole satellite cluster. For the characteristic of EOS cluster observation, some hypotheses are made as follows:

(a) Target access windows have been calculated before scheduling. One target access window corresponds with one observation task.

(b) Once a satellite starts to perform an observing task, the action must be completed before the start of the next task.

(c) Each target only needs to be observed once. That means if there are  $n$  targets, each target can be performed by  $m$  satellites, then there are  $n \times m$  tasks. Each target only should be observed by one satellite.

**2.1. Problem Description.** To model the agent team formation problem for observation task-oriented satellite cluster, some notations are introduced as follows:

(a) Given observation target set, denoted as  $TARGET$ ,  $\forall i \in TARGET$ , it can be formulated as  $i = (ReqTW_b^i, ReqTW_e^i, ReqPL_i, ReqRL_i)$ .  $ReqTW_b^i$  and  $ReqTW_e^i$  are the earliest observation request and latest observation request. That means the observation time of the target  $i$  should not be earlier than  $ReqTW_b^i$  or later than  $ReqTW_e^i$ .  $ReqPL_i$  is the observation payload request of the target  $i$ . Only the satellites which are equipped with the required observation payload are qualified to do observation to target  $i$ . Observation payload is diverse, including visible light sensor, infrared sensor, synthetic aperture radar, and hyperspectral sensor.

$ReqRL_i$  is the request for the lowest sensor resolution of the target  $i$ . The resolution of satellite which performs the observation to the target  $i$  should be higher than  $ReqRL_i$ .

(b) Given a satellite cluster, denoted as  $SAT$ ,  $\forall k \in SAT$ , it can be formulated as  $k = (PLSet^k, \Delta T_{pc}^k, \Delta T_{cir}^k, \Delta T_{day}^k, UtTaskNum_{Sat}^k, CofCost_{Sat}^k, \beta^k, \theta^k)$ .  $PLSet^k$  is the observation payload set of satellite  $k$ . There may be more than one type of sensor located in a satellite.  $\Delta T_{pc}^k$  is the minimal duration of working pattern changing from one pattern to another. When a satellite takes image of a target, it needs to change its rolling angle to point the target. So when a satellite performs continuously observation from target A to target B, it needs to change its rolling angle from pointing to A to pointing to B. The changing duration cannot be less than  $\Delta T_{pc}^k$ .  $\Delta T_{cir}^k$  and  $\Delta T_{day}^k$  are the maximal working duration of the sensor of satellite  $k$  in one circle and one day.  $UtTaskNum_{Sat}^k$  is the number of observation tasks the satellite  $k$  undertakes. In the process of satellite agent team formation, if satellite  $k$  undertakes new tasks,  $UtTaskNum_{Sat}^k$  increases.  $CofCost_{Sat}^k$  is the usage cost coefficient of satellite  $k$ .  $\beta^k$  and  $\theta^k$  are the continuing service capability coefficient of satellite  $k$ , which will be described in formula (3) (in Section 2.2).  $CofCost_{Sat}^k$ ,  $\beta^k$ , and  $\theta^k$  are closely related to the cost of the satellite agent team.

(c) There is an observation task set denoted as  $TASK$ .  $\forall j \in TASK$ , it can be formulated as  $j = (W\_Set_j, Tar_j, Cost_{Task}^j, x_j)$ .  $W\_Set_j$  is the target access window set of the observation task  $j$ .  $\forall W_j^k \in W\_Set_j$ ,  $W_j^k = [t_{bj}^k, t_{ej}^k]$  is a target access window of satellite  $k$  and target  $Tar_j$ .  $t_{bj}^k$  and  $t_{ej}^k$  are the beginning time and end time of the time frame that is available for satellite  $k$  to perform observation task  $j$ .  $Tar_j \in TARGET$  is the target which is corresponding to the observation task  $j$ .  $Cost_{Task}^j$  is the cost coefficient of performing task  $j$ .  $x_j$  is the decision variable of our problem.  $x_j = (x_j^{sat}, x_j^w)$ .  $x_j^{sat} \in SAT$  indicates which satellite will perform the observation task  $j$ . And  $x_j^w \in W\_Set_j$  indicates the observation task  $j$  which will be performed in target access window of  $W\_Set_j$ .

(d) The satellite agent team, which is a subset of satellite cluster that can perform all the observation tasks, is  $SatAgentTeam$ . We can figure out the  $SatAgentTeam$  according to decision variable  $x_j$ . Obviously,  $SatAgentTeam \subseteq SAT$ .

**2.2. Model Construction.** We need to find a satellite agent team from the whole satellite cluster that has the minimal cost to perform all the observation tasks. Based on the description and assumption of the problem, we give our model as follows:

$$TeamCost = \sum_{j \in TASK} Cost_{TS}^j + \sum_{k \in SAT} Cost_{Sat}^k \rightarrow MIN \quad (1)$$

The whole cost of a satellite agent team consists of two parts.

The first part is  $\sum_{j \in TASK} Cost_{TS}^j$ . It is the sum of the cost of each target to be performed. If task  $j$  is performed by satellite  $k$ , the cost of task performing can be formulated as

$$Cost_{TS}^j = Cost_{Task}^j \times CofCost_{Sat}^{x_j^{sat}} \quad (2)$$

The second part is the  $\sum_{k \in SAT} Cost_{Sat}^k$  ( $k = x_j^{sat} \in SAT$ ). It is the sum of accumulate fatigue cost for each satellite, which is related to the undertaken task count of each satellite. The more task a satellite undertakes, the higher the accumulated fatigue cost of the satellite will be. The accumulated fatigue cost of satellite  $k$  can be formulated as

$$Cost_{Sat}^k = \left( \frac{UtTaskNum_{Sat}^k}{\beta^k} \right)^2 \times \theta^k \quad (3)$$

From formula (3), we can see that the increment of accumulated fatigue cost can be more if a satellite has already undertaken a lot of tasks than the situation that a satellite has only undertaken a few tasks.

The mainly constraints in the model we consider are as follows.

(C1) *The Time Window Constraint for Observation Tasks.* The observation time of the target  $Tar_j$  should not be earlier than  $ReqTW_b^{Tar_j}$  or later than  $ReqTW_e^{Tar_j}$

$$\begin{aligned} & \forall j \in TASK, \\ & \forall W_j^k \in W\_Set_j, \\ & \rightarrow ReqTW_b^{Tar_j} \leq t_{bj}^k < t_{ej}^k \leq ReqTW_e^{Tar_j} \end{aligned} \quad (4)$$

(C2) *The Constraint of Observation Payload and Resolution Requirement Matching.* The payload and the resolution of a satellite which will perform an observation task need to match the payload and resolution requirement of the target.

$$\begin{aligned} & \forall j \in TASK, \\ & \forall W_j^k \in W\_Set_j, \\ & \rightarrow ReqPL_{Tar_j} \cap PLSet^k \neq \emptyset \end{aligned} \quad (5)$$

(C3) *The Constraint of Sensor Working Mode Changing Span.* It takes some time to change from one working pattern to another. The period of changing pattern from task  $g$  to  $h$  should be longer than  $\Delta T_{pc}^k$ .

$$\begin{aligned} & \forall g, h \in TASK, \\ & \forall W_g^k, W_h^k \in W\_Set_j, \\ & t_{bg}^k \geq t_{bh}^k, \\ & \rightarrow (t_{bg}^k - t_{eh}^k) \cdot Do(x_g^k) \cdot Do(x_h^k) \\ & \geq \Delta T_{pc}^k \cdot Do(x_g^k) \cdot Do(x_h^k) \end{aligned} \quad (6)$$

Input: TASK: observation task set  
 SAT: satellite cluster

Output: The optimal satellite agent team  $team_M^{best}$

Begin

(1) set  $team_0 = team_M^{best} = \emptyset$ ,  $team = team_0$

(2) TF4EachTask(1,  $team$ , TASK, SAT,  $team_M^{best}$ )

(3) Return  $team_M^{best}$

End

ALGORITHM 1: STCFDB.

in which  $k \in SAT$  and  $Do()$  is the function of indicating whether the current task will be performed. For a task  $j$ , if it will be performed,  $Do(x_j^k)=1$ ; else  $Do(x_j^k)=0$ .

(C4) *The Constraint of Sensor Maximal Working Time in One Orbit Circle.* For a satellite  $k$ , the accumulated working duration of the sensor in a single satellite orbit circle could not be longer than  $\Delta T_{cir}^k$ .

$$\begin{aligned} &\forall j \in TASK, \\ &\forall W_j^k \in W\_Set_j, \\ &\forall q \in \mathbb{N} \\ &\rightarrow \sum_j [(t_{ej}^k - t_{bj}^k) \cdot Circle(j, k, q)] \leq \Delta T_{cir}^k \end{aligned} \quad (7)$$

in which  $Circle((j, k, q)$  is the function of indicating whether the satellite  $k$  will perform task  $j$  in circle  $q$ . if it will,  $Circle((j, k, q) = 1$ ; else it equals 0.

(C5) *The Constraint of Sensor Maximal Working Time in One Day.* For a satellite  $k$ , The accumulated working duration of the sensor in a single day could not be longer than  $\Delta T_{day}^k$ .

$$\begin{aligned} &\forall j \in TASK, \forall W_j^k \in W_{Set_j}, \forall d \in \mathbb{N} \\ &\rightarrow \sum_j [(t_{ej}^k - t_{bj}^k) \cdot Day(j, k, d)] \leq \Delta T_{day}^k \end{aligned} \quad (8)$$

in which  $Day(j, k, d)$  is the function of indicating whether the satellite  $k$  will perform task  $j$  in the day  $d$ . if it will,  $Day(j, k, d) = 1$ ; else it equals 0.

### 3. The Complete Algorithm for Task-Oriented Satellite Agent Team Formation Based on Depth-First Branch-and-Bound Search

From the model described above, we can see that task-oriented satellite agent team formation is a complicated combinatorial optimization problem. We would like to adopt an exact search method based on branch-and-bound search-based technology.

Constraints C1 and C2 are satellite and target matching constraint. We can handle with constraint C1 and C2 using

rule based expert system [43]. For example, if the satellite does not have the required observation payload, the satellite will not exist in the candidate satellite set of performing the target. So since we have effective rules, we only need to perform the constraints C3 and C4 in the searching process.

The key searching process is as follows. Firstly, we build an empty satellite agent team (denoted as  $team_0$ ) as the root of the searching tree. Each child of the root indicates a matching case of satellite set and task 1. Secondly, we build subtrees of the parent node to indicate the matching of satellite set and task 2. Then we build subtree of the parent node to handle task 3 and so on. The unique path from root to a node indicates a case of satellites task performing series. So the height of the searching tree equals the total task number. When the traversal of all tasks is completed, the searching process is finished.

We propose task-oriented Satellite agent Team Complete Formation algorithm based on Depth-first traversal with Branch-and-bound search (STCFDB). The main steps of STCFDB are illustrated in Algorithm 1.

In STCFDB, we adopt the recursive procedure TF4Each-Task to obtain the best satellite agent team formation solution  $team_M^{best}$ . The main steps of TF4EachTask are shown in Algorithm 2.

In TF4EachTask, we implement tree depth-first traversal by recursive method. In statement (2), function *Evaluation* calculates the cost of the team. Statement (3) is the termination condition of our algorithm. The condition  $m > M$  means that all the observation tasks have been handled. If the cost of the team is lower than that of the current best team, the current best solution will be replaced. Statement (7)-(9) is the branch-and-bound approach. If the cost of the team is higher than that of the current best solution, and the searching process is terminated. So, all subtrees of the current node are pruned. In statement (10)-(12), Function *AddSol2Team* is to generate all new teams to handle the next task. And then, the recursive process is performed in statement (11) and (12). The main steps of *AddSol2Team* are in Algorithm 3.

In *AddSol2Team*, for the  $i$ -th task from observation task set, we only need to add each pair of satellite (which is capable of performing the task) and the  $i$ -th task to existing satellite agent team set which has already handled task 1 to task  $i-1$ .

For example, if there are three tasks and four satellites, every satellite can perform all the three tasks. The partial searching tree can be illustrated as Figure 3.

```

Input:  m: observation task index
        team: current team under formation
        TASK: observation task set
        SAT: satellite cluster
        teambest: The best team has been found
Output: The best team has been found teambest
Begin
(1)  $M = |TASK|$ 
(2)  $team.Cost = Evaluation(team)$ 
(3) if ( $m > M$ )
(4)   if ( $team.Cost < team^{best}.Cost || team^{best} = \emptyset$ )
(5)      $team^{best} = team$ 
(6)   return
(7) else
(8)   if ( $team.Cost > team^{best}.Cost \&\& team^{best} \neq \emptyset$ )
(9)     return
(10)  $teamSet = AddSol2Team(m+1, team, TASK, SAT)$ 
(11) foreach team in teamSet
(12)   TF4EachTask( $m+1, team, TASK, SAT, team^{best}$ )
End

```

ALGORITHM 2: TF4EachTask.

```

Input:  i: observation task index
        team: current team under formation
        TASK: observation task set
        SAT: satellite cluster
Output: The sub-tree set of current node
Begin
(1) set  $teamSet = \emptyset$ 
(2)  $task = TASK[i]$ 
(3) foreach sat in SAT
(4)   if (sat can perform task)
(5)      $new\_team = team + (sat, task)$ 
(6)      $teamSet = teamSet + \{new\_team\}$ 
(7) return  $teamSet$ 
End

```

ALGORITHM 3: AddSol2Team.

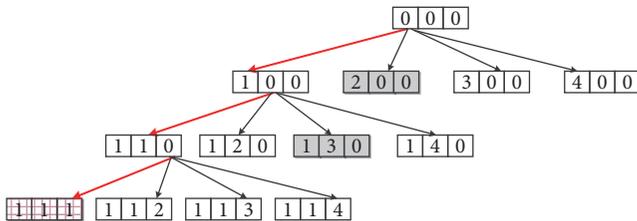


FIGURE 3: Illustration of the partial searching tree.

In Figure 3, node 

1	4	3
---	---	---

 means that task 1 will be performed by satellite 1, task 2 will be performed by satellite 4, and task 3 will be performed by satellite 3. The depth-first traversal process is shown as arrows. And the gray node is the node pruned by branch-and-bound strategy.

From the pseudocode of STCFDB we can see that if the task number is  $M$ , the satellite number is  $N$ , each satellite can perform all the tasks, and there is no pruning in the searching process, the time complexity of the exact algorithm is  $O(N^M)$ . In other words, in some bad case, STCFDB has exponential time complexity.

#### 4. The Heuristic Algorithm for Task-Oriented Satellite Agent Team Formation Based on Breadth-First Search with Pruning Strategy

As analyzed in Section 3, the time complexity of STCFDB is exponential, which cannot be applied to large-scale task-oriented satellite agent formation problem. In order to reduce computing time of our algorithm, we introduce a heuristic tree pruning strategy and propose task-oriented Satellite

```

Input:  TASK: observation task set
        SAT: satellite cluster
Output: The optimal or sub-optimal satellite agent team  $team_M^{p,best}$ 
Begin
(1)  $M = |TASK|$ 
(2) set  $team_0 = team_M^{best} = \emptyset, i=0, TEAM_0 = \{team_0\}$ 
(3) while  $i < M$ 
(4)    $TEAM_{i+1} = \emptyset$ 
(5)   foreach  $team$  in  $TEAM_i$ 
(6)      $teamSet = AddSol2Team(i+1, team, TASK, SAT)$ 
(7)      $TEAM_{i+1} = TEAM_{i+1} + teamSet$ 
(8)   OrderbyCostAsc( $TEAM_{i+1}$ )
(9)    $TEAM_{i+1} = HeuristicPruning (TEAM_{i+1}, i + 1)$ 
(10) OrderbyCostAsc( $TEAM_M$ )
(11)  $team_M^{p,best} = TEAM_M[0]$ 
(12) return  $team_M^{p,best}$ 
End

```

ALGORITHM 4: STFBHP.

```

Input:  The team set before pruning  $Team\_set$ 
        The layer of the tree  $k$ 
Output: The team set after pruning  $Team\_Pruned$ 
Begin
(1) set  $Team\_Pruned = \emptyset$ 
(2)  $CostBase = Team\_set[0].Cost$ 
(3)  $CostInc = (CostBase \cdot \epsilon) / k^2$ 
(4)  $Team\_Pruned = Team\_Pruned + \{Team\_set[0]\}$ 
(5)  $Scale = 1$ 
(6) foreach  $team$  in  $Team\_set$ 
(7)   if ( $team.Cost > CostBase + CostInc$ )
(8)      $Team\_Pruned = Team\_Pruned + \{team\}$ 
(9)      $CostBase = team.Cost$ 
(10)     $Scale = Scale + \omega$ 
(11)     $CostInc = CostInc \cdot (1 + scale \cdot \epsilon / k)$ 
(12) return  $Team\_Pruned$ 
End

```

ALGORITHM 5: HeuristicPruning.

agent Team Formation algorithm based on Breadth-first traversal with Heuristic Pruning (STFBHP).

STCFDB is based on depth-first traversal. It cannot do any subtree pruning operation before a solution has been found. Furthermore, even if a solution has been found, the quality of the obtained solution is bad, there is few branches which can be pruned, and the searching space is also too large to reduce the computing time. So we need to find a new pruning strategy.

So firstly, we change depth-first traversal to breadth-first traversal. And then we introduce a heuristic pruning strategy to make sure that in one layer of the searching tree, only some selected nodes will be extended (their subtrees will be generated and add to the searching tree).

The main steps of STFBHP are shown in Algorithm 4.

In STFBHP,  $TEAM_i (i \in [0, M])$  is the satellite agent team set that can handle from  $task[1]$  to  $task[i]$ . Function

AddSol2Team in statement (6) is described as Algorithm 3. In statement (8), the function OrderbyCostAsc ( $TEAM_{i+1}$ ) sorts the teams in set  $TEAM_{i+1}$  by cost in ascending order.

HeuristicPruning in statement (9) is the heuristic pruning function. The main steps of HeuristicPruning are show in Algorithm 5.

The key idea of HeuristicPruning is nonuniform sampling. The  $Team\_set$  has been sorted by cost in ascending order, so the  $Team\_set[0]$  is the team with minimal cost. The algorithm adds it to  $Team\_Pruned$  (the agent team set after pruning) firstly in statement (4). And then the algorithm samples some teams and adds them to  $Team\_Pruned$ . In the region where the team cost is low, the sampling is dense, and in the region where the team cost is high, the sampling is sparse.  $\epsilon$  and  $\omega$  are constant parameters of STFBHP. How to set their values will be discussed in Section 6.



FIGURE 4: The distribution of the cost of satellite agent teams that can be retained in one layer of searching tree after heuristic pruning.

Because of the nonuniform sampling strategy in function HeuristicPruning, there are a lot of subtree being pruned. The searching space of STFBHP is much smaller than that of STCFDB.

The time complexity of STCFDB is analyzed in Theorem 1.

**Theorem 1.** *STFBHP is a polynomial time algorithm.*

*Proof.* From the statements of algorithm STFBHP, we can see that the running time of STFBHP is mainly related to the number of nodes retained after pruning in each layer of the searching tree. In STFBHP,  $TEAM_k$  ( $k \in [1, M]$ ) is the whole set of satellite agent teams that can handle *TASK* [1] to *TASK* [k]. (*TASK* is the set of tasks).

We suppose that  $C_{min}$  and  $C_{max}$  are the minimum and maximum team cost of items in  $TEAM_k$ . We denote  $CostInc$  (in statement (11) of HeuristicPruning) in each iteration (in statement (6)) as  $CostInc_0, CostInc_1, \dots, CostInc_{n-1}$  (supposing there are  $n$  iterations in statement (6), so there are  $n+1$  retained solutions in  $k$ -th layer of the searching tree after pruning).

From the statements (6)-(11) of HeuristicPruning, we can know

$$CostInc_0 = \frac{(C_{min} \cdot \varepsilon)}{k^2} \quad (9)$$

$$CostInc_n = CostInc_{n-1} \cdot \left(1 + (1 + n \cdot \omega) \cdot \frac{\varepsilon}{k}\right) \quad (10)$$

We denote  $C_{min}, C_1, C_2, \dots, C_n$  as the cost of satellite agent teams after heuristic pruning in  $k$ -th layer of the searching tree. The distribution of  $C_{min}, C_1, C_2, \dots, C_n$  can be illustrated as Figure 4.

In Figure 4, the increment between  $C_b, C_{b+1}$  is more than (and the closest to)  $CostInc_b$ .

From Figure 4, we can know that

$$C_{max} > C_n > C_{min} + \sum_{i=0}^{n-1} CostInc_i \quad (11)$$

From (9), (10), and (11), we can get

$$\begin{aligned} C_{max} - C_{min} &> CostInc_0 \\ &\cdot \left[1 + \sum_{x=1}^{n-1} \prod_{i=1}^x \left(1 + (1 + i\omega) \cdot \frac{\varepsilon}{k}\right)\right] \\ &> CostInc_0 \cdot \sum_{x=0}^{n-1} \left(1 + (1 + \omega) \cdot \frac{\varepsilon}{k}\right)^x \end{aligned}$$

$$\begin{aligned} &= \frac{C_{min} \cdot \varepsilon}{k^2} \cdot \frac{(1 + (1 + \omega) \cdot \varepsilon/k)^n - 1}{(1 + \omega) \cdot \varepsilon/k} \\ &= \frac{C_{min} \cdot [(1 + (1 + \omega) \cdot \varepsilon/k)^n - 1]}{k \cdot (1 + \omega)} \end{aligned} \quad (12)$$

We change (12) as the following inequality formula:

$$\left(1 + (1 + \omega) \cdot \frac{\varepsilon}{k}\right)^n < k \cdot (1 + \omega) \cdot \frac{C_{max} - C_{min}}{C_{min}} + 1 \quad (13)$$

Put the natural logarithm on both sides of (13); we can get

$$\begin{aligned} n \cdot \ln\left(1 + (1 + \omega) \cdot \frac{\varepsilon}{k}\right) \\ < \ln\left(1 + k \cdot (1 + \omega) \cdot \frac{C_{max} - C_{min}}{C_{min}}\right) \end{aligned} \quad (14)$$

so we can obtain

$$n < \frac{\ln\left(1 + k \cdot (1 + \omega) \cdot \frac{C_{max} - C_{min}}{C_{min}}\right)}{\ln\left(1 + (1 + \omega) \cdot \varepsilon/k\right)} \quad (15)$$

Because of the inequality formula  $x/(1+x) \leq \ln(1+x) \leq x$ , so we can get

$$n < \frac{1 + (1 + \omega) \cdot \varepsilon/k}{(1 + \omega) \cdot \varepsilon/k} \cdot k \cdot (1 + \omega) \cdot \frac{C_{max} - C_{min}}{C_{min}} \quad (16)$$

As we mentioned above, the number of the nodes retained after heuristic pruning in  $TEAM_k$  ( $k$ -th layer of searching tree) is  $n+1$ . So, upper bounds of the nodes (denoted as  $NodeNum_k^{max}$ ) retained in  $k$ -th layer of STFBHP can be represented as

$$\begin{aligned} NodeNum_k^{max} &= n + 1 \\ &< k \cdot [k + (1 + \omega \cdot \varepsilon)] \cdot \frac{C_{max} - C_{min}}{C_{min}} \\ &\quad + 1 \end{aligned} \quad (17)$$

In (17),  $\omega$  and  $\varepsilon$  are constant parameters and  $C_{max}$  and  $C_{min}$  in  $TEAM_k$  are also constants, so  $NodeNum_k^{max}$  is only associated with  $k^2$ . The running time of STFBHP is mainly related to the number of teams retained after pruning in each layer. The time complexity of handling the  $k$ -th layer is  $O(k^2)$ . And there are  $M$  layers in searching tree.

So the time complexity of STFBHP is  $O(1^2 + 2^2 + 3^2 + \dots + M^2) = O(M^3)$ . STFBHP is a polynomial time algorithm. Thus, the original proposition is proved.  $\square$

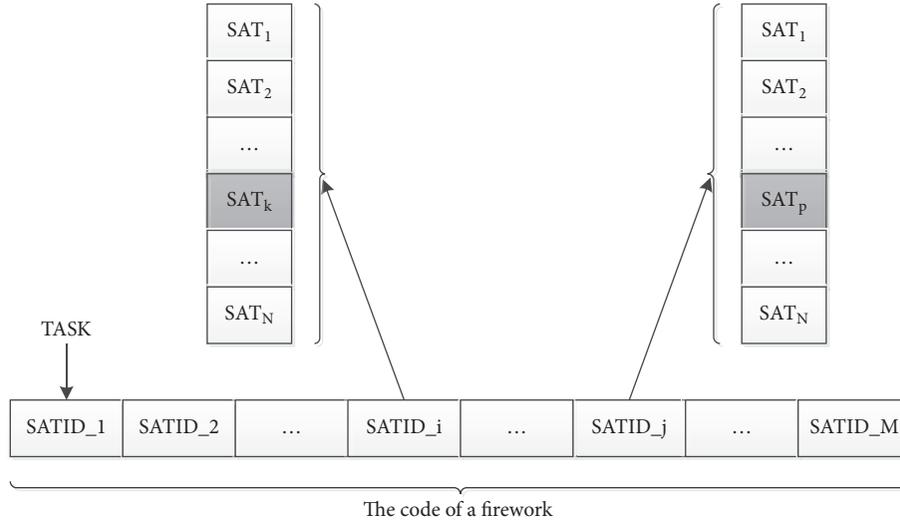


FIGURE 5: Code of STFFA.

## 5. The Swarm Intelligence Optimization for Task-Oriented Satellite Agent Team Formation Based on Fireworks Algorithm

Some studies have indicated that traditional multisatellite observation scheduling is a typical NP-Hard problem [44, 45], with no fast algorithm currently. Task-oriented satellite agent team formation is similar to the traditional multisatellite observation scheduling problem, to which exhaustive search (STCFDB) seems ineffective and the performance of heuristic method (STFBHP) may not be good enough for large-scale problem.

We intend to propose another approach which adopts swarm intelligence optimization algorithm to solve this problem.

*Fireworks Algorithm* (FWA) is one of the most outstanding swarm intelligence algorithms, which is an iterative optimization algorithm inspired by the emergent swarm behavior of fireworks [42, 46]. In the FWA, swarm fireworks explosion (search) processes are employed and mechanisms for keeping diversity of sparks are also well designed. FWA has been widely used in complex function optimization problems, combinatorial optimization, and planning problems and achieves good results [47–49].

Taking the features of our problem into account, we propose task-oriented Satellite agent Team Formation algorithm based on Fireworks Algorithm (STFFA).

**5.1. Encoding and Decoding.** In order to use FWA to solve our problem, firstly we should establish the correspondence between sparks position and problem solution. Consider the characters of our problem, we need to give the relationships between tasks and the satellites which can undertake the tasks. So we use multidimension integer encoding as is shown in Figure 5.

As illustrated in Figure 5, SATID<sub>*i*</sub> represents the ID of the satellite that completes the task *i*,  $i \in [1, M]$ . For example, SATID<sub>*i*</sub> equals SAT<sub>*k*</sub> in Figure 5, which means that satellite *k*

will perform the task *i*. We can see that the coding of STFFA is similar to that in the searching tree which is illustrated in Figure 3. But in the searching tree, the value of some positions may be zero, which means that the corresponding task has not been assigned to a satellite. But in STFFA, zero cannot be contained in the code. Every individual in the swarm group represents a solution of our problem.

**5.2. Design of Fireworks Explosion.** The key of STFFA is how to design the explosion factor of fireworks. The details of explosion factor are as follows.

**5.2.1. Number of Sparks.** In our problem, we need to find the satellite agent team with minimal cost. We can describe our problem as

$$\text{Minimize } C(X) = \text{TeamCost} \in \mathbb{R}, \quad (18)$$

where  $X = (x_1, x_2, x_3, \dots, x_M)$  is the code (in Figure 5) which denotes a location in the potential space. TeamCost is objective function as (1). Then the number of sparks generated by each firework  $x_i$  is defined as follows:

$$S_i = S_i^C + S_i^F, \quad (19)$$

where

$$S_i^C = G \cdot \frac{C_{\max} - C(X_i) + \xi}{\sum_{k=1}^n (C_{\max} - C(X_k)) + \xi} \quad (20)$$

where  $G = g \cdot \text{GroupNum} \cdot \text{TaskNum}$

$$S_i^F = H \cdot \frac{F_{\max} - F(X_i) + \xi}{\sum_{k=1}^n (F_{\max} - F(X_k)) + \xi} \quad (21)$$

where  $H = h \cdot \text{GroupNum} \cdot \text{TaskNum}$ .

From (19) we can see that the number of sparks is the sum of two parts.  $S_i^C$  is the same as the original definition of number of sparks in FWA.  $C_{\max}$  is the maximum (worst) value

of the objective function among all the fireworks in group.  $\xi$ , which denotes the smallest constant in the computer, is utilized to avoid zero-division-error. *GroupNum* is the group number of FWA, and the *TaskNum* is the total task number in the problem.  $g$  is a constant parameter of STFFA.

The other part is  $S_i^F$  which is similar to  $S_i^C$ .  $F(X_i)$  in (21) can be defined as follows:

$$F(X_i) = \sum_{i=1}^{SatNum} \left( SN_i - \frac{taskNum}{SatNum} \right)^2 \quad (22)$$

$SN_i$  is the task number undertaken by satellite  $i$ . *SatNum* is the total satellite number in satellite cluster, so if some satellites undertake too many tasks while others undertake too few tasks, the value of  $F(X_i)$  is high. But from objective function (1) we can see that the higher the value of  $F(X_i)$  is, the higher the accumulate fatigue cost is. So the less probability  $X_i$  is a good solution. And  $h$  is also a constant parameter of STFFA.  $F(X_i)$  can be considered as the load balancing degree of the whole satellite cluster.

**5.2.2. Amplitude of Explosion.** The amplitude of explosion for each firework in STFFA is defined as follows:

$$A_i = A \cdot \frac{C(X_i) - C_{min} + \xi}{\sum_{k=1}^n (C(X_k) - C_{min}) + \xi} \quad (23)$$

where  $A = a \cdot GroupNum \cdot SatNum$ .

$C_{min}$  is the minimum (best) value of the objective function among all the fireworks in the group, which is the same as the original definition of amplitude of explosion in FWA. *SatNum* is the total satellite number in satellite cluster.  $a$  is a constant parameter of STFFA.

**5.2.3. Sparks Generating.** The number and explosion amplitude of sparks have been figured out. In this section, we will discuss how to generate sparks.

In the explosion, sparks may undergo the effects of explosion from random  $z$  directions (dimensions). So we generate sparks randomly in  $z$  different dimensions as follows:

$$z = round(M \cdot random(0, 1)) \quad (24)$$

where  $M = |TASK|$  is the total number of tasks. It is also the dimension of coding in STFFA. And  $random(0, 1)$  is a uniform distribution over  $[0, 1]$ .

There are two types of sparks being generated, ordinary sparks and Gaussian sparks.

For ordinary sparks, if the location of fireworks is  $X_i = (x_1^i, x_2^i, x_3^i, \dots, x_M^i)$ , we firstly generate  $z$ -dimensions to be changed (using (24)) and then for each dimension, the value of the dimension alters randomly in the range of explosion amplitude as (23). And then the location of spark can be obtained.

$$\hat{x}_j^i = ceiling(x_j^i + A_i \cdot random(-1, 1)) \quad (25)$$

where  $x_j^i$  is the  $j$ -dimension of  $X_i$  ( $j \in [1, M]$ ) and  $\hat{x}_j^i$  is the location of spark in  $j$ -dimension. In our problem, the value

of  $x_j^i$  means that the satellite ID that undertakes task  $j$ , so  $ceiling()$ , is employed.

In order to keep the diversity of sparks, Gaussian spark is designed. For  $x_j^i$  ( $j$ -dimension of  $X_i$ ),  $\hat{x}_j^i$  can be figured out according to

$$\hat{x}_j^i = ceiling(x_j^i \cdot Gaussian(1, 1)) \quad (26)$$

If the obtained location of spark (both ordinary spark and Gaussian spark) is found to fall out of the potential space, we need to map it to the potential space according to

$$\hat{x}_j^i = \hat{x}_j^i \% (\max(x_j^i) - \min(x_j^i)) \quad (27)$$

where  $\max(x_j^i)$  and  $\min(x_j^i)$  are the maximal value and minimal value of  $j$ -dimension of  $X_i$ . In our problem,  $\max(x_j^i) - \min(x_j^i)$  is the total number of candidate satellites which can perform task  $i$ .

**5.2.4. Selection of Locations.** In STFFA, for each explosion generation, *GroupNum* locations should be selected for the firework explosion. For the first generation, the locations of fireworks explosion are generated randomly. And then, for the generation  $p+1$ , the locations of fireworks explosion are selected from the locations of fireworks and sparks in  $p$ -th generation. In STFFA, the best current location which has ever been found is always kept for the next explosion generation. The other *GroupNum* - 1 locations are selected based on their distance to other locations in order to keep diversity of explosion group. For two locations,  $X_a = (x_1^a, x_2^a, x_3^a, \dots, x_M^a)$  and  $X_b = (x_1^b, x_2^b, x_3^b, \dots, x_M^b)$ , the distance between  $X_a$  and  $X_b$  can be defined as

$$Dis(X_a, X_b) = \sum_{j=1}^M Chk(x_j^a, x_j^b) \quad (28)$$

where

$$Chk(x_j^a, x_j^b) = \begin{cases} 0, & \text{if } x_j^a = x_j^b \\ 1, & \text{else} \end{cases} \quad (29)$$

From (29), we can see that, for each task (dimension) of two solutions (locations), if the satellite which will perform the task is the identical, the distance of this dimension is 0; else the distance of this dimension is 1. And the distance between two solutions is the sum of the distance between each dimension of the two solutions.

And then the selection probability of a solution (location)  $X_i$  is defined as follows:

$$p(X_i) = \frac{R(X_i)}{\sum_{j=1}^M R(X_j)} \quad (30)$$

where

$$R(X_i) = \sum_{j=1}^M Dis(X_i, X_j) \quad (31)$$

From (30), we can see that if there is more difference between  $X_i$  and other solutions, the higher probability  $X_i$  will be selected to the explosion group of next generation.

TABLE 1: Illustration of experiment scenarios for STFBHB parameter analyzing.

ID	Target	Task	Satellite
sc1	10	20	15
sc2	20	40	20
sc3	30	60	30
sc4	40	80	40

5.3. *Constraint Handling in STFFA.* In the process of sparks generation, some invalid solution may appear. For example, one satellite cannot perform a certain task, because constraint C4 of the satellite is violated. So we need to handle the constraint C3 and C4. When an explosion group of new generation is generated, we adopt solution repair method to transform invalid solutions to valid solutions. That is if a solution violated some constraints, we randomly select another satellite from candidate satellite set to perform the task, until all constraints are satisfied. If all candidate satellites cannot perform the task (this situation seldom happens), then the solution (location) will be discarded, and an alternative new solution will be generated again by explosion spark.

## 6. Experimental Results and Analysis

In order to validate the effectiveness of the three algorithms mentioned above, we selected 50 satellites running various orbits from the ephemeris database of STK and 100 metropolises as targets to be observed to simulate the real agent team formation problem for observation task-oriented satellite cluster. We randomly generate 10 scenarios with various target number, task number, and satellite number, notated as sc1 to sc10.

We suppose that the observation payload set for all satellites includes five types of sensors (visible light, infrared, and SAR (Synthetic Aperture Radar) sensors with different spatial resolution). For each satellite we select one to five sensors to be equipped on it randomly. For each experiment scenario, we randomly set 0-10% access windows to be occupied for each satellite in order to simulate the situation that some satellites have undertaken some observation tasks before current agent team formation. The  $CofCost_{Sat}^k$ ,  $\beta^k$ , and  $\theta^k$  for each satellite and  $Cost_{Task}^j$  for each task are generated randomly.

The scheduling period is 24 hours. Computation platform is configured as follows: Inter Core-i5 2.70 GHz CPU + 8 GB RAM + Win7, programmed with Microsoft Visual Studio 2010 C#.

6.1. *Algorithm Parameters Analyzing in STFBHB.*  $\varepsilon$  and  $\omega$  are key parameters of STFBHB. In this section, their influence to the performance of STFBHB is analyzed.

We generated four different scenarios with different problem size. The details of the four scenarios are shown in Table 1.

And then we applied STFBHB with different  $\varepsilon$  and  $\omega$  on the experiment scenarios sc1 to sc4. The computing results

are shown in Figures 6, 7, 8, and 9. Subfigure (a) and (b) of Figures 6, 7, 8, and 9 compare the performance (team cost) of STFBHB with various  $\varepsilon$ ,  $\omega$  on scenarios sc1 to sc4, respectively. And subfigures (c) and (d) of Figures 6, 7, 8, and 9 compare the running time of STFBHB with different  $\varepsilon$ ,  $\omega$  on scenarios sc1 to sc4, respectively.

As indicated in Figures 6, 7, 8, and 9, the running time of STFBHB is decreased with the increase of  $\omega$  and  $\varepsilon$  obviously. And the value of the two parameters has less influence on the computing results of the team cost than that of the algorithm running time. For example, In Figure 8(c), the running time of STFBHB ( $\varepsilon = 0.3$ ,  $\omega = 0.02$ ) is only 27% of that of STFBHB ( $\varepsilon = 0.1$ ,  $\omega = 0.01$ ) and in Figure 8(a), the performance (team cost) of STFBHB ( $\varepsilon = 0.3$ ,  $\omega = 0.02$ ) only drops 0.1% compared with that of STFBHB ( $\varepsilon = 0.1$ ,  $\omega = 0.01$ ).

From the subfigures (c) and (d) of Figures 6, 7, 8, and 9, we can see that the running time of STFBHB drops sharply when the value of  $\omega$  is between 0.01 and 0.03 and the value of  $\varepsilon$  is between 0.3 and 0.9.

From the subfigure (a) of Figures 6, 7, 8, and 9, we can also see that the mean of the team cost with large  $\omega$  is larger than that with small  $\omega$ . And the team cost is low when the value of  $\omega$  is between 0.01 and 0.03. From the subfigure (b) of Figures 6, 7, 8, and 9, we can also see that the variance of team cost with high  $\varepsilon$  is larger than that with low  $\varepsilon$ . So in other words, the performance stability of STFBHB is decreasing with the increase of  $\varepsilon$ . The team cost is relatively stable when the value of  $\varepsilon$  is between 0.1 and 0.3.

Considering the compromise of performance and consuming time, we choose parameters  $\varepsilon = 0.3$  and  $\omega = 0.03$  for STFBHB in the later experiments for comparing with other algorithms.

Figure 10 shows the nodes number in each layer of the searching tree of STFBHB with three different group of  $\varepsilon$  and  $\omega$  in experiment scenarios sc4.

As is illustrated in Figure 10, the number of nodes which will be extended in each layer decreases as the parameters  $\omega$  and  $\varepsilon$  increase. This is because when the values of  $\omega$  and  $\varepsilon$  are large, there are fewer nodes need to be extended and searched, and then the running time of STFBHB cut down.

6.2. *Performance Experiment for The Algorithms.* In order to verify the effectiveness and efficiency of the proposed algorithm in this paper, we conduct some experiments for small, medium, and large-scale problem, respectively. And original Fireworks Algorithm (FWA) is also used as the baseline for comparison in our experiment.

In order to conquer the randomness of FWA and STFFA, each algorithm has been run for 50 times in each scenario, and we record the average results (team cost, running time, and iteration times). For FWA and STFFA, the algorithms exit when no improvements appear during 100 successive iterations.

The parameters of FWA and STFFA are shown in Table 2.  $GroupNum$  is the group number of the algorithms.  $lm$  is the lower bound of explosion spark number and  $um$  is the upper bound of explosion spark number. If the number of the sparks generated by algorithms in each generation is less than  $lm$  or more than  $um$ , it will be set to  $lm$  or

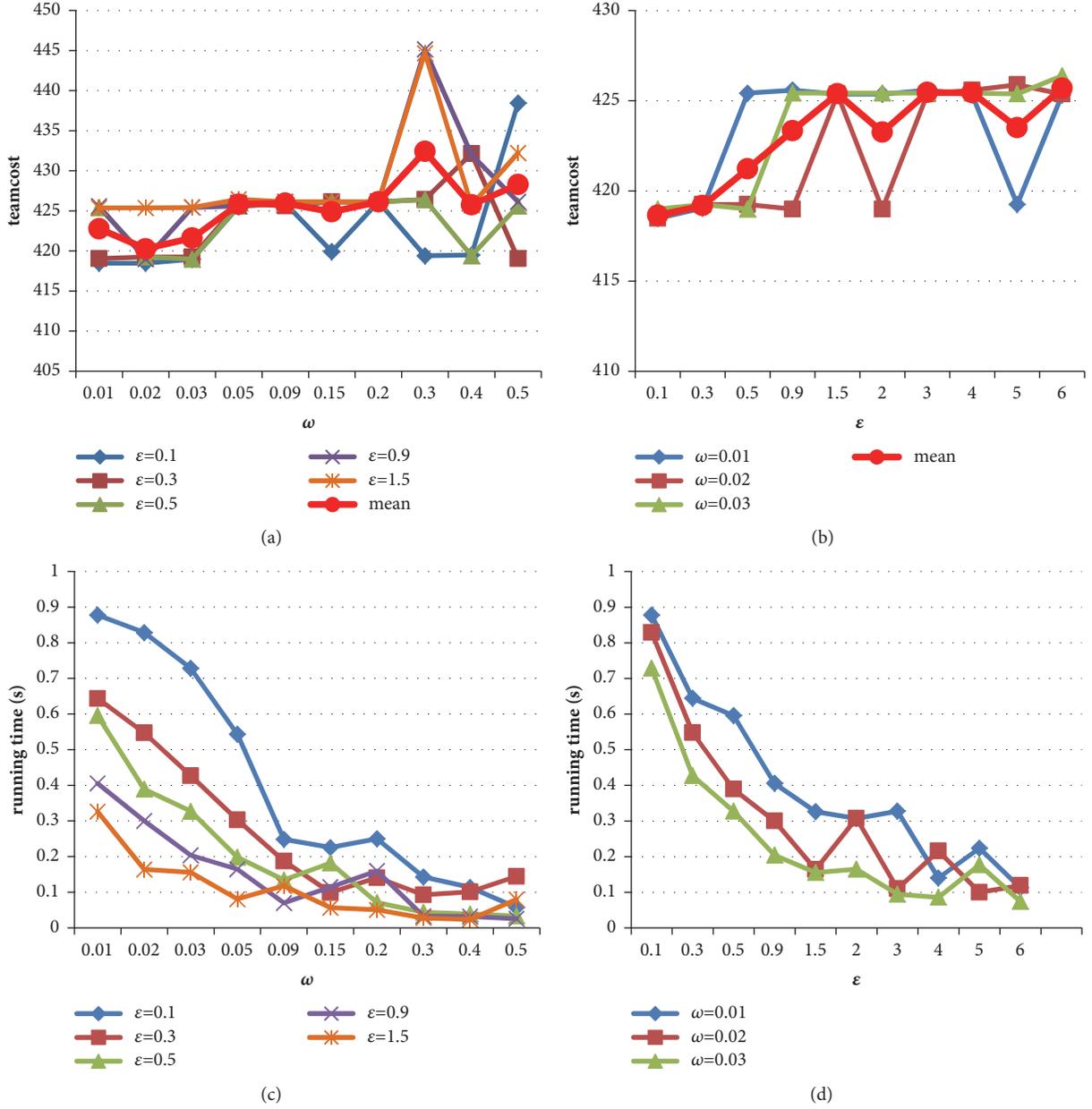


FIGURE 6: The computing results of STFBHB with different value of algorithm parameters on scl.

TABLE 2: The parameters of FWA and STFFA.

	FWA	STFFA
<i>GroupNum</i>	50	50
<i>lm</i>	1	1
<i>um</i>	200	200
<i>GFNum</i>	50	50
<i>a</i>	0.4	0.4
<i>g</i>	0.4	0.2
<i>h</i>	not exist	0.2

*um*, *GFNum* is the number of sparks produced by Gauss operator in each generation. *a* is a constant parameter of

FWA and STFFA in the explosion amplitude computation formulation. *g* is a constant parameter of FWA and STFFA in the explosion spark number computation formulation which is related to the objective function value. And *h* is a unique constant parameter of STFFA. It is in the explosion spark number computation formulation which is related to the load balancing degree of the whole satellite cluster.

As mentioned in Section 6.1, the parameters of STFBHB are set as  $\epsilon = 0.3$  and  $\omega = 0.03$ .

The range of the running time of the algorithms are large. In order to illustrate them in one figure, logarithmic coordinate system is adopted in our experiment.

We employ the following equation to normalize the team cost of the algorithms in each scenario:

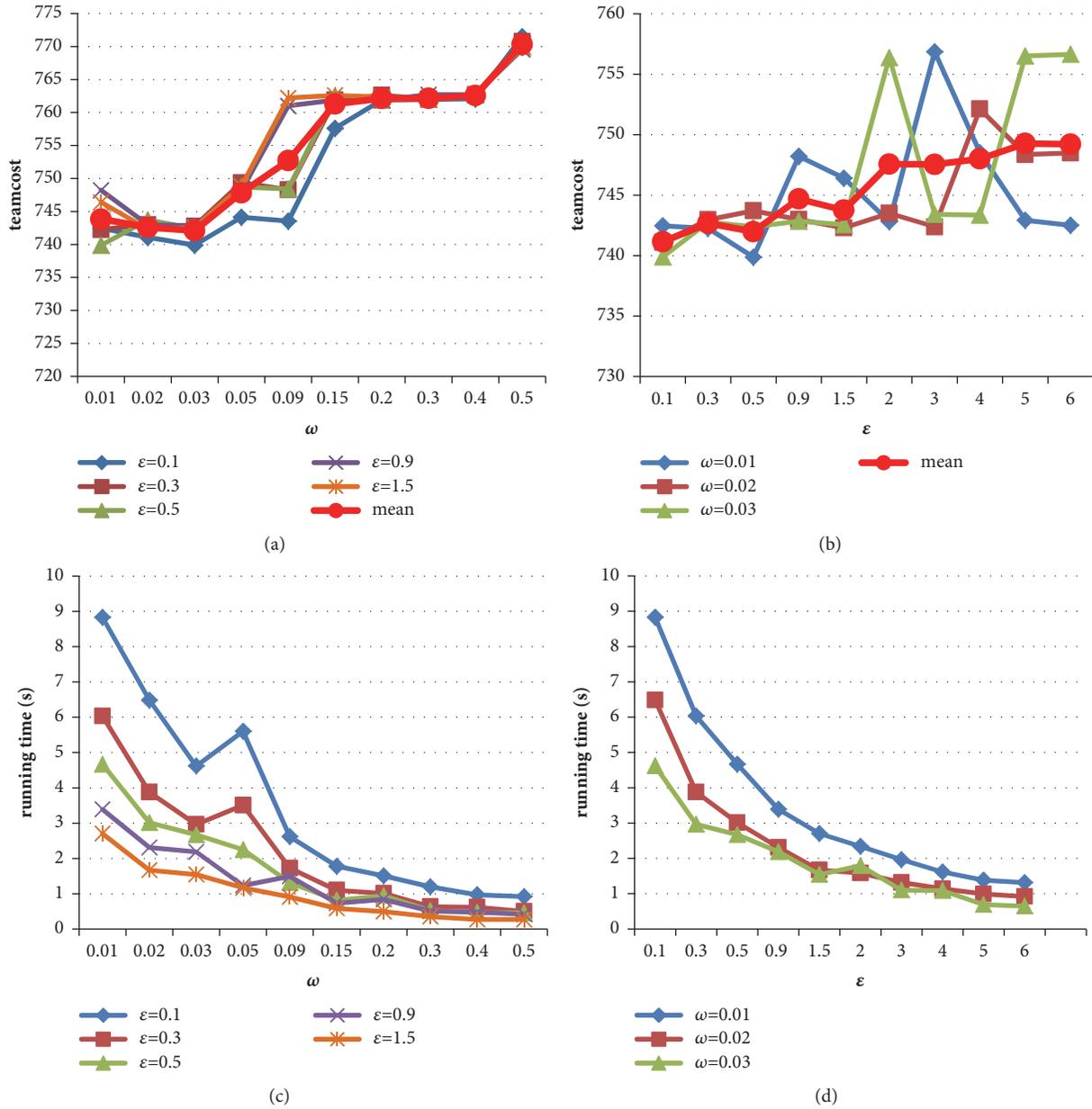


FIGURE 7: The computing results of STFBHB with different value of algorithm parameters on sc2.

$$\text{ratio} = \frac{\text{teamcost of each algorithm}}{\text{teamcost of STFBHB}} \quad (32)$$

We leverage our algorithms on small-scale, medium-scale, and large-scale scenarios to see the performance. If satellite number in a scenario is equal to or less than 10, it is small-scale. If the satellite number is more than 10 and equal to or less than 20, it is medium-scale. If a scenario involves more than 20 satellites, it is a large-scale scenario.

6.2.1. *Experiment of Small-Scale Scenarios.* Table 3 and Figures 14 and 15 compare the computation results of the four algorithms in scenario sc5, in which the satellite number is 5

and the target number is 10. *TaskNum* in Table 3 is the task number in sc5.

As illustrated in Table 3 and Figure 11, when the problem scale is small, all algorithms find the optimal solutions. From Figure 12 we can see that FWA and STFFA based on swarm intelligence take more time to do computation than other two algorithms. The running time of FWA and STFFA is even more than that of STCFDB which is a complete algorithm. The running time of FWA and STFFA is very close to each other on scenario sc5 with different task number.

Then we generate another scenario sc6 which is a little bigger than sc5. There are 10 satellites and 10 targets in sc6. Table 4 and Figures 13 and 14 compare the computation results of the four algorithms in scenario sc6.

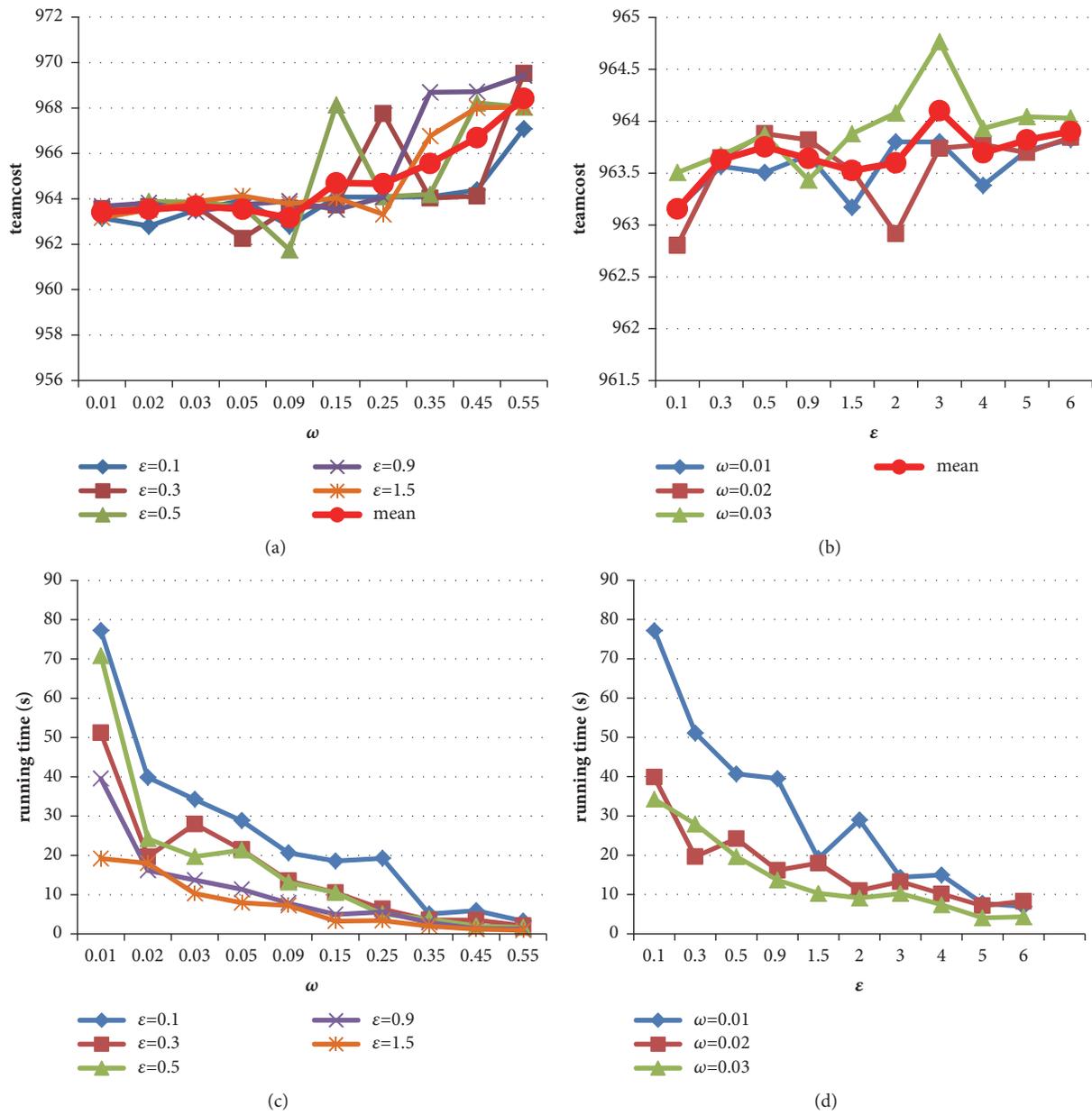


FIGURE 8: The computing results of STFBHB with different value of algorithm parameters on sc3.

TABLE 3: The team cost of four algorithms on sc5.

TaskNum	STCFDB	STFBHB	FWA	STFFA
10	310.11	310.11	310.11	310.11
12	405.73	405.73	405.73	405.73
14	509.75	509.75	509.75	509.75
16	622.01	622.01	622.01	622.01
18	696.55	696.55	696.55	696.55
20	777.30	777.30	777.30	777.30

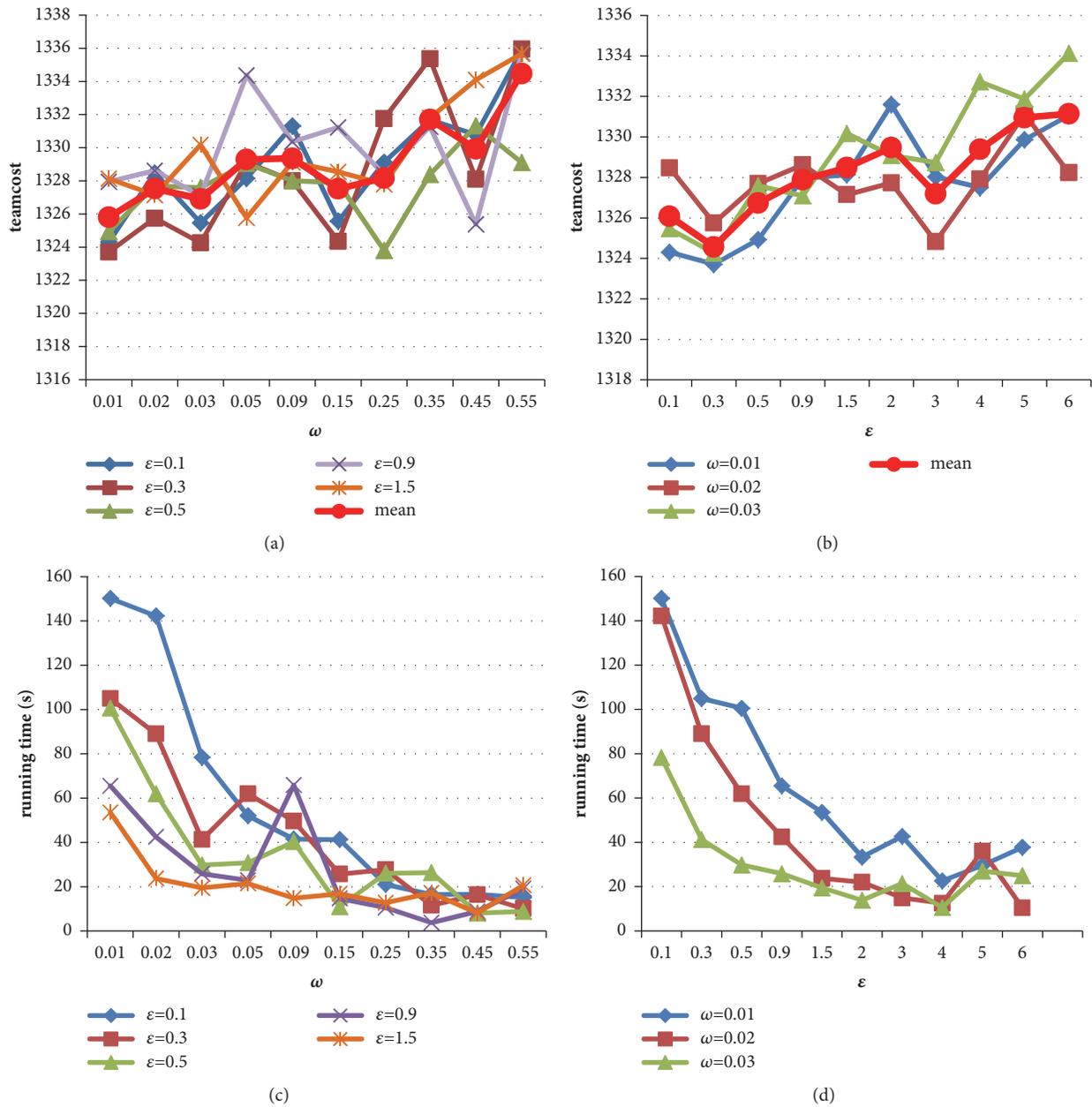


FIGURE 9: The computing results of STFBHB with different value of algorithm parameters on sc4.

TABLE 4: The team cost of four algorithms on sc6.

TaskNum	STCFDB	STFBHB	FWA	STFFA
10	208.60	208.60	209.33	209.33
12	272.33	272.33	277.58	277.58
14	355.74	355.74	355.74	355.74
16	439.57	439.57	439.70	439.72
18	497.55	497.55	503.89	503.94
20	563.08	563.08	564.89	564.89

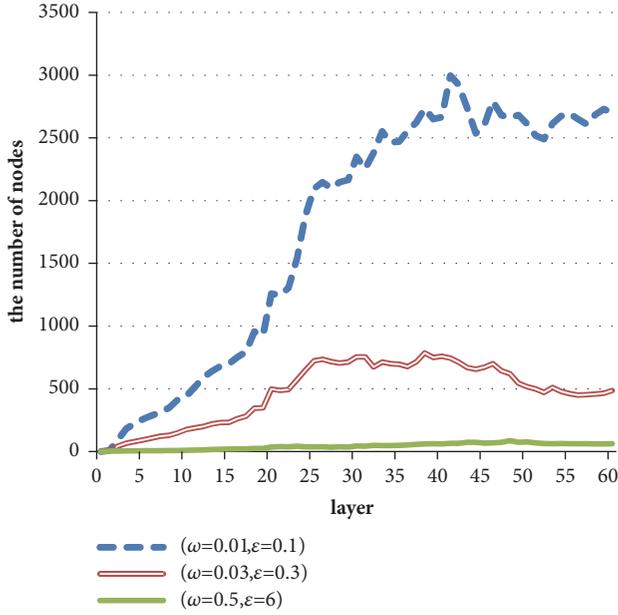


FIGURE 10: Nodes number in each layer of searching tree of STFBHB with different  $\omega$  and  $\epsilon$  on scenarios sc4.

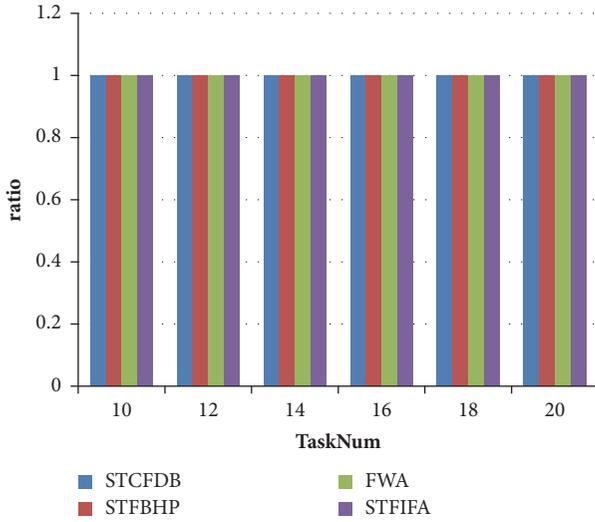


FIGURE 11: The comparison of algorithms' computing results (ratio) on sc5.

In Table 4, the bold font indicates that the algorithm finds the optimal solution on scenario sc6. From Table 4 and Figure 13, we can know that, besides STCFDB which is an exact algorithm, STFBHP finds optimal solutions in all cases. In small-scale scenarios, the branches of the searching tree are not too many, so the heuristic pruning strategy in STFBHP does not miss the optimal solutions. The performance of FWA and STFFA is not very good. They are not suitable for small-scale scenarios.

Figure 14 shows the running time of all algorithms. Because STFBHP is a polynomial time algorithm, the running time of which is the shortest among all the algorithms. The running time of STCFDB is even longer than that of

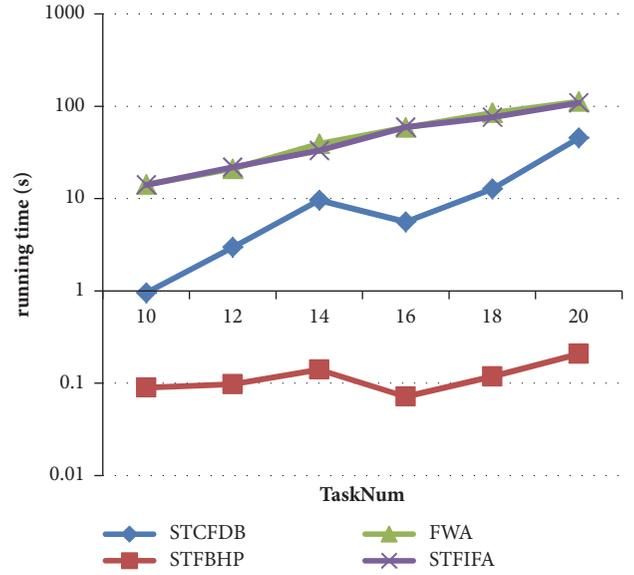


FIGURE 12: The running time of different algorithms on sc5.

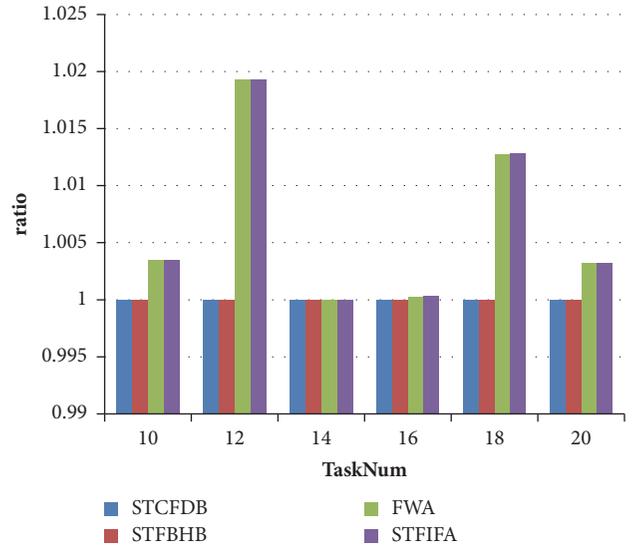


FIGURE 13: The comparison of algorithms' computing results (ratio) on sc6.

STFFA and FWA when task number is beyond 12. Due to the exponential time complexity of STCFDB, when task number is 20 and satellite number is 10, its running time exceeds 2.8 hours.

6.2.2. *Experiment of Medium-Scale Scenarios.* For further verification of our algorithms, we conduct some medium-scale experiments. We generate two groups of scenarios sc7 and sc8, which are shown in Table 5. The computing results are shown in Table 6 and Figures 15, 17, 16, and 18.

STCFDB cannot figure out the result of sc7-1 in 8 hours! The computing time of STCFDB is too long to make sense in real-world applications, so we only compare the performance of STFBHP, FWA, and STFFA in the later experiments.

TABLE 5: Illustration of medium scale experiment scenarios.

ID	Target	Task	Satellite
sc7-1	10	20	15
sc7-2	10	30	15
sc7-3	20	40	15
sc7-4	20	50	15
sc7-5	30	60	15
sc7-6	30	70	15
sc7-7	40	80	15
sc8-1	10	20	20
sc8-2	10	30	20
sc8-3	20	40	20
sc8-4	20	50	20
sc8-5	30	60	20
sc8-6	30	70	20
sc8-7	40	80	20

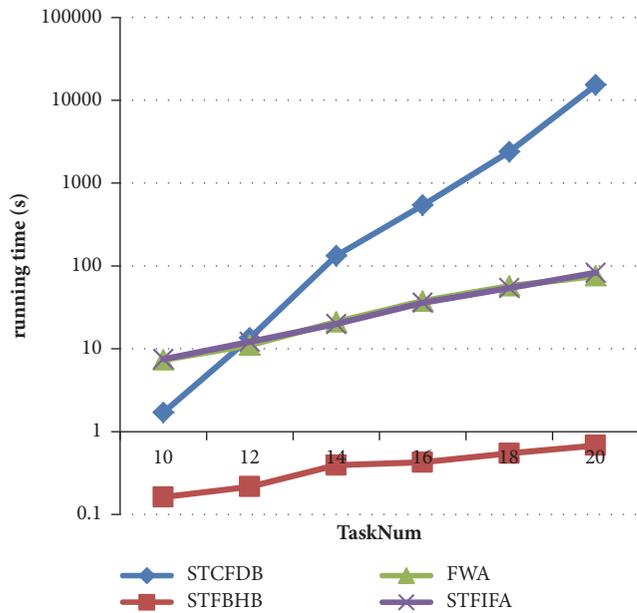


FIGURE 14: The running time of different algorithms on sc6.

In Table 6, the bold font indicates that the algorithm obtains the best solution among all three algorithms on scenarios sc7 and sc8. As illustrated in Table 6 and Figures 15 and 16, STFBHB performs well on sc7, when *TaskNum* is small, while STFFFA and FWA get better results when *TaskNum* is large on sc7 (sc7-5 to sc7-7) and on sc8 (satellite number of sc8 is 20, which is larger than that of sc7). This is because when task number is small or satellite number is small, there are fewer branches in the searching tree, and the heuristic pruning strategy of STFBHB is more efficient. But as the task number increases, the global search ability of FWA and STFFFA manifests and gets better results. Compared with FWA, the performance of STFFFA is better. This is because STFFFA has unique sparks number computing

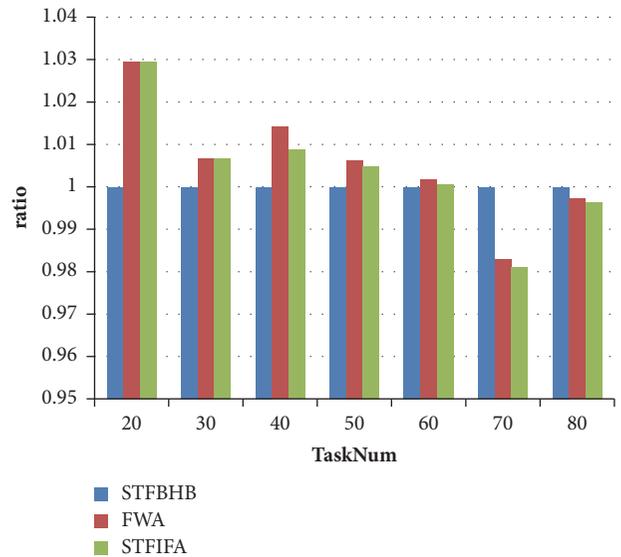


FIGURE 15: The comparison of algorithms' computing results (ratio) on sc7.

method ( $F(X_i)$ ) which considers the load balance of satellite cluster and gets better results.

From Figures 17 and 18, we can see that it only takes tens of seconds for STFBHB to obtain results on sc7 and sc8, while the running time of STFFFA is around 1000 times of that of STFBHB.

6.2.3. *Experiment of Large-Scale Scenarios.* We also generate two large-scale scenarios sc9 and sc10 for further experiment, which contains 30 satellites and 40 satellites, respectively. The details are shown in Table 7.

The computing results are shown in Table 8 and Figures 19, 20, 21 and 22.

In Table 8, the bold font indicates that the algorithm obtains the best solution among all three algorithms on

TABLE 6: The team cost of different algorithms on medium scale experiment scenarios.

ID	STFBHB	FWA	STFFA
sc7-1	<b>419.25</b>	431.63	431.63
sc7-2	<b>706.48</b>	711.33	711.22
sc7-3	<b>1092.72</b>	1111.17	1102.31
sc7-4	<b>1437.85</b>	1445.93	1445.78
sc7-5	1905.36	<b>1904.89</b>	1904.96
sc7-6	2293.34	2256.66	<b>2249.97</b>
sc7-7	2762.59	2753.53	<b>2751.83</b>
sc8-1	301.08	<b>298.44</b>	298.59
sc8-2	482.05	479.20	<b>478.86</b>
sc8-3	742.78	724.85	<b>722.46</b>
sc8-4	972.03	956.51	<b>954.79</b>
sc8-5	1300.02	1272.53	<b>1271.03</b>
sc8-6	1512.56	1490.96	<b>1486.74</b>
sc8-7	1815.65	1785.50	<b>1780.23</b>

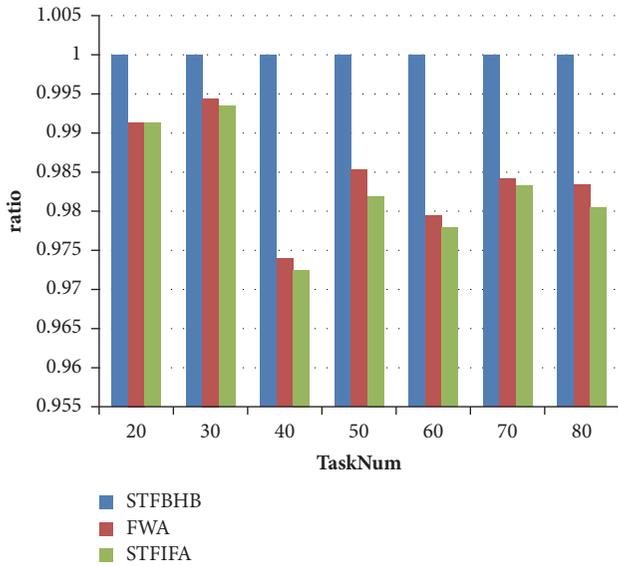


FIGURE 16: The comparison of algorithms' computing results (ratio) on sc8.

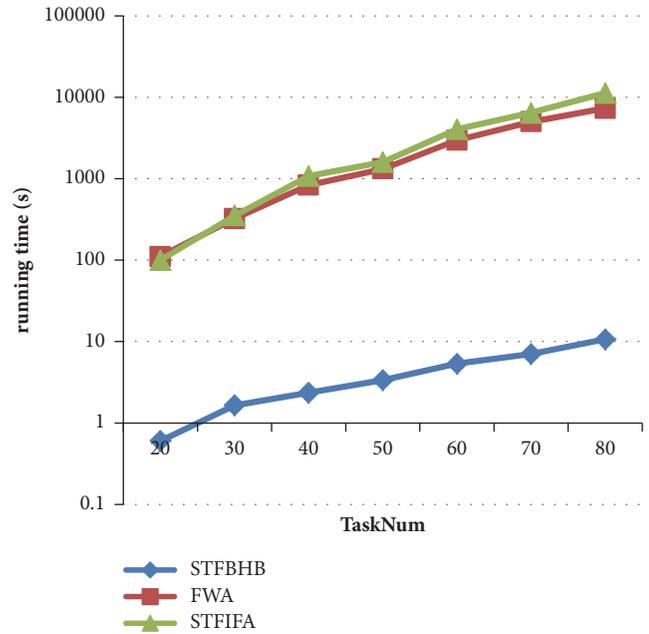


FIGURE 17: The running time of different algorithms on sc7.

scenarios sc9 and sc10. As shown in Figures 19 and 20, STFFA obtains better results than FWA and STFBHB on more subscenarios of sc9 and sc10. When satellite number reaches 30, there are so many nodes in each layer of searching tree in STFBHB. The heuristic strategy of STFBHB cannot prune branches in a global perspective. So STFFA and FWA which are based on swarm intelligence get better results. STFFA with unique sparks number computing method ( $F(X_i)$ ) which takes the load balance of satellite cluster into consider also obtains better results than FWA on more large-scale subscenarios. With the help of  $F(X_i)$ , the exploitation capacity of STFFA is stronger than that of FWA.

From Figures 21 and 22, we can know that the advantage of running time of STFBHB is still very obvious in large-scale scenarios. So STTFA is suitable for the situations that the

requirement of running time is lenient and the requirement of solution optimization degree is high.

### 6.3. Convergence Speed Comparison for FWA and STFFA.

In order to analyze the convergence of STFFA, we employ STFFA and FWA on medium-scale (sc7, sc8) and large-scale (sc9, sc10) scenarios. Each algorithm runs 50 times, respectively. In each time, the final iteration times of FWA are recorded. If STFFA finds the solution with the same team cost of the solution found by FWA, then the iteration times are recorded. If the solution of STFFA is inferior to the solution of FWA, the final iteration times of STFFA are recorded.

The mean iteration times of FWA and STFFA are compared in Figures 23, 24, 25, and 26.

TABLE 7: Illustration of large scale experiment scenarios.

ID	Target	Task	Satellite
sc9-1	10	20	30
sc9-2	10	30	30
sc9-3	20	40	30
sc9-4	20	50	30
sc9-5	30	60	30
sc9-6	30	70	30
sc9-7	40	80	30
sc10-1	10	20	40
sc10-2	10	30	40
sc10-3	20	40	40
sc10-4	20	50	40
sc10-5	30	60	40
sc10-6	30	70	40
sc10-7	40	80	40

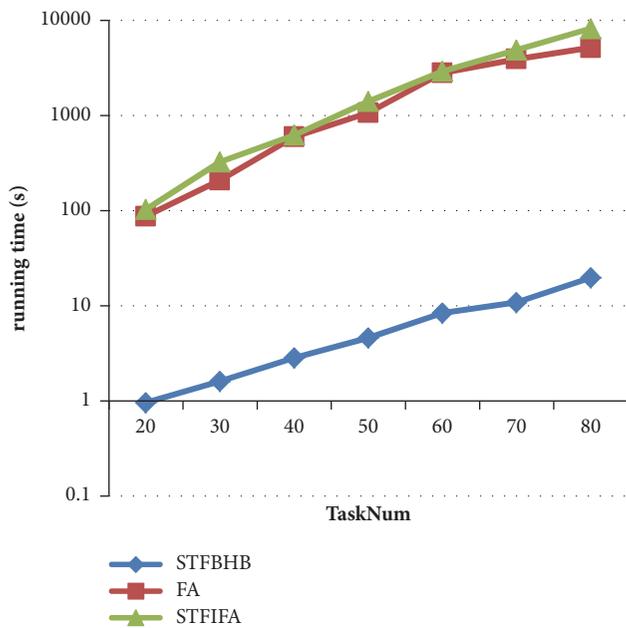


FIGURE 18: The running time of different algorithms on sc8.

As indicated in Figures 23, 24, 25, and 26, the iteration times of STFFA are less than FWA in most subscenarios of sc7, sc8, sc9, and sc10. So we can conclude that the convergence speed of STFFA is faster than that of FWA by and large. This is because  $F(X_i)$  which is related to the load balance degree of satellite cluster makes STFFA do search in more promising region of solution space and gets better results quickly.

### 7. Conclusion

Agent team formation for observation task-oriented satellite cluster is a new problem to be solved in the field of aerospace resource scheduling. It is different from the

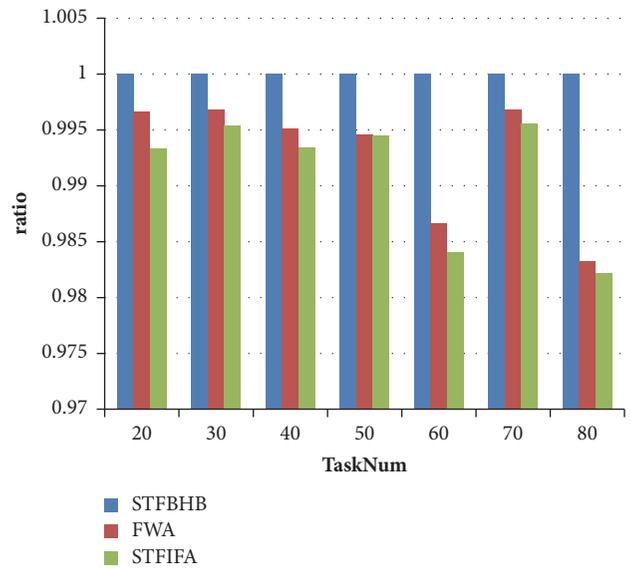


FIGURE 19: The comparison of algorithms' computing results (ratio) on sc9.

traditional satellite observation scheduling in scheduling aim and problem scale.

Through the characteristics analysis of the observation process of large-scale satellite cluster, we establish a constraint satisfaction optimization model based on agent and multiagent theory and then propose three novel algorithms including STCFDB, STFBHP, and STFFA. They are complete search algorithm, heuristic search algorithm, and swarm intelligence optimization algorithm, respectively. From the experiment results, we can make a conclusion that every algorithm we propose has its applicable scenarios.

(1) STCFDB is a complete searching algorithm, which guarantees that if a path to the goal exists, the algorithm will reach the goal. But the exponential time complexity of STCFDB makes it can only solve small-scale problem.

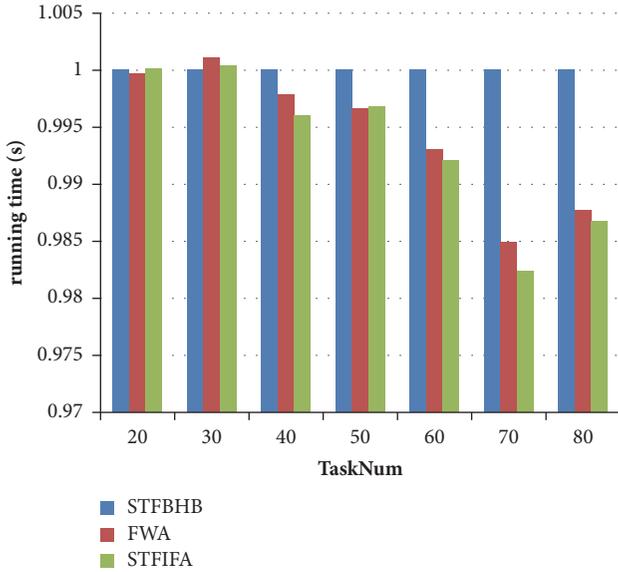


FIGURE 20: The comparison of algorithms' computing results (ratio) on sc10.

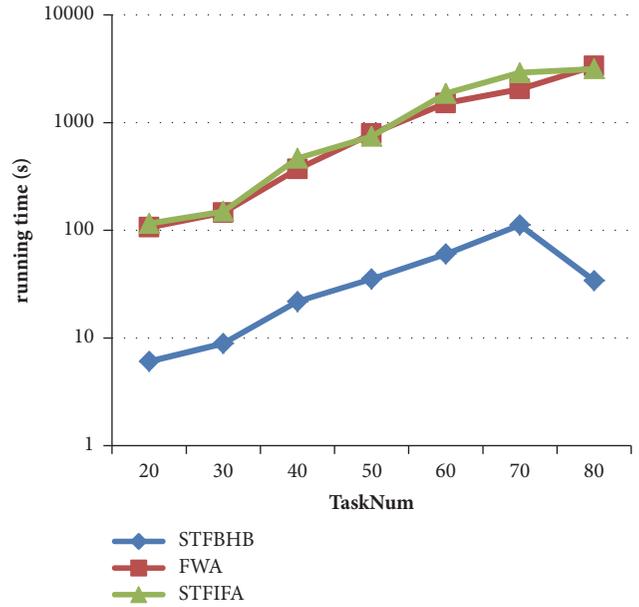


FIGURE 22: The running time of different algorithms on sc10.

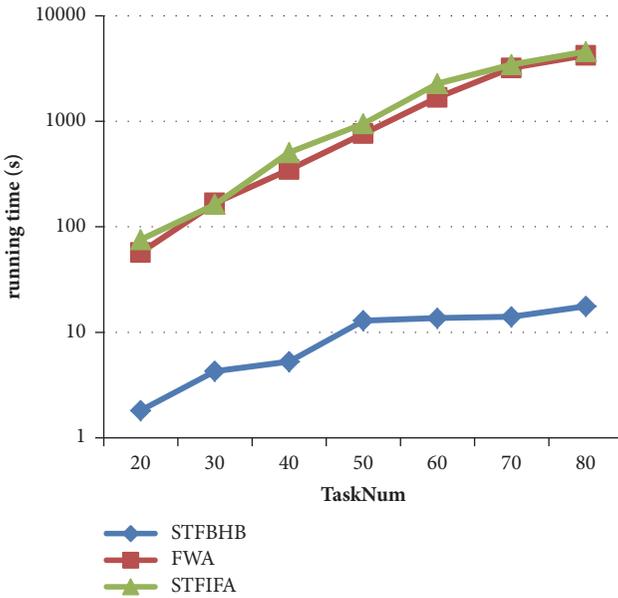


FIGURE 21: The running time of different algorithms on sc9.

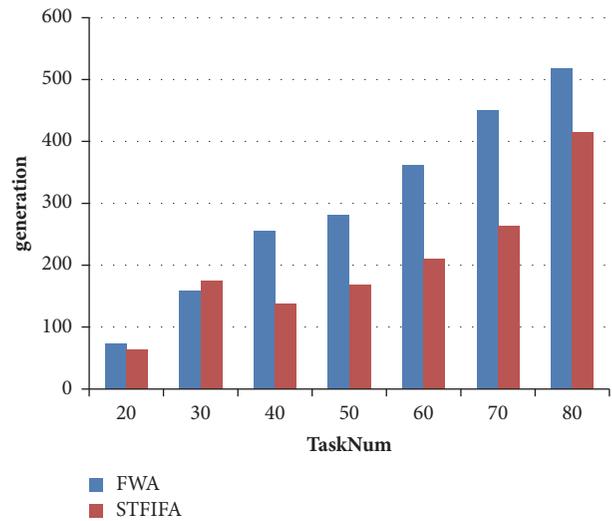


FIGURE 23: Comparison of convergence speed between FWA and STFIFA on sc7.

(2) STFBHB is a heuristic searching algorithm with pruning strategy. It can give high quality solutions when problem scale is small or medium. STFBHB is a polynomial time complexity algorithm and the running time of it is the shortest of the three algorithms.

(3) STFFA which is based on swarm intelligence is suitable for medium and large-scale problems. STFFA always obtains better results compared to the other algorithms when satellite number and task number are both large. The running time of STFFA is also long (around 10000 seconds in some large-scale scenarios).

Our future work will consider designing some heuristic operators referring to domain knowledge and history scheduling data. The heuristic operators can combine with our algorithm and enhance the effectiveness and efficiency of the searching process. Another further work is analyzing the performance bounds of STFBHP theoretically.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

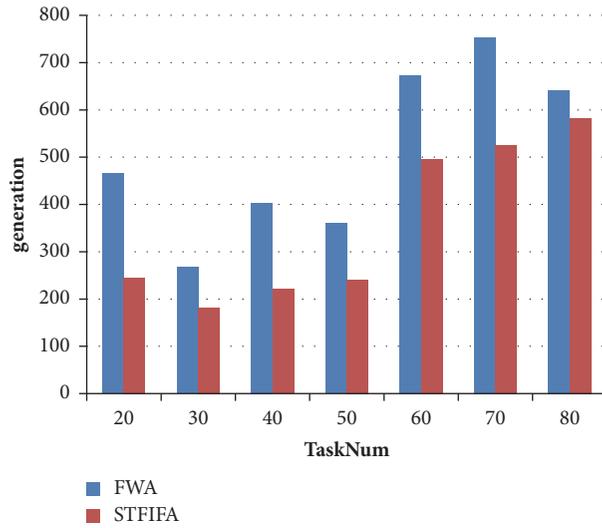


FIGURE 24: Comparison of convergence speed between FWA and STFIFA on sc8.

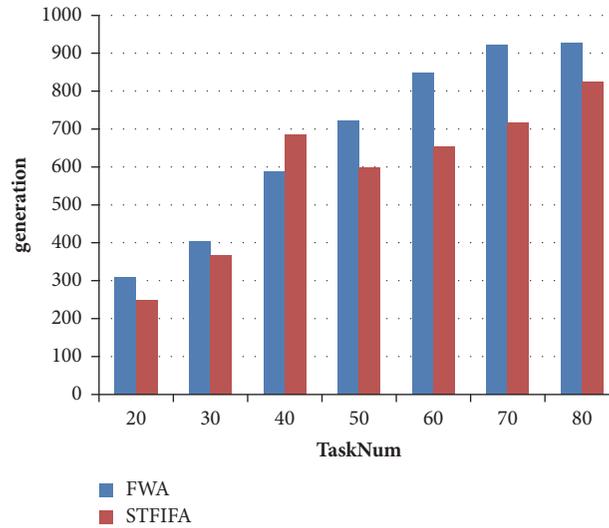


FIGURE 25: Comparison of convergence speed between FWA and STFIFA on sc9.

TABLE 8: The team cost of different algorithms on large-scale experiment scenarios.

ID	STFBHB	FWA	STFFA
sc9-1	250.27	249.28	<b>248.63</b>
sc9-2	415.11	414.12	<b>413.17</b>
sc9-3	589.04	586.19	<b>584.26</b>
sc9-4	768.30	764.03	<b>763.91</b>
sc9-5	1001.05	987.09	<b>983.34</b>
sc9-6	1188.92	1166.04	<b>1165.85</b>
sc9-7	1406.99	1382.99	<b>1380.57</b>
sc10-1	243.3900	243.3728	<b>243.1981</b>
sc10-2	<b>400.4246</b>	401.2105	400.4555
sc10-3	568.8278	567.2267	<b>566.8236</b>
sc10-4	740.8387	<b>738.2841</b>	739.6536
sc10-5	959.0193	953.9334	<b>949.9732</b>
sc10-6	1145.2050	1128.0140	<b>1125.2667</b>
sc10-7	1346.8868	1329.3575	<b>1327.1223</b>

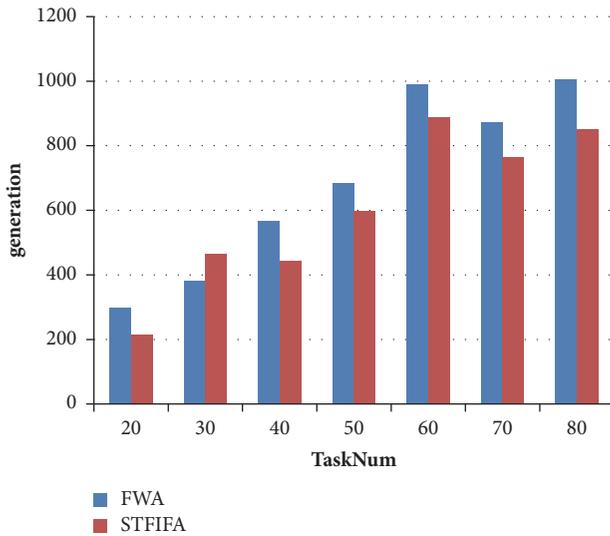


FIGURE 26: Comparison of convergence speed between FWA and STFIFA on scl0.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

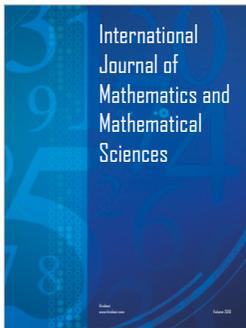
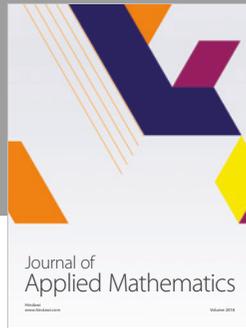
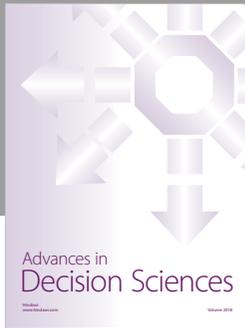
## Acknowledgments

This work was supported by the National Natural Science Foundation of China (61101184 and 61174159).

## References

- [1] H. Chen, Y. Zhou, C. Du, and J. Li, "A satellite cluster data transmission scheduling method based on genetic algorithm with rote learning operator," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 5076–5083, Canada, July 2016.
- [2] C. Wang, H. Chen, B. Zhai, J. Li, and L. Chen, "Satellite observing mission scheduling method based on case-based learning and a genetic algorithm," in *Proceedings of the 28th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016*, pp. 627–634, USA, November 2016.
- [3] X. Liu, G. Laporte, Y. Chen, and R. He, "An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time," *Computers & Operations Research*, vol. 86, pp. 41–53, 2017.
- [4] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers & Operations Research*, vol. 57, pp. 1–16, 2015.
- [5] B. Zhai, H. Chen, J. Li, and N. Jing, "A coevolution genetic algorithm for conflict resolution of multi-user satellite observation activities," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference, GECCO 2016 Companion*, pp. 59–60, USA, July 2016.
- [6] S. Lee, W. C. Jung, and J.-H. Kim, "Task scheduling algorithm for the communication, ocean, and meteorological satellite," *ETRI Journal*, vol. 30, no. 1, pp. 1–12, 2008.
- [7] H. Chen, J. Li, N. Jing, and J. Li, "User-oriented data acquisition chain task planning algorithm for operationally responsive space satellite," *Journal of Systems Engineering and Electronics*, vol. 27, no. 5, pp. 1028–1039, 2016.
- [8] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A Lagrangian heuristic for satellite range scheduling with resource constraints," *Computers & Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.
- [9] P. Tangpattanakul, N. Jozefowicz, and P. Lopez, "A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite," *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015.
- [10] Shuang Peng, Hao Chen, Jun Li, and Ning Jing, "Approximate Path Searching Method for Single-Satellite Observation and Transmission Task Planning Problem," *Mathematical Problems in Engineering*, vol. 2017, Article ID 7304506, 16 pages, 2017.
- [11] F. Xhafa, X. Herrero, A. Barolli, L. Barolli, and M. Takizawa, "Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem," *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1086–1100, 2013.
- [12] Y. Li, R. Wang, and M. Xu, "Rescheduling of observing spacecraft using fuzzy neural network and ant colony algorithm," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 678–687, 2014.
- [13] H. Chen, J. Wu, W. Shi, J. Li, and Z. Zhong, "Coordinate scheduling approach for EDS observation tasks and data transmission jobs," *Journal of Systems Engineering and Electronics*, vol. 27, no. 4, Article ID 7669701, pp. 822–835, 2016.
- [14] P. Shuang, C. Hao, L. Jun, J. Ning, and W. Jiangjiang, "Multi-agent collaborative planning method of emergency mission based on scheme fusion strategy," in *Proceedings of the 2014 Symposium on Computer Applications and Communications, SCAC 2014*, pp. 87–92, China, July 2014.
- [15] C. Wang, J. Li, N. Jing, J. Wang, and H. Chen, "A distributed cooperative dynamic task planning algorithm for multiple satellites based on multi-agent hybrid learning," *Chinese Journal of Aeronautics*, vol. 24, no. 4, pp. 493–505, 2011.
- [16] . Peng Feng, . Hao Chen, . Shuang Peng, . Luo Chen, and . Longmei Li, "A method of distributed multi-satellite mission scheduling based on improved contract net protocol," in *Proceedings of the The 2015 11th International Conference on Natural Computation*, pp. 1062–1068, Zhangjiajie, China, August 2015.
- [17] L. Barbulescu, "Leap Before You Look: An Effective Strategy in an Oversubscribed Scheduling Problem[C]," in *Proceedings of the In the proceeding of 19th National Conference on Artificial Intelligence (AAAI-04)*, San Jose, California, USA, 2004.
- [18] C. K. Pang, A. Kumar, C. H. Goh, and C. V. Le, "Nano-satellite swarm for SAR applications: Design and robust scheduling," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 853–865, 2015.
- [19] J. E. Moores, K. A. Carroll, I. DeSouza et al., "The small reconnaissance of atmospheres mission platform concept, part 2: design of carrier spacecraft and atmospheric entry probes," *International Journal of Space Science and Engineering*, vol. 2, no. 4, p. 345, 2014.
- [20] W. Truskowski, H. Hallock, and C. Rouff, *Autonomous and autonomic systems: with applications to NASA intelligent spacecraft operations and exploration systems[M]*, Springer Science Business Media, 2009.
- [21] E. Dekens, S. Engelen, and R. Noomen, "A satellite swarm for radio astronomy," *Acta Astronautica*, vol. 102, pp. 321–331, 2014.
- [22] G. P. Asner, R. E. Martin, and J. Mascaro, "Coral reef atoll assessment in the South China Sea using Planet Dove satellites,"

- Remote Sensing in Ecology and Conservation*, vol. 3, no. 2, pp. 57–65, 2017.
- [23] 2016, China's aerospace in 2016, White Paper, <http://www.fmprc.gov.cn/ce/cgvienna/chn/zgbd/t1427271.htm>.
- [24] H. Chen, C. Du, J. Li, N. Jing, and L. Wang, "An approach of satellite periodic continuous observation task scheduling based on evolutionary computation," in *Proceedings of the the Genetic and Evolutionary Computation Conference Companion*, pp. 15–16, Berlin, Germany, July 2017.
- [25] E. Andrejczuk, J. A. Rodriguez-Aguilar, and C. Sierra, "A Concise Review on Multiagent Teams: Contributions and Research Opportunities," in *Multi-Agent Systems and Agreement Technologies*, vol. 10207 of *Lecture Notes in Computer Science*, pp. 31–39, Springer International Publishing, Cham, 2017.
- [26] T. Okimoto, N. Schwind, M. Clement, T. Ribeiro, K. Inoue, and P. Marquis, "How to form a task-oriented robust team," in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pp. 395–403, Turkey, May 2015.
- [27] T. Okimoto, T. Ribeiro, D. Bouchabou, and K. Inoue, "Mission oriented robust multi-team formation and its application to robot rescue simulation," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 454–460, USA, July 2016.
- [28] C. Crawford, Z. Rahaman, and S. Sen, "Evaluating the Efficiency of Robust Team Formation Algorithms," in *Autonomous Agents and Multiagent Systems*, vol. 10002 of *Lecture Notes in Computer Science*, pp. 14–29, Springer International Publishing, Cham, 2016.
- [29] L. S. Marcolino, A. X. Jiang, and M. Tambe, "Multi-agent team formation: Diversity beats strength?" in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*, pp. 279–285, China, August 2013.
- [30] S. Liemhetcharat and M. Veloso, "Forming an effective multi-robot team robust to failures," in *Proceedings of the 2013 26th IEEE/RSJ International Conference on Intelligent Robots and Systems: New Horizon, IROS 2013*, pp. 5240–5245, Japan, November 2013.
- [31] M. Farhangian, M. K. Purvis, M. Purvis, and B. T. Savarimuthu, "Modeling the Effects of Personality on Team Formation in Self-assembly Teams," in *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, vol. 9387 of *Lecture Notes in Computer Science*, pp. 538–546, Springer International Publishing, Cham, 2015.
- [32] I. Rochlin, Y. Aumann, D. Sarne, and L. Golosman, "Efficiency and fairness in team search with self-interested agents," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 3, pp. 526–552, 2016.
- [33] C. Skinner and S. Ramchurn, "The robocup rescue simulation platform[C]," in *Proceedings of the In the Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1647–1648, 2010.
- [34] J. Parker, E. Nunes, J. Godoy, and M. Gini, "Exploiting Spatial Locality and Heterogeneity of Agents for Search and Rescue Teamwork\*," *Journal of Field Robotics*, vol. 33, no. 7, pp. 877–900, 2016.
- [35] P. Sujit, J. George, and R. Beard, "Multiple UAV coalition formation," in *Proceedings of the 2008 American Control Conference (ACC '08)*, pp. 2010–2015, Seattle, WA, June 2008.
- [36] Q.-S. Hua, D. Yu, F. C. Lau, and Y. Wang, "Exact algorithms for set multicover and multiset multicover problems," in *Algorithms and computation*, vol. 5878 of *Lecture Notes in Comput. Sci.*, pp. 34–44, Springer, Berlin, 2009.
- [37] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proceedings of the 21st Annual Conference on World Wide Web, WWW'12*, pp. 839–848, France, April 2012.
- [38] S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proceedings of the 22nd International Conference on World Wide Web, WWW 2013*, pp. 1077–1087, Brazil, May 2013.
- [39] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9121–9134, 2009.
- [40] M. Farhangian, M. Purvis, M. Purvis, and T. B. Savarimuthu, "Agent-Based Modeling of Resource Allocation in Software Projects Based on Personality and Skill," in *Advances in Social Computing and Multiagent Systems*, vol. 541 of *Communications in Computer and Information Science*, pp. 130–146, Springer International Publishing, Cham, 2015.
- [41] L. Soriano Marcolino, H. Xu, D. Gerber et al., "Multi-agent Team Formation for Design Problems," in *Coordination, Organizations, Institutions, and Norms in Agent Systems XI*, vol. 9628 of *Lecture Notes in Computer Science*, pp. 354–375, Springer International Publishing, Cham, 2016.
- [42] Y. Tan and Y. Zhu, *Fireworks Algorithm for Optimization*, Springer-Verlag, Berlin, Heidelberg, Germany, 2010.
- [43] L. Chang, Z. Zhou, Y. You, L. Yang, and Z. Zhou, "Belief rule based expert system for classification problems with new rule activation and weight calculation procedures," *Information Sciences*, vol. 336, no. 1, pp. 75–91, 2016.
- [44] L. Barbulescu, J. P. Watson, L. D. Whitley, and A. E. Howe, "Scheduling space-ground communications for the air force satellite control network," *Journal of Scheduling*, vol. 7, no. 1, pp. 7–34, 2004.
- [45] A. J. Vazquez and R. S. Erwin, "On the tractability of satellite range scheduling," *Optimization Letters*, vol. 9, no. 2, pp. 311–327, 2015.
- [46] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pp. 2069–2077, Mexico, June 2013.
- [47] Y. Tan, *Fireworks algorithm: a novel swarm intelligence optimization method*, Springer, 2015.
- [48] N. Bacanin and M. Tuba, "Fireworks algorithm applied to constrained portfolio optimization problem," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015*, pp. 1242–1249, Japan, May 2015.
- [49] J. Li, S. Zheng, and Y. Tan, "The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 153–166, 2017.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

