Feeder Link Handover Strategy Based on Hybrid Clonal Selection Algorithm in the Dense LEO Constellation

Tianqi Yu[1], Qiang Zhang[2], Weixiao Meng[1](✉), Shuyi Chen[1], Sebastián E. Godoy[3], and Gabriel Saavedra[3]

[1] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China
120L032616@stu.hit.edu.cn, wxmeng@hit.edu.cn
[2] Beijing Institute of Spacecraft System Engineering, Beijing, China
[3] Department of Electrical Engineering, Universidad de Concepcón, Concepcón, Chile
{segodoy,gasaavedra}@udec.cl

Abstract. In order to extend the connection time of the feeder link for dense LEO constellation and make full use of the resources of the gateway station, this article proposes a feeder link handover strategy for dense LEO constellation based on Hybrid Clonal Selection Algorithm. Firstly, the satellite-earth visible window is defined as an independent task, the link handover problem is transformed into a task assignment problem, and the corresponding mathematical model is established. Then, the task execution time window is calculated using the Conflict Resolution Algorithm, and the associated satellite coincidence time window is reassigned. Then, the Clonal Selection Algorithm is used for global search to complete the whole link handover. In order to enhance the local optimization ability of the algorithm, a Hybrid Clonal Selection Algorithm is proposed by combining the local search algorithm of adaptive neighborhood selection. Finally, the model is set up and the performance comparative analysis experiment is carried out. The experimental results show that the Hybrid Clonal Selection Algorithm can improve the task scheduling rate and extend the link connection time without sacrificing the link switching times.

Keywords: Dense LEO Constellation · Feeder Link Handover Strategy · Hybrid Clonal Selection Algorithm

Supported by the 62201176 The National Science Fund for Young Scholars; 62271168 National Natural Science Foundation of China under Grand; YESS20210339 Young Elite Scientist Sponsorship Program by CAST; YQ2023F005 Natural Science Foundation of Heilongjiang Province of China; LJYXL2022-049 Heilongjiang Province Outstanding Doctoral Dissertation Funding Project; JD22A001 The Key R&D Plan of Heilongjiang Province of China under Grand.

H.-H. Chen and W. Meng (Eds.): WiSATS 2024, LNICST 606, pp. 381–398, 2025.
https://doi.org/10.1007/978-3-031-86203-8_30

1 Introduction

In the future space-air-ground integrated network, the LEO satellite network can supplement the vacant part of the ground communication network and provide network services to all parts of the world [1]. Due to its low-orbit characteristics, the satellite-to-ground transmission delay is greatly shortened, which can meet the current large-scale delay sensitive services and provide the possibility for real-time communication [2]. During the orbit of a LEO satellite, its subsatellite coverage area is constantly changing, so it needs to constantly switch the connection with the ground user terminal and the ground gateway station [3]. So far, the number of gateway stations in China is relatively small. In recent years, the number of LEO satellites has dramatically increased, and the distribution of resources of gateway stations has become particularly important [4]. Therefore, an urgent problem in the LEO satellite communication system is the efficient handover of the feeder link between the satellite and the gateway station.

The feeder link handover problem of LEO constellation belongs to the Satellite Range Scheduling Problem (SRSP). It is an NP-complete problem [5]. In general, both exact solution algorithms and heuristic algorithms can solve SRSP.

As for the exact solution algorithm, literature [6] solves the feeder link handover problem based on weighted bipartite graph. However, the exact solution algorithm can only be used to solve the SRSP for small-scale examples [7].As for the heuristic algorithm, both heuristic-based optimization algorithm and meta-heuristic algorithm can solve SRSP under large scale examples. For example, literature [8] uses the Greedy Algorithm to solve the resource scheduling problem of satellite ground station. Literature [9] adopted Simulated Annealing Algorithm to solve the ground station scheduling problem. Literature [10] adopted Improved Adaptive Genetic Algorithm to improve the efficiency of ground stations and reduce the degree of task conflict. Literature [11] uses Ant Colony Algorithm to transform the visible time window into task sequence to maximize the total task duration.

Clone Selection Algorithm (CSA) is a meta-heuristic algorithm with strong global search ability and excellent performance in solving NP-complete problems. However, with the increase of solving complexity, traditional CSA is easy to fall into local optimization, and the solving accuracy decreases. Literature [12] provides an improved scheme to solve this problem, but the effect is not good in the face of large-scale examples.

In this article, a Hybrid Clonal Selection Algorithm (HCSA) is proposed, which combines CSA with local search algorithm to efficiently solve the problem of feeder link handover in dense LEO constellation. Firstly, the visible windows are defined as independent tasks, the link handover problem is transformed into task assignment problem, and the Conflict Resolution Algorithm (CRA) is used to solve the task conflict problem. Then CSA is used to complete the feeder link handover through the process of coding, decoding and calling the cloning operator. Finally, a local search algorithm based on adaptive neighborhood selection is proposed to enhance the local search ability of the algorithm, which can be used to solve SRSP in large-scale examples.

2 System Model

There are m satellites in the system, represented by the set $S = \{1, 2, ..., m\}$; There are n gateway stations, represented by set $U = \{1, 2, ..., n\}$, and a single gateway station u is equipped with a pairs of antennas, represented by set $\Delta(u) = \{(u-1)a+1, (u-1)a+2, ..., (u-1)a+a\}$. A scheduling period $[0, T]$ is set for the dense LEO constellation system. In the scheduling period, if the satellite and the gateway station meet the visibility constraints, the link can be established with any antenna of the gateway station. Define the link building time t and the link breaking time e, then the feeder link period is $[t, e]$, that is, "visible window".

The objective of this paper is to extend the duration of the total "visible window" of the dense LEO constellation system as long as possible under the condition of satellite-earth visibility constraint during the scheduling period.

2.1 Task Model Construction

Build task set $G = \langle s_g, Q_g, [t_g, e_g], g, v_g \rangle$ for each "visible window". Where, g is the task number assigned to each "visible window", s_g represents the unique satellite associated with task g, $Q_g \subseteq K$ represents the set of antennas that meet the visibility constraints in task g, and $[t_g, e_g]$ represents the "visible window" of task serial number g.

The problem of feeder link handover is transformed into task assignment problem. Use $\Pi_k = \langle g_1, g_2, \ldots, g_{|\Pi_k|} \rangle$ to represent the sequence of tasks performed by the antenna k. Use T_g^k and E_g^k to indicate the start time and end time of antenna k executing task g, respectively. t_g and e_g are the start and end moments of task g before cutting. T_g^k and E_g^k are the start and end moments of task g executed by antenna k after cutting, satisfying $t_g \le T_g^k \le E_g^k \le e_g, \forall g \in \Pi_k$. The constraint conditions of each parameter are as follows (Fig. 1):

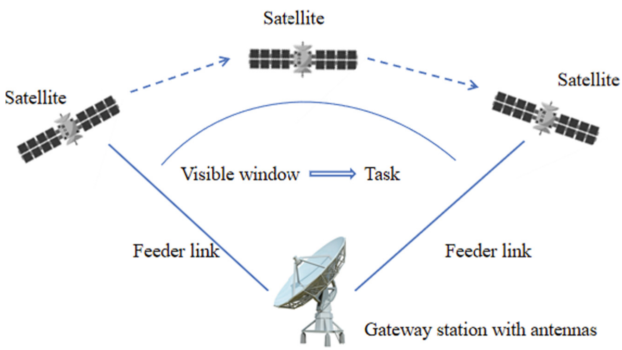


Fig. 1. Conversion of the feeder link handover problem

1) Only the antenna in Q_g can perform the task g:

$$k \in Q_g, \forall k, \forall g \in \Pi_k \tag{1}$$

2) Antenna turn adjustment time.

$$T_{g+1}^{k} - E_{g}^{k} \geq \Xi(g_x, g_{x+1}), \forall k, \forall x \in \{1, 2, \ldots, |\Pi_k| - 1\} \tag{2}$$

The adjustment time for the antenna to perform adjacent tasks g_x and g_{x+1} is $\Xi(g_x, g_{x+1})$:

$$\Xi(g_x, g_{x+1}) = \begin{cases} HT, s_{g_x} \neq s_{g_{x+1}} \\ 0, \ s_{g_x} = s_{g_{x+1}} \end{cases} \tag{3}$$

3) The task is constrained by the minimum time that the antenna executes PT.

$$E_g^k - T_g^k \geq PT, \forall g \in \Pi_k \tag{4}$$

4) Each task can only be performed by a single antenna at the same time.

$$\left|E_g^k - T_g^l\right|\left|E_g^l - T_g^k\right| \leq 0, \forall k, l \in K \, with \, k \neq l, \forall g \in \Pi_k \cap \Pi_l \tag{5}$$

The mathematical model of the optimization objective is as follows:

$$Max\, J = \sum_{k \in K} \sum_{g \in \Pi_k} \left(E_g^k - T_g^k\right) \tag{6}$$

2.2 Calculation of the Task Time Window

Problem Description: As shown in Fig. 2(a), task g and task h in the task sequence Π_k of antenna k are adjacent, $t_g \leq t_h$ and $e_g \geq t_h$. At this time, task h cannot be fully executed after task g is executed by the antenna line.

Solution: In order to solve the task time window conflict problem, a Conflict Resolution Algorithm (CRA) is proposed. On the premise of satisfying the formula (1)-(5), the start and end time of task g is kept unchanged, and the start time T_h^k and end time E_h^k of task h by antenna k is recalculated according to the start and end time t_h and e_h of task h, where $t_h \leq T_h^k \leq E_h^k \leq e_h, h \in \Pi_k$, The segmented task h can also be executed by other antennas to maximize resource utilization.

In fact, the CRA can be used to output the start and end time ($t_g \leq t_h$) of any two tasks g and h in the task sequence Π_k of antenna k being executed by antenna k.

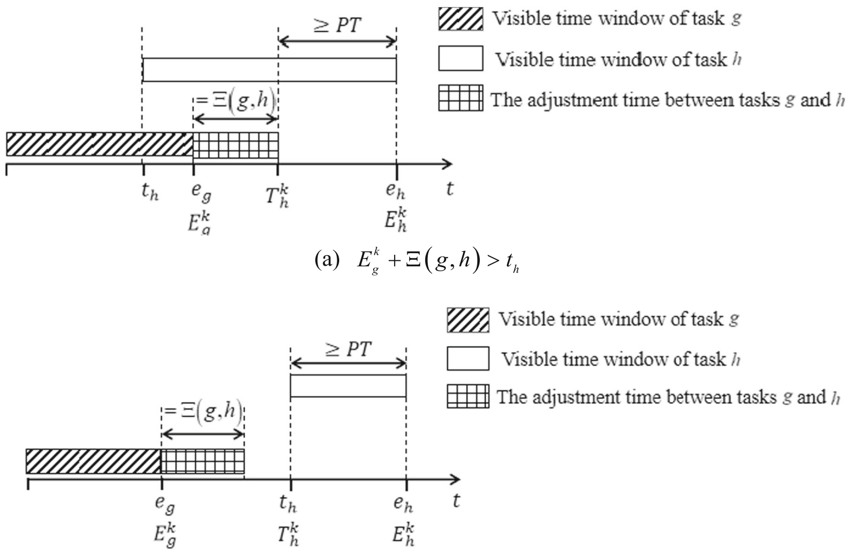
$$e_h - \max\left\{E_g^k + \Xi(g, h), t_h\right\} \geq PT \tag{7}$$

$$E_g^k + \Xi(g, h) \geq t_h \tag{8}$$

According to formula (7) and (8) is valid or not, it can be discussed in the following three cases.

1) Formula (7) is not valid. In this case, the time for task h to be executed by antenna k is less than the minimum time constraint PT, $T_h^k = E_h^k = 0$.

2) Formula (7) is valid, and formula (8) is not valid. In this case, the duration for task h to be executed by antenna k is longer than the minimum duration constraint PT, and the time after task g is switched over is later than the start time of task h, as shown in Fig. 2(a). Let $T_h^k = E_g^k + \Xi(g, h)$, $E_h^k = e_h$.
3) Formula (7) is valid, formula (8) is not valid. In this case, the duration for task h to be executed by antenna k is longer than the minimum duration constraint PT, and the time after task g is switched over is earlier than the start time of task h, as shown in Fig. 2(b). Let $T_h^k = t_h$, $E_h^k = e_h$.



(a) $E_g^k + \Xi(g,h) > t_h$

(b) $E_g^k + \Xi(g,h) \le t_h$

Fig. 2. Task g and h visible window

Table 1 shows the Conflict Resolution Algorithm:

2.3 Associated Satellite Coincidence Time Window Allocation

Since the satellite is uniquely associated with the task, when the task segment is assigned to the antenna, the satellite can no longer establish feeder links with other antennas during the visible window. Therefore, in other tasks associated with the satellite, the part that overlaps with the visible time window above needs to be reassigned to ensure that the satellite only establishes a feeder link with the single station antenna during the same period.

The time window $\left[T_h^k, E_h^k\right]$ assigned to antenna k after task h is cut can be obtained by calculating the CRA. Define task h' and time window $\left[T_h^k, E_h^k\right]$. According to the different visible Windows of satellite task j, the following cases are discussed, and the time window after reassignment is $\left[T_j, E_j\right]$.

Table 1. Conflict Resolution Algorithm

Conflict Resolution

Input: Antenna k task sequence Π_k any two tasks g , h

Output: T_h^k 、 E_h^k

```
if  The duration for task h to be executed by antenna k is less than the minimum duration
    constraint PT
    if  The switchover time of task g is later than the start time of task h
        Let T_h^k = E_g^k + Ξ(g,h), E_h^k = e_h ;
    else
        Let T_h^k = t_h , E_h^k = e_h ;
    end
else
    Let T_h^k = E_h^k = 0 ;
end
```

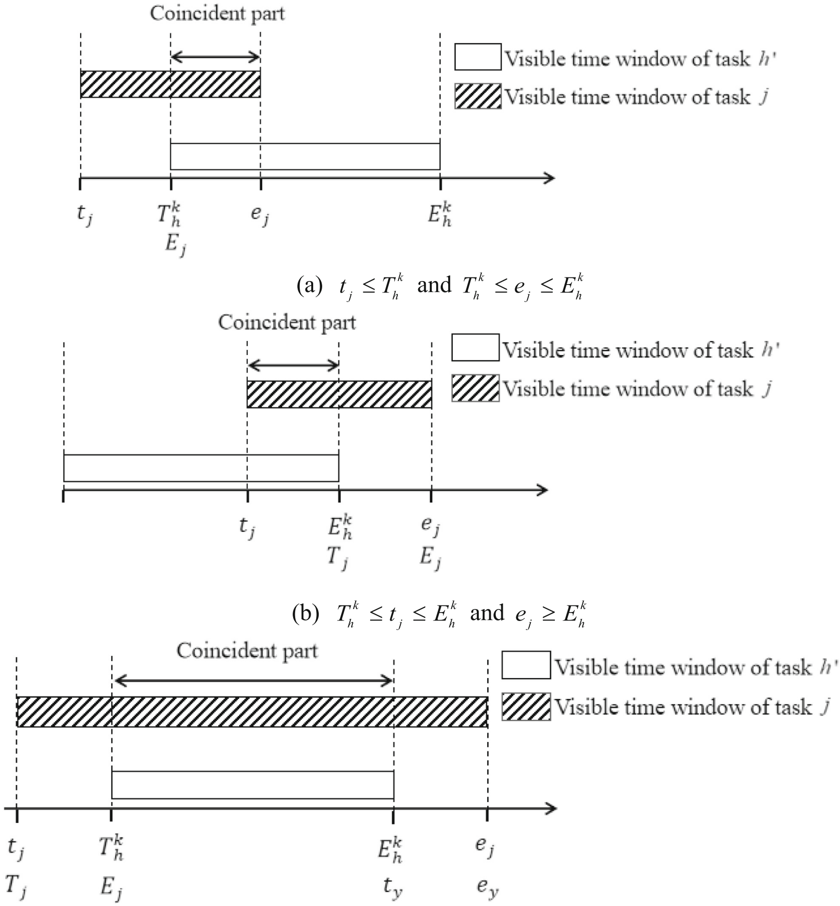
1) $e_j \leq T_h^k$ or $t_j \geq E_h^k$. At this time, task j and task h' do not coincide in the time window, so there is no need to reallocate, that is, $T_j = t_j$ and $E_j = e_j$.
2) $t_j \geq T_h^k$ and $e_j \leq E_h^k$. At this time, task j time window is completely covered by task h' time window, so task j cannot be executed by any antenna, and its time window is set to 0, that is, $T_j = E_j = 0$.
3) $t_j \leq T_h^k$ and $T_h^k \leq e_j \leq E_h^k$. The second half of the time window of task j coincides with the time window of task h', as shown in Fig. 3(a). Make $T_j = t_j$, $E_j = T_h^k$.
4) $T_h^k \leq t_j \leq E_h^k$ and $e_j \geq E_h^k$. At this time, the visible window of task j overlaps with the time window of task h', as shown in Fig. 3(b). $T_j = E_h^k$, $E_j = e_j$.
5) $t_j \leq T_h^k$ and $e_j \geq E_h^k$. In this case, task h' is completely overwritten in the time window, as shown in Fig. 3(c). Since task j is divided into three segments, the middle part overlaps, so let $T_j = t_j$, $E_j = T_h^k$, and set a new task y, its visible window $t_y = E_h^k$, $e_y = e_j$.

3 Hybrid Clonal Selection Algorithm for Feeder Link Handover in Dense LEO Constellation

3.1 Clonal Selection Algorithm

Define the following three concepts to distinguish them:

1) Antenna k executable task set, W_k: a set of tasks that meet the visibility constraints.
2) Task set assigned to antenna k, w_k: A new task set generated randomly from the antenna executable task set W_k.
3) Sequence of tasks performed by antenna k, Π_k: the sequence of tasks performed by the antenna in the final sequence.



(a) $t_j \le T_h^k$ and $T_h^k \le e_j \le E_h^k$

(b) $T_h^k \le t_j \le E_h^k$ and $e_j \ge E_h^k$

(c) $t_j \le T_h^k$ and $e_j \ge E_h^k$

Fig. 3. Associated satellite coincidence time window reallocation

Coding. $n \times a$ subclass antibody constitutes antibody π, and N_P antibodies constitute population P. Where, the subclass antibody π_k represents the situation where tasks are randomly assigned to antenna k, the antibody π represents the situation where tasks are randomly assigned to all antennas, and the population P represents the set of randomly formed antibodies π.Subclass antibodies π_k are $|W_k|$ dimensional binary vectors, $W_k = \{g \in G | k \in Q_g\}$. For task g in W_k, if $\pi_k[g] = 1$, the task is added to set w_k, $w_k = \{g \in W_k | \pi_k[g] = 1\}$. Coding is the process by which the antibody π gets the task set $w = [w_1, \ldots, w_{n \times a}]$.

Decoding. From the task set w_k assigned to the antenna, the task sequence Π_k performed by the antenna is obtained. The corresponding directed graph $Graph_k = (V, A, f)$ must be obtained from the task set w_k.

Here are the three parts of the $Graph_k$: V, A, f.

1) Node set V: Node set $V = \{v_0, v_1, \ldots, v_{|w_k|}, v_{|w_k|+1}\}$. $\{v_1, \ldots, v_{|w_k|}\}$ corresponds to each task in w_k. Define the source node of the start task g_s and the destination node of the end task g_e. Specify time Windows for starting and ending tasks as $[-PT, 0]$ and $[T, T + PT]$, respectively. The antenna turn adjustment time between g_s and g_e and the other task g is 0.
2) Arc set A: Any two tasks g and h in task set w_k can establish an arc (v_g, v_h), and arc set A contains all arcs.
3) Arc weight set f: The visible window $\left[T_g^k, E_g^k\right]$ and $\left[T_h^k, E_h^k\right]$ of any two tasks g and h $(t_g \leq t_h)$ in task set w_k can be obtained by the CRA. Then the weight of the arc between nodes v_g and v_h is $f_{g,h} = T_h^k - E_g^k$. If $t_g > t_h$, then the weight $f_{g,h}$ of the arc between nodes v_g and v_h is infinite.

By inputting the obtained directed graph $Graph_k$ into Dijkstra algorithm, the task sequence Π_k executed by antenna k can be obtained. The sequence of tasks performed by all antennas in turn forms solution set $\Pi = [\Pi_1, \ldots, \Pi_{n\times a}]$.

Antibody Affinity Calculation. In order to evaluate the quality of antibody π, antibody affinity $F(\pi)$ is defined by the following formula:

$$F(\pi) = \sum_{k\in K} \sum_{g\in\Pi_k} \left(E_g^k - T_g^k\right) / \sum_{g\in G} \left(e_g - t_g\right) \tag{9}$$

Clonal Selection Operator. In order to further evaluate the overall quality of the antibody and define the antibody excitation degree $sim(\pi)$, the calculation process is given below.

$$aff\left(\pi, \pi'\right) = \sum_{k\in K} \sum_{g\in W_k} \partial_k\left(\pi, \pi', g\right) \tag{10}$$

$$\partial_k\left(\pi, \pi', g\right) = \begin{cases} 1, \textit{if } \pi_k[g] = \pi_k'[g] \\ 0, \textit{else} \end{cases} \tag{11}$$

$$R\left(\pi\pi'\right) = \begin{cases} 1, \textit{if } aff\left(\pi, \pi'\right) / \sum_{k=1}^{n\times a} |W_k| > \sigma \\ 0, \textit{else} \end{cases} \tag{12}$$

$$den(\pi) = 1/N_P \times \sum_{\pi'\in P} R\left(\pi, \pi'\right) \tag{13}$$

$$AF(\pi) = (F(\pi) - F_{\min})/(F_{\max} - F_{\min}) \tag{14}$$

$$sim(\pi) = AF(\pi) \times e^{-den(\pi)/\beta} \tag{15}$$

In the formula, $\sigma = 0.5$, $\beta = 2$,$F_{\min}$ and $F_{\max}$ are the minimum and maximum affinity values of all antibodies in the population P, respectively.

Cloning Operator. The cloning operator is called to copy $cls(\pi)$ copies of each antibody π in the original population P to form a clonal antibody group $clone(\pi)$ with

N_P antibodies π. All the replicating antibodies constitute a new population, P_c, called the clonal population. The parameter $cls(\pi)$ is called the clone scale of π, and the calculation formula is as follows:

$$cls(\pi) = M\left(N_c \times sim(\pi) / \sum_{\pi' \in p} sim(\pi') + \varphi\right) \tag{16}$$

In the formula, N_c is the clone size of population P, $N_c = 2 \times N_P$. $\varphi = 1.2$.$M[\cdot]$ is the downward integer function, for example $M[1.5] = 1$.

Mutation Operator. The mutation operator is called to mutate the replication antibody π in clonal population P_c, and the newly formed antibody forms mutation population P_c', so as to achieve local search.

The mutation probability $pm(\pi)$ of $\pi \in P_c$ is calculated as follows:

$$pm(\pi) = \left[\exp\left(N_P \times AF(\pi) / \sum_{\pi' \in P} AF(\pi')\right) + \rho\right]^{-1} \tag{17}$$

In the formula, ρ is the mutation probability adjustment factor, let $\rho = 1$.

The mutation operator generates a random number *rand* for each bit of antibody π coding $\pi_k[g]$, and *rand* is evenly distributed across the interval [0, 1]. When $rand < pm(\pi)$, let $\pi_k[g] = 1 - \pi_k[g]$; When $rand \geq pm(\pi)$, $\pi_k[g]$ stays the same.

Clone Suppression. First, all the antibodies in the original population P are assigned to the next generation population P', so that $P' = P$. Then, for each antibody π of the next generation population P', the antibody with the highest affinity of clonal antibody group $clone(\pi)$ after mutation operator is set to π^b, if $F(\pi^b) > F(\pi)$, the original antibody π of P' is replaced by π^b. After updating, all antibodies in the next generation population P' were superior to those in the original population P.

Population Refresh. All the antibodies in P' are sorted according to the affinity of antibodies from the largest to the smallest, and the $M[pref \times N_P]$ antibodies with the lowest affinity are eliminated, where $M[\cdot]$ is the downward integer function, *pref* is the population refresh ratio, and $pref = 0.35$ is set. Then, $M[pref \times N_P]$ new antibodies are randomly generated and added to the next generation population P', so that P' completes the second update.

3.2 A Local Search Algorithm

In this section, a local search algorithm based on adaptive neighborhood selection is proposed to enhance the performance of the algorithm in local optimization.

1) add action: Randomly select the component in antibody π whose coding value is equal to 0, and turn $\pi_k[g] = 0$ into $\pi_k[g] = 1$. The add action corresponds to the neighborhood structure $N_1(\pi)$.
2) remove action: Randomly select task g in set Π, satisfying $\pi_k[g] = 1$, and let $\pi_k[g] = 0$. The remove action corresponds to neighborhood structure $N_2(\pi)$.

3) exchange action: In set Q_g, two pairs of antennas k and l that meet condition $\pi_k[g] \neq \pi_l[g]$ are randomly selected, and the values of $\pi_k[g]$ and $\pi_l[g]$ are exchanged. The exchange action corresponds to the neighborhood structure $N_3(\pi)$.

Table 2 shows the local search algorithm.

Table 2. Local search algorithm based on adaptive neighborhood selection

Local Search
Input: antibody π , Threshold parameter Ψ
Output: antibody π_{best}
Initialize: let $\psi = 1$, $\pi_{best} = \pi$, $p_r = 1/3$; $suc_r = 0$, $sel_r = 0$, $r \in \{1,2,3\}$;
while $\psi \leq \Psi$ **do**
The new antibody π' is generated by randomly selecting the neighborhood structure $N_r(\pi)$ in the way of roulette;
$sel_r = sel_r + 1$;
if $F(\pi') > F(\pi)$ **then**
$\pi_{best} = \pi'$ and $\pi = \pi'$; $suc_r = suc_r + 1$;
end
$\psi = \psi + 1$;
Update every 25 iterations according to formula (18) and formula (19);
end
Output locally optimal antibody π_{best};

Each iteration of the local search algorithm will randomly select one of the three neighborhood structures in the way of roulette, and the probability of each neighborhood structure being selected is p_1, p_2, p_3. For each iteration of 25 rounds, the probability will be updated according to formula (18) and (19).

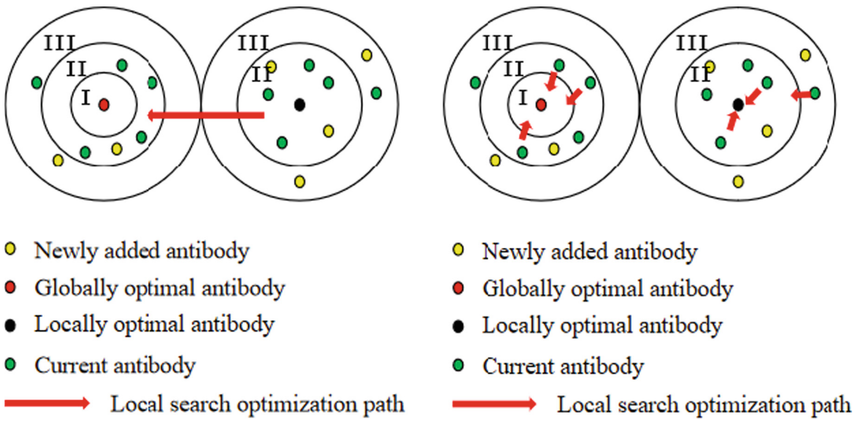
$$p'_r = \zeta \times p_r + (1 - \zeta) \times suc_r / sel_r, r \in \{1, 2, 3\} \tag{18}$$

$$p_r = p'_r / \sum_{m=1}^{3} p'_m, r \in \{1, 2, 3\} \tag{19}$$

In the formula, $\zeta = 0.7$, sel_r is the number of times $N_r(\pi)$ is selected in every 25 iterations; suc_r is the number of times in each 25 iterations that $N_r(\pi)$ is selected to generate better antibodies. Table 2 shows the local search algorithm based on adaptive neighborhood selection.

Figure 4 shows how local search algorithm improves CSA.

As shown in Fig. 4(a), newly added antibodies may be near the global optimal solution or the local optimal solution. However, it is almost impossible for antibodies distributed near the local optimal solution to be optimized to the global optimal solution. This will result in many antibodies used to calculate the local optimal solution instead



(a) Stuck in local optimality (b) Stuck in optimization stagnation

Fig. 4. How local search algorithm improves CSA

of the global optimal solution in the iteration process, that is, trapped in local optimality. The remove action in the local search algorithm can directly change the current optimal path, so that the antibodies near the local optimal solution have more chances to evolve to the global optimal solution.

As shown in Fig. 4(b), antibodies with poor affinity in layer III will be eliminated or optimized to layer II. However, the affinity of antibodies in layer II is in the middle, which means that the antibodies in this layer cannot be eliminated, and because it is far from the global optimal antibody, it cannot be optimized by high-frequency mutations. Therefore, more and more antibodies enter and stay in layer II, which makes the antibody concentration too high and the diversity insufficient. In this case, the local search algorithm can accelerate the antibody self-optimization process, making the antibody in layer II accelerate into the layer I.

3.3 Hybrid Clonal Selection Algorithm

Combining CSA with local search algorithm, a Hybrid Clonal Selection Algorithm (HCSA) with stronger search ability is proposed. The specific process is as follows:

1) Generate the original population P containing N_P antibody randomly, complete the coding of all antibodies in P, set the current iteration number $\phi = 1$, set the maximum iteration number Φ of the algorithm, and complete the initialization.
2) Call the cloning operator to expand the original population P into a clonal population P_c.
3) Call the mutation operator to transform clonal population P_c into mutation population P_c'.
4) Call the clone suppression and population refresh operators to generate and optimize the next generation population P' and take it as the new original population, $P = P'$.
5) Use the local search algorithm based on adaptive neighborhood selection for local search and self-optimization of all antibodies in P to improve antibody quality.

6) Update the current iteration number $\phi = \phi + 1$, if $\phi \le \Phi$, the program returns to step 2); If $\phi > \Phi$, the program ends, the optimal antibody obtained in the whole process and the corresponding task sequence Π are output.

4 Numerical Results

4.1 Model Construction

STK software is used to build a dense LEO constellation model and a gateway station model. Based on the data provided by the latest Starlink LEO satellite, the parameters of the Walker-Delta constellation model are shown in Table 3.

Table 3. The parameters of the Walker-Delta constellation model

Parameter	Value
Altitude	550 km
Planes	72
Satellite per plane	22
Inclination	53°
Cone Half Angle	56.5°
Inter Plane Spacing	39

Select 12 cities in China to deploy the gateway stations, the minimum elevation Angle of the gateway stations is set to 25°, and each gateway station is equipped with 4 antennas. Gateway stations are equipped with sensors, with the cone half angle set to 65°.

4.2 Algorithm Evaluation Index and Parameter Setting

Two evaluation indexes are proposed: task scheduling rate and average number of link switching times.

Define the task scheduling rate, as in formula (20):

$$\mu = \sum_{k \in K} \sum_{g \in \Pi_k} \left(E_g^k - T_g^k \right) / (T \times n \times a) \tag{20}$$

Define the average number of link switching times N_{total} in each scheduling period of a single station and single antenna, which is referred to as the average number of link switching times, as shown in formula (21):

$$N_{ave} = \frac{N_{total}}{n \times a} \tag{21}$$

In the formula, N_{total} is the total number of link switching times of all gateway antennas in the simulation process.

Table 4 describes the parameters of the algorithm.

Table 4. Algorithm parameter setting

Parameter	Value
Scheduling duration T	10 min
Number of antennas in a gateway station a	4
Population scale N_P	50
Task adjustment time HT	20 s
Minimum link duration PT	20 s
CSA/HCSA maximum iterations Φ	20
Local search algorithm maximum iterations Ψ	100

4.3 Analysis of Conflict Resolution Algorithm

Firstly, the validity of the CRA is verified. The task processing method of CRA is changed to single task selection without task cutting, and the new algorithm is defined as Single Task Selection Algorithm (STSA).

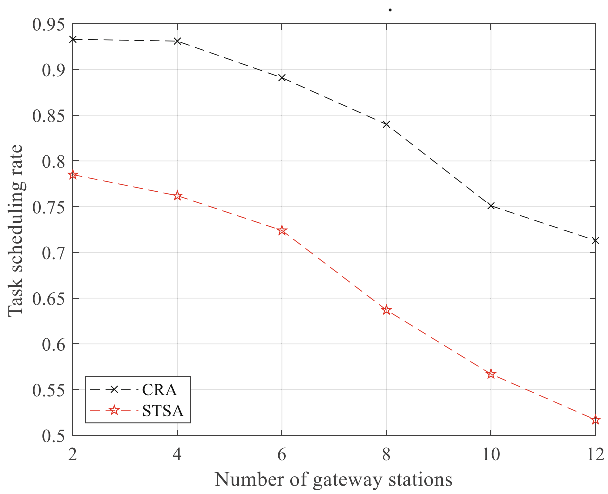


Fig. 5. Scheduling rates under different task processing algorithms

The performance of the CRA and the STSA was analyzed, and the task scheduling rate of the two was compared, as shown in Fig. 5. Among them, the two task processing algorithms are combined with Greedy Algorithm (GA) to avoid the influence of random factors on the experiment.

As shown in Fig. 6, the task scheduling rate of the CRA is significantly improved compared with the STSA. This is because the CRA "cuts" the visible window, and the cut fragments can be allocated to different antennas. In the STSA, when the task is assigned to the antenna, the remaining visible window cannot be used by other antennas.

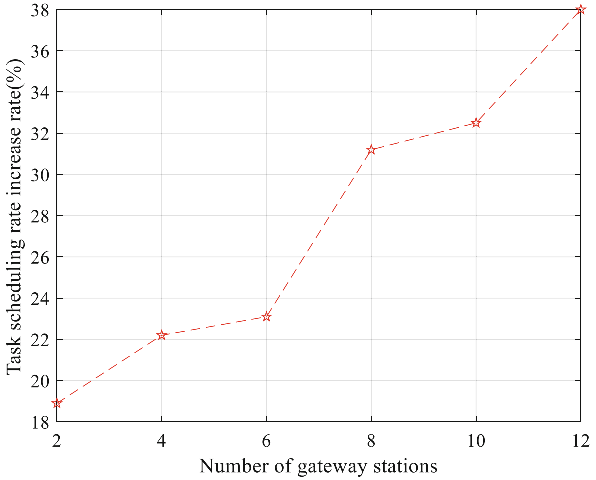


Fig. 6. Scheduling rate of CRA is higher than STSA

Therefore, with the increase of the number of gateway stations, the number of tasks that can be assigned to the antenna in the STSA decreases sharply. However, there are still a large number of task fragments in the CRA.

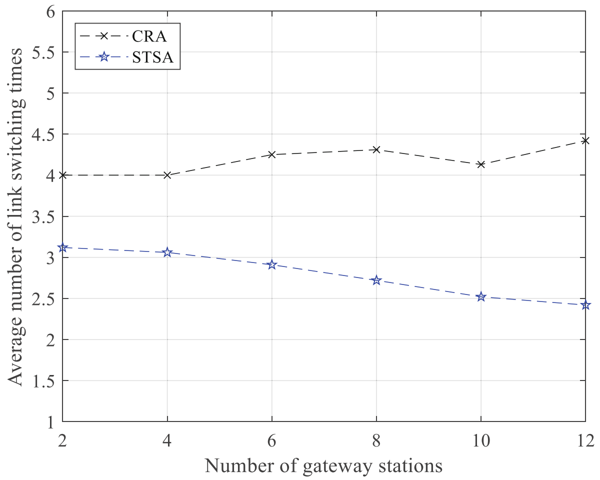


Fig. 7. Average number of link switching times under task processing algorithms

As shown in Fig. 7, the average number of feeder link switching times under the STSA is significantly lower than that under the CRA. This does not mean that STSA is better. Because under the STSA, it is found that some antennas are completely vacant, resulting in the feeder link cannot be switched.

Figure 8 shows an example. When the number of gateway stations is 12, the final task sequence of the antenna under the STSA is shown in the figure. The row number is the antenna number, a total of 48 antennas. The non-zero number represents the task number. It can be seen that No. 32, No. 34, No. 35, No. 36, No. 44, No. 47, No. 48 antenna has no task to perform, that is, no feeder link is established with the satellite within the scheduling time, and the antenna task is vacant.

29	0	323	0	0	0
30	0	342	0	0	0
31	0	345	0	0	0
32	0	0	0	0	0
33	0	372	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	380	391	0	0
38	0	386	0	0	0
39	0	379	0	0	0
40	0	397	0	0	0
41	0	449	431	0	0
42	0	435	0	0	0
43	0	439	0	0	0
44	0	0	0	0	0
45	0	485	0	0	0
46	0	462	478	0	0
47	0	0	0	0	0
48	0	0	0	0	0

Fig. 8. Task execution sequence of STSA when the number of stations is 12

Therefore, STSA cannot solve the problem of dense constellation feeder link handover well. In contrast, the CRA can effectively realize the feeder link handover and improve the resource utilization rate of the gateway station, which is suitable for solving the SRSP in large-scale calculation cases.

4.4 Analysis of Hybrid Clone Selection Algorithm

Select 2, 4, 6, 8, 10 and 12 stations from the established model, and use GA, CSA and HCSA respectively to calculate their task scheduling rates. Among them, all the task processing algorithms choose CRA. Since the Clone Selection Algorithm and the Hybrid Clone Selection Algorithm are meta-heuristic algorithms with randomness, each experiment is repeated 10 times and the average task scheduling rate is taken. GA is a heuristic algorithm, the results are not random, a single experiment can be done. Table 4 lists the parameters of the algorithm.

As shown in Fig. 9, the task scheduling rate varies with the number of gateway stations under GA, CSA, and HCSA. It can be seen that with the increase of the number of gateway stations, the task scheduling rate of the three algorithms decreases. This is because when the number of gateway stations is small, there are a large number of LEO satellites visible and not yet occupied, so the gateway station can choose almost all the satellites in the visible range to establish feeder links. With the increase in the number of gateway stations, the first few gateway stations have established feeder links with LEO satellites, while the remaining gateway stations can no longer establish links with occupied satellites, so the number of satellites that can be used is greatly reduced, resulting in a decline in task scheduling rate. Among them, the task scheduling rate of GA decreases the fastest, and the task scheduling rate of HCSA decreases the slowest.

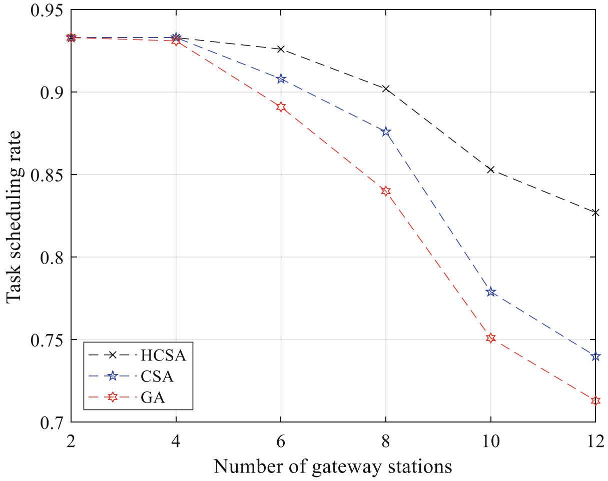


Fig. 9. Task scheduling rate under different algorithms

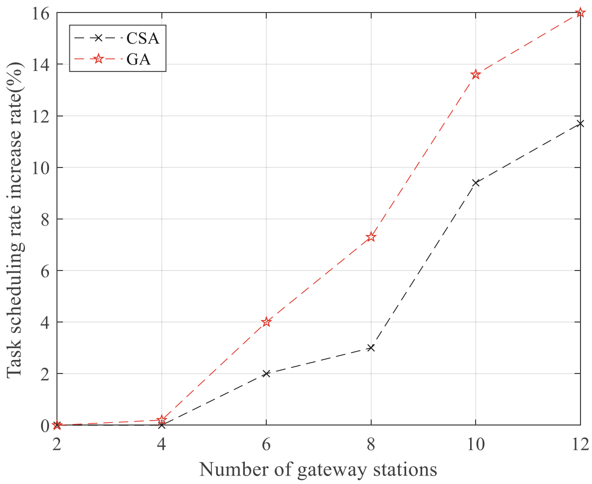


Fig. 10. The HCSA has higher task scheduling rate than other algorithms

As shown in Fig. 10, with the increase in the number of gateway stations, it can be seen that the task scheduling rate of the HCSA is higher than that of other algorithms. Compared with GA, when the number of gateway stations is 12, the task scheduling rate increases by 16.0%. Compared with the CSA, when the number of ground stations is 12, the task scheduling rate increases by 11.7%.

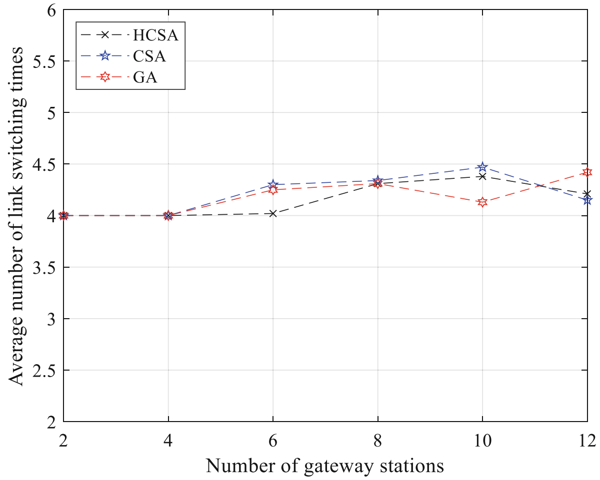


Fig. 11. Average number of link switching times under different algorithms

The following analyzes the average number of feeder link switching times under different algorithms, as shown in Fig. 11. It can be seen that when the number of gateway stations is small, there is little difference in the switching times of feeder links under the three algorithms, which is relatively stable. With the increase of the number of gateway stations, the GA has a large difference in the number of feeder link switching times compared with the other two algorithms. This is because the GA does not consider global optimization, so the result is quite different from the global optimal solution. However, the switching times of feeder link under HCSA and CSA are not significantly more than that under GA. Therefore, the HCSA can effectively improve the task scheduling rate without sacrificing the number of switching times.

5 Conclusion

In order to extend the connection time of the feeder link for dense low orbit constellation and make full use of the resources of the gateway station, a link handover strategy for dense LEO constellation based on HCSA is proposed. The simulation results show that compared with the STSA, the CRA can effectively utilize the station resources, and is suitable for the actual scenario where the constellation scale is expanding. The HCSA enhances the local optimization ability of the algorithm by combining the local search algorithm, which can effectively improve the task scheduling rate. The HCSA could extend the link connection time without sacrificing the link switching times, and is suitable for solving the problem of feeder link handover in dense LEO constellation.

References

1. Chen, Q., Yang, L., Guo, J., et al.: LEO mega-constellation network: networking technologies and state of the art. J. Commun. **43**(05), 177–189 (2022)
2. Jin, L., Wang, L., Jin, X., et al.: Research on the application of LEO satellite in IOT. In: 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), pp. 739–741 (2022)
3. Judice, A., Venusamy, K., Livin, J.:Multilayer LEO satellite constellation coverage analysis and its current research directions. In: 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), pp. 1–5 (2022)
4. He, G., Gao, X., Sun, L., et al.: A review of multibeam phased array antennas as LEO satellite constellation ground station. IEEE Access **9**, 147142–147154 (2021)
5. Barbulescu, L., Watson, J.P., Whitley, L.D., et al.: Scheduling space-ground communications for the air force satellite control network. J. Sched. **7**(1), 7–34 (2004)
6. Feng, L., Liu, Y., Wu, L., et al.: A satellite handover strategy based on mimo technology in LEO satellite networks. IEEE Commun. Lett. **7**, 1505–1509 (2020)
7. Song, Y., Ma, X., Li, X., et al.: Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem. Swarm Evol. Comput. **49**, 194–205 (2019)
8. Zhang, H.: Resource scheduling method of satellite ground station based on greedy algorithm. Radio Eng. **40**(12), 4–6, 30 (2010)
9. Xhafa, F., Herrero, X., Barolli, A., et al.: A simulated annealing algorithm for ground station scheduling problem. In: 2013 16th International Conference on Network-Based Information Systems, pp. 24–30 (2013)
10. Lin, H., Tang, X., Zhang, B., et al.: Multi-resource optimal scheduling method for satellite communication based on improved adaptive genetic algorithm. In: 2022 7th International Conference on Computer and Communication Systems (ICCCS), pp. 604–609 (2022)
11. Jin, Y., Gao, Y., et al.: Mission planning method of satellite ground station based on ant colony algorithm. Comput. Modern. **12**, 11–15 (2012)
12. Pan, C., Luo, J., Zhou, L., et al.: Planar Acrobot linear active disturbance rejection robust stabilization based on immune optimization. Control Decis. **35**(12), 3053–3058 (2020)