Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 70 (2022) 286–302

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Multiple GEO satellites on-orbit repairing mission planning using large neighborhood search-adaptive genetic algorithm

Peng Han, Yanning Guo, Chuanjiang Li [*], Hui Zhi, Yueyong Lv

Harbin Institute of Technology, Harbin 150001, China

Received 18 November 2021; received in revised form 2 April 2022; accepted 12 April 2022
Available online 20 April 2022

Abstract

This paper proposes a novel large neighborhood search-adaptive genetic algorithm (LNS-AGA) for many-to-many on-orbit repairing mission planning of geosynchronous orbit (GEO) satellites with mission deadline constraint. In the many-to-many on-orbit repairing scenario, several servicing spacecrafts and target satellites are located in GEO orbits which have different inclination, right ascension of ascending node (RAAN) and true anomaly. Each servicing spacecraft need to rendezvous with target satellites to perform repairing missions under limited fuel. The mission objective is to find the optimal servicing sequence and orbit rendezvous time of each servicing spacecraft to minimize total Δv cost of all servicing spacecrafts with all target satellites repaired. Firstly, a time-dependent orbital rendezvous strategy is proposed, which can handle the mission deadline constraint. Besides, it is also cost-effective compared with the existing strategy. Based on this strategy, the many-to-many on-orbit repairing mission planning model can be simplified to an integer programming problem, which is established based on the vehicle routing problem with time windows (VRPTW) model. In order to efficiently find a feasible optimal solution under complicated constraints, a hybrid adaptive genetic algorithm combining the large neighborhood search procedure is designed. The operations of "destroy" and "repair" are used on the elite individuals in each generation of the genetic algorithm to enhance local search capabilities. Finally, simulation results under several different scenarios show the effectiveness and superiority of the presented algorithm and orbital rendezvous strategy.

Keywords: Geosynchronous satellite; Mission planning; On-orbit repairing; Adaptive genetic algorithm; Large neighborhood search

1. Introduction

With the rapid development of aerospace technology, on-orbit servicing (OOS) compromising refueling, repair, assembly, debris removal has attracted much attention from researchers and engineers (Rousso et al., 2021; Davis et al., 2019). During the past decades, one of the most successful OOS missions is the upgrade and repair of the Hubble Space Telescope achieved by space shuttles (Li et al., 2019; Hastings and Joppin, 2006). However, the space shuttle was permanently retired in 2011 due to its high cost. In the future, with the increment in space exploration, the cost of on-orbit servicing needs to be further reduced. Thus, the OOS modes of one-to-many and many-to-many is proposed, i.e., one or more servicing spacecraft (SSC) provide services to multiple satellites successively. How to effectively manage and schedulemultiple SSC to obtain as many service benefits as possible needs to be solved urgently. Therefore, the study of mission planning and orbit maneuver strategy of multiple servicing spacecraft has great significance.

This paper mainly focuses on the repairing of geosynchronous satellites. It is known that nearly hundreds of high-value satellites are parked in GEO which play an important role in earth observation, communication relay,

[*] Corresponding author.
E-mail address: lichuan@hit.edu.cn (C. Li).

https://doi.org/10.1016/j.asr.2022.04.034
0273-1177/© 2022 COSPAR. Published by Elsevier B.V. All rights reserved.

and meteorological observation. Once the satellite is broken or run out of fuel during daily operation, building and launching a new satellite is more costly than repair the old one. Besides, the geosynchronous orbit is a precious space resource, the broken satellite will cause a waste of GEO resources (Qi et al., 2019). Therefore, building an OOS system for GEO satellites is necessary and cost-effective.

Many researchers have studied the OOS mission planning problem for multiple satellites in the past few years, which can be divided into three major categories based on the OOS mission type. (i). On-orbit refueling problem (Zhang et al., 2014; Zhou et al., 2015b; Zhou et al., 2015a; Zhao et al., 2017; Zhang et al., 2019) which is necessary to consider how to reasonably allocate the servicing spacecraft's limited fuel to orbit maneuver or target satellites. A special example is (Du et al., 2015) proposed a cooperative maneuver strategy between service vehicle and target satellite, simulation results indicated that the cooperative strategy is more fuel-efficient than the non-cooperative strategy. (ii). Debris removal problem (Li and Baoyin, 2019; Bang and Ahn, 2019; Shen et al., 2018; Yu et al., 2014). For the debris removal mission, the main difference from other OOS missions is that the SSC needs to take the target to the grave orbit after rendezvous to the target. (iii). Mixed types (Daneshjou et al., 2017; Sarton du Jonchay et al., 2021). (Shen and Tsiotras, 2002) did a pioneering work that solving the optimal scheduling problem for a circular satellite constellation on-orbit servicing including repairing faulty hardware, upgrading the operating systems, refueling the satellites. In the mixed OOS mission scenario, one obvious flaw is that each SSC needs to carry a variety of loads to meet different OOS missions, so that the SSC will have a huge dry weight which leads to more fuel consumption during the orbital maneuver. To authors' best knowledge, the research on on-orbit repairing mission planning is still rare. A typical example of on-orbit repairing is replacing a failed battery with a new battery and assisting a solar array that failed to deploy properly (Davis et al., 2019). The lack of power supply may bring about permanent damage to some equipment which causes enormous economic loss to ground users. Therefore, compared with the other OOS mission, the on-orbit repairing mission usually has more urgent mission completion time. Thus, we mainly focus on the many-to-many on-orbit repairing mission planning problem with mission deadline constraints for GEO satellites which is a combinatorial-continuous problem for simultaneously determining the servicing route and orbital rendezvous time of each SSC.

Different types of many-to-many OOS missions can be seen as a multi-spacecraft continuous rendezvous problem (Federici et al., 2021). To solve this problem, the orbital rendezvous strategy must be clarified to simulate the SSC maneuver from one target satellite to another. In this paper, the SSC and target satellites are located in the GEO with different inclination, RAAN and true anomaly. Existing literature usually use a time-independent orbital rendezvous model in which the process of phasing maneuver is usually ignored (Chen and Yu, 2017; Zhang et al., 2019; Zhou et al., 2015b; He and Melton, 2019). However, the orbital rendezvous time must be controlled and determined to deal with the mission deadline constraint. Some literature also propose the time-relevant orbit maneuver strategy: (Daneshjou et al., 2017) uses Lambert algorithm to determine the rendezvous trajectory. Based on the Lambert algorithm, (Federici et al., 2021) proposes a four-impulse strategy for multirendezvous trajectories optimization problem. However, it is only applicable to coplanar orbital transfer. Inspired by (Zhou et al., 2015b), a mixed time-dependent orbital rendezvous strategy combing the planar change and phasing maneuver is presented in this paper. Simultaneously, the mixed strategy can usually generate a more cost-effective trajectory than the Lambert algorithm. The orbit rendezvous time can be specified by determining the number of revolutions for the phasing maneuver, hence the mission planning problem can be transformed from the combinatorial-continuous problem to an integer programming problem, moreover, the solution space can be further reduced. Thus, the mission planning model is established based on the VRPTW (Ombuki et al., 2006) model, in which the servicing route and number of revolutions for each SSC needs to be determined.

To solve the many-to-many OOS mission planning problem which is obviously NP-hard, some researchers used the deterministic algorithm, e.g. column-generation technique(Bang and Ahn, 2019), branch and bound (Chen and Ho, 2018). Those method usually need to approximate the mathematical model to a linear model. Besides, they are very time-consuming even ineffective for large-scale problems. Intelligent metaheuristic algorithms are mostly used in the existing literature like particle swarm algorithm (Chen and Yu, 2017; Zhou et al., 2015b; Daneshjou et al., 2017), ant colony algorithm (Zhang et al., 2019; Shen et al., 2018), simulated annealing (Li and Xu, 2020), evolutionary algorithms (Federici et al., 2021; Danilo et al., 2021), etc. Intelligent algorithms usually have a flexible form that can handle complex constraints and objective functions. Genetic algorithm (GA) is also a metaheuristic algorithm which is widely used in aerospace mission design, like on-orbit refueling problems (Zhang et al., 2014; Zhou et al., 2015a; Zhao et al., 2017) and earth-observing scheduling (Wang et al., 2019). GA has a great ability of global exploration in the solution space, however, it does not employ a local search and is easy to fall into local optima (Baniamerian et al., 2019). Therefore, it may degrade the solution quality even get an unfeasible solution under tight mission time constraints. To make up for this defect, we present a hybrid algorithm called large neighborhood search adaptive genetic algorithm (LNS-AGA). The large neighborhood search (LNS) (Ropke and Pisinger, 2006; Pisinger and Ropke, 2010; Shaw, 1998) that has been widely used in routing and scheduling problems is adopted to enhance the general

GA's local search capabilities. In addition, the adaptive crossover probability and mutation probability are used in the GA procedure to prevent falling into local optima and speed up the iterative efficiency.

The main contribution of this paper can be summarized as follows:

(1) The mission deadline constraint is considered in the on-orbit repairing mission planning problem, which has a significant meaning in practice.
(2) A time-dependent mixed orbital rendezvous strategy is proposed to simulate the SSC maneuver from one target satellite to another, which is more cost-effective than the Lambert algorithm used in (Daneshjou et al., 2017). Then, the mission planning problem can be formulated as an integer programming problem, the solution space can be further reduced.
(3) A hybrid LNS-AGA is designed for solving the mission planning problem, which performs a local search procedure that can have better performance than traditional genetic algorithm and some intelligent algorithms.

The rest of the paper is organized as follows. In Section 2, the mission planning problem is further clarified including a more specific description of the mission scenario, design of the orbital rendezvous strategy and the mathematical model's establishment. After that, Section 3 introduces the design of hybrid LNS-AGA algorithm used for effectively solving this problem. Later on, Section 4 presents one practical scenario and several comparative experiments to demonstrate the effectiveness of the proposed methods. Finally, Section 5 gives some conclusions of this paper and the future path of this research topic.

2. Problem formulation

2.1. Many-to-many on-orbit repairing mission scenario

In the many-to-many on-orbit repairing scenario, multiple repairing required target satellites and SSC are running on GEO, with different orbital inclination, RAAN, and phase angle. Every target satellite needs to be repaired by exactly one SSC and each SSC can provide on-orbit repairing for any target satellite. The on-orbit repairing mission must be completed before the pre-set mission deadline. Before performing on-orbit repairing to a target, the SSC should first perform orbital rendezvous to close to the target and dock with the target. The SSC will make a velocity increment by impulse during preform orbital rendezvous. Every SSC could carry a limited fuel for orbital rendezvous, so the SSC must satisfy the maximum velocity increment constraint throughout the whole mission process. Otherwise, it will not be able to complete the scheduled on-orbit repairing mission. After completing all on-orbit service tasks, in order not to affect the regular operation of the repaired targets, SSC needs to perform orbital maneuver, away from the last target slowly, and arrive at a parking orbit waiting for the next mission. The cost of orbit maneuver of this process will be ignored. The optimal on-orbit repairing strategy needs to determine the repairing route of each SSC, and the orbital maneuver strategy of each on-orbit repairing process, the velocity increment cost of all SSC is minimized.

Fig. 1 gives a feasible solution to an instance of a many-to-many on-orbit scenario with 6 target satellites and 2 SSC. In this solution, the two SSC will start from different initial orbit and dock with 3 target satellites in order. The SSC #1 will perform an on-orbit repairing for target satellites #1, #2 and #5 in order. The SSC #2 will perform an on-orbit repairing for target satellites #6, #3 and #4 in order. After completing the repairing mission, those two SSC will go to the parking orbit slowly. The mission deadline constraint and maximum velocity increment constraint mentioned above will be satisfied during the whole procedure. Consequently, all target satellites are repaired exactly once by only one SSC.

2.2. Time-dependent orbital rendezvous strategy

The SSC needs a long-distance orbital maneuver to reach each target, then the SSC can dock with the target and perform an on-orbit repairing mission. In this paper, only the Δv cost of the long-distance orbital maneuver process is considered, and the Δv cost of the docking process is ignored.

The SSC need to perform a maneuver procedure to let the SSC and target have the same inclination, RAAN and true anomaly to perform the docking process. This procedure can be achieved by planar maneuver and phasing maneuver. According to (Shi et al., 2010; Friesen et al., 1992), the perturbation effect can be ignored on GEO within one year and the repairing mission usually needs to be completed in a short time (more detailed analysis can be seen in A). Therefore, the two-body dynamics is used in most OOS mission planning research for GEO

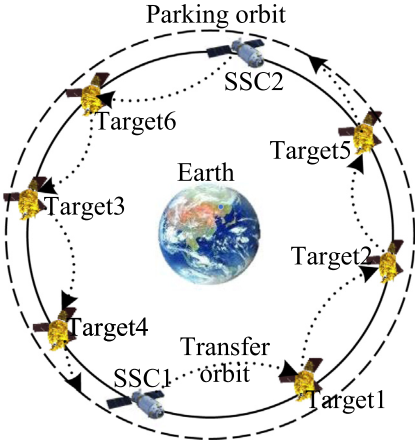


Fig. 1. Many-to-many on-orbit repairing scenario.

satellites (Yu et al., 2014; Zhou et al., 2015b; Zhang et al., 2019). In this paper, we also use the simplified dynamics to calculate the Δv cost for each maneuver leg.

2.2.1. Planar change

Planar change lets the SSC run on the same orbital plane with the target through one impulse maneuver. As shown in Fig. 2, SSC can only maneuver at the intersection of two orbital planes. Therefore, SSC has two chances of plane change maneuver in one orbital period. The dihedral angle α between two orbital planes is given by

$$\cos\alpha = \sin I_s \sin I_t \cos(\Omega_s - \Omega_t) + \cos I_s \cos I_t \tag{1}$$

where $I_s, I_t, \Omega_s, \Omega_t$ represent the inclination and RAAN of the SSC and target, respectively. the intersection points of the two orbital planes can be calculated by

$$\begin{cases} \boldsymbol{r}_{m1} = a\frac{\mathbf{h_s}\times\mathbf{h_t}}{||\mathbf{h_s}\times\mathbf{h_t}||} \\ \boldsymbol{r}_{m2} = -\boldsymbol{r}_{m1} \end{cases} \tag{2}$$

where a is the orbit radius of GEO orbit, $\boldsymbol{h}_s$, $\boldsymbol{h}_t$ represent the angular momentum of SSC and target, respectively. The angular momentum of the orbit $\boldsymbol{h}$ can be calculated by

$$\boldsymbol{h} = \sqrt{\mu a}\begin{bmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3}$$

where μ is the Earth's gravitational constant, $\boldsymbol{h}$ is constant without considering any perturbation.

Since SSC and target are located on the circular GEO orbit, it is also necessary to ensure that the velocity remains unchanged after the impulse maneuver, only the velocity direction will be changed. We define $\Delta\boldsymbol{v}_1$ as an orbital plane change impulse maneuver of α, it can be calculated by:

$$\begin{aligned} \Delta\boldsymbol{v}_1 &= \boldsymbol{v}_t - \boldsymbol{v}_s \\ ||\Delta \mathrm{v}_1|| &= 2||\Delta \mathrm{v}_t|| \sin(\alpha/2) \end{aligned} \tag{4}$$

where $\boldsymbol{v}_s, \boldsymbol{v}_t$ is the velocity of SSC and target at the maneuver point, respectively.

As the last repairing mission is finished, the SSC needs to coast for a while on the original orbit until it reaches the nearest plane change maneuver point for the next target. Given a SSC initial state of $\{\boldsymbol{r}_s^0, \boldsymbol{v}_s^0\}$, the coast time before planar change maneuver can be calculated by

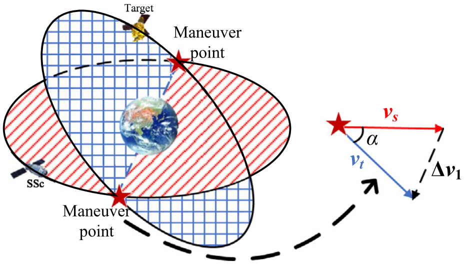


Fig. 2. Sketch of orbital plane change maneuver.

$$t_{coast} = \begin{cases} \arccos\left(\frac{\boldsymbol{r}_{m1}\cdot\boldsymbol{r}_s^0}{||\boldsymbol{r}_{m1}||||\boldsymbol{r}_0||}\right)T_{GEO}/(2\pi), & (\boldsymbol{r}_s^0 \times \boldsymbol{r}_{m1})\cdot\boldsymbol{h}_s > 0 \\ \left(\pi - \arccos\left(\frac{\boldsymbol{r}_{m1}\cdot\boldsymbol{r}_s^0}{||\boldsymbol{r}_{m1}||\left||\boldsymbol{r}_s^0\right||}\right)\right)T_{GEO}/(2\pi), & (\boldsymbol{r}_s^0 \times \boldsymbol{r}_{m1})\cdot\boldsymbol{h}_s < 0 \end{cases} \tag{5}$$

where T_{GEO} is the GEO orbital period.

2.2.2. Phasing maneuver

After the plane change maneuver, the SSC and the target will run in the same circular GEO orbit, so they have a fixed phase difference. We define the phasing angle between the SSC and target as

$$\theta = \begin{cases} \psi, & |\psi| \leqslant \pi \\ -2\pi + |\psi|, & \psi > \pi \\ 2\pi - |\psi|, & \psi < -\pi \end{cases} \tag{6}$$

where $\psi = (\Omega_s + \omega_s) - (\Omega_t + \omega_t)$, ω_s and ω_t denote the true anomaly at the same time of SSC and target, respectively. As shown in Fig. 3, SSC needs to perform an impulse maneuver and then enter the phasing orbit. The phasing orbit is tangent to the original orbit at the maneuver point. After the SSC runs a certain number of turns in the phasing orbit, it returns to the maneuver point. It should be ensured that the target also runs exactly to the tangent point, then the SSC performs an impulse maneuver to meet the target. We know that the SSC can only meet the target at a specific time within one orbital period. So, the phasing time is

$$t_{phase} = \frac{2\pi k_{GEO} + \theta}{2\pi}T_{GEO} \tag{7}$$

where k_{GEO} is target's number of revolutions in phasing maneuver. The semimajor axis of the phasing orbit can be calculated by

$$a_{phase} = r_{GEO}\left(\frac{2\pi k_{GEO} + \theta}{2\pi k_{phase}}\right)^{2/3} \tag{8}$$

where r_{GEO} is the orbit radius of GEO satellite, $k_{\mathbf{phase}}$ denotes chaser's number of revolutions in phasing maneuver. Then the Δv for phasing maneuver can be calculated by

$$\Delta v_{phase} = 2\sqrt{\mu}\left|\sqrt{\frac{2}{r_{GEO}} - \frac{1}{a_{phase}}} - \sqrt{\frac{1}{r_{GEO}}}\right| \tag{9}$$

Inspection of Eq. (8) and Eq. (9) reveals that when k_{GEO} is determined, $k_{phase} = k_{GEO}$ must be satisfied in order to have the minimum Δv_{phase} cost.

Also, we can know that these two impulse maneuvers ($\Delta\boldsymbol{v}_2$ and $\Delta\boldsymbol{v}_3$) for phasing maneuver is in line with the target velocity direction at the maneuver point. Based on Fig. 3, we could know that

$$\begin{cases} \Delta\boldsymbol{v}_2 = \frac{1}{2}\Delta v_{phase}\mathrm{sgn}(\theta)\frac{\boldsymbol{v}_m}{||\boldsymbol{v}_m||} \\ \Delta\boldsymbol{v}_3 = -\Delta\boldsymbol{v}_2 \end{cases} \tag{10}$$

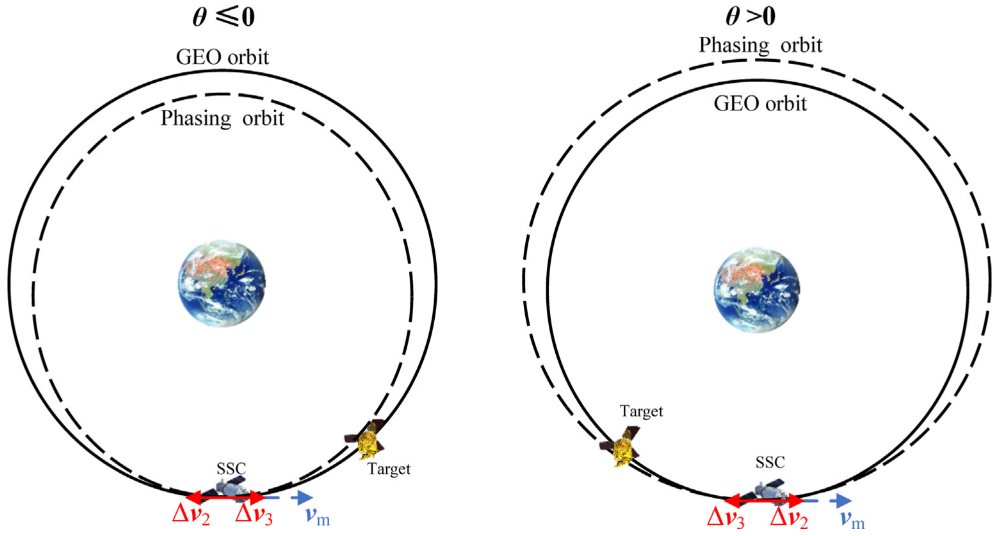


Fig. 3. Sketch of the phasing maneuver.

where $\boldsymbol{v}_{\mathrm{m}}$ is the target velocity at the maneuver point, it can be calculated by the orbital elements of the target.

2.2.3. Mixture of planar change and phasing maneuver

Each SSC and target will keep a constant phase difference because they run in the circular GEO orbit. So, the SSC can perform the phasing maneuver at any time, hence we can set the phasing maneuver point at the plane change point. As shown in Fig. 4, the plane change maneuver impulse $\boldsymbol{\Delta v}_1$ and the first phasing maneuver impulse $\boldsymbol{\Delta v}_2$ can be combined into one impulse maneuver. Based on that, the total Δv cost for one orbital rendezvous can be expressed as

$$\Delta v = ||\boldsymbol{\Delta v}_1 + \boldsymbol{\Delta v}_2|| + ||\boldsymbol{\Delta v}_3|| \tag{11}$$

It is obvious that after the combination of the two strategies, there will be less Δv cost than three impulse maneuver methods. Given a certain phasing revolution number k_{GEO}, the total time of rendezvous maneuver can be calculated by

$$t_{\mathrm{r}} = t_{\mathrm{coast}} + t_{\mathrm{phase}} \tag{12}$$

2.3. Mathematical model for the mission planning problem

In general, the mission planning problem is formulated as a combinatorial-continuous problem when selecting the servicing route and the orbital rendezvous time of each SSC as the decision variable. The rendezvous time can be determined by the number of revolutions of phasing maneuver for each rendezvous, which is an integer variable. By selecting the integer variable as the decision variable, the problem can be transformed to an integer programming problem. Therefore, the many-to-many on-orbit repairing mission planning model can be established based on the modeling method of widely studied VRPTW[19] problem. As shown in Fig. 5, all SSC and targets can be defined on a fully directed graph $G = (N, A)$, where N is a set of nodes comprises a subset N_{s} of n SSC initial nodes, a subset N_{t} of n target nodes and a virtual endpoint e. Setting a virtual endpoint can help us easily model and construct constraints. A is the set of edges connected by any two nodes in N. A fleet of SSC will be represented by V consists of several identical SSC with a total maximum velocity increment. Each SSC starts from its corresponding initial node, reaches the target node, and finally reaches the virtual endpoint after completing the on-orbit repairing mission. Define a decision variable x_{ijk} to describe whether each SSC passes through an edge in edge set A, $x_{ijk} = 1$ means that the edge (i, j) is traversed from node i to j by the SSC k. The SSC cannot perform an orbital maneuver between two SSC initial nodes because that is meaningless. Each SSC initial node can only send one SSC, which has the same number as the initial node. All SSC should arrive at the virtual endpoint e after completing the on-orbit repairing mission. The transfer cost between nodes is related to orbital maneuver time and orbi-

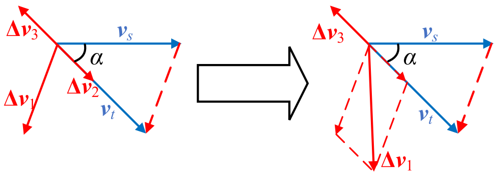


Fig. 4. Mixture of the two maneuver strategies.

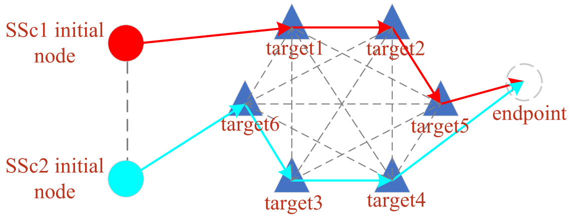


Fig. 5. Many-to-many fully directed graph.

Table 1
Partial parameters used in mission planning models.

Parameter	Definition
s_{ik}	Arrival time of k-th SSC to target node i, $i \in N_t, k \in V$
t_{r}^{ij}	Orbital maneuver time between node i and j, $i, j \in N$
c_{ij}	Δv cost from node i to j, $i, j \in N$
t_{d}^{i}	On-orbit repairing time of target $i, i \in N$
T_{end}	Mission deadline time
$\Delta v_{\max}$	Maximum velocity increment of each SSC
n_{ij}	Number of revolutions in phasing maneuver Orbital elements of node i.
$E_i = \{a_i, I_i, \Omega_i, \omega_i\}$	(a_i: semimajor axis; I_i: inclination; Ω_i: RAAN; ω_i: true anomaly)

tal position. The orbital parameters of each node are fixed, so the orbital maneuver time between nodes will be taken as the decision variable. In this paper, based on the orbital rendezvous strategy described in Section 2.2, it will be replaced by the target's number of revolutions in phasing maneuver. Each target node should be visited and repaired exactly once by a single SSC within the mission deadline. The objective is to determine the optimal repairing sequence and orbital maneuver time of each SSC while minimizing the total Δv cost of all SSC.

The remaining parameters which will be used in the mission planning model are defined in Table 1.

Based on the definition, the mission planning model can be formulated as follows.

(1) Decision variables: $x_{ijk}, n_{ij}, \forall i \in N, j \in N, k \in V$
(2) Objective function:

$$\min: \quad \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \tag{13}$$

(3) Constraints:

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \forall i \in N_{\mathrm{t}} \tag{14}$$

$$\sum_{i \in N} \sum_{j \in N_{\mathrm{s}}} x_{ijk} = 0, \forall k \in V \tag{15}$$

$$\sum_{j \in N} x_{ijk} = 1, \forall k \in V, i \in N_{\mathrm{s}}, k = i \tag{16}$$

$$\sum_{j \in N} x_{ijk} = 0, \forall k \in V, i \in N_{\mathrm{s}}, k \neq i \tag{17}$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \forall k \in V, h \in N_{\mathrm{t}} \tag{18}$$

$$\sum_{i \in N} x_{i,e,k} = 1, \forall k \in V \tag{19}$$

$$c_{ie} = 0, \forall i \in N \tag{20}$$

$$s_{ik} + t_{\mathrm{d}}^{i} + t_{\mathrm{r}}^{ij} - M\left(1 - x_{ijk}\right) \leqslant s_{jk}, \ \forall i \in N, j \in N, k \in V \tag{21}$$

$$0 < s_{ik} + t_{\mathrm{d}}^{i} \leqslant T_{\mathrm{end}}, \forall i \in N, k \in V \tag{22}$$

$$[t_{\mathrm{r}}^{ij}, c_{ij}] = f_{\mathrm{OM}}\left(E_i, E_j, s_{ik}, n_{ij}\right), \ \forall i \in N_{\mathrm{s}} \cup N_{\mathrm{t}}, j \in N_{\mathrm{s}} \cup N_{\mathrm{t}} \tag{23}$$

$$\sum_{i,j \in N} c_{ij} x_{ijk} \leqslant \Delta v_{\max} \ \forall k \in V \tag{24}$$

$$x_{ijk} \in \{0, 1\}, \forall i \in N, j \in N, k \in V \tag{25}$$

$$1 \leqslant n_{ij} < T_{\mathrm{end}}/T_{\mathrm{GEO}}, \ n_{ij} \in \mathbb{Z} \ \forall i \in N, j \in N \tag{26}$$

The objective function (13) is to minimize the total Δv cost of all SSC. Constraint (14) ensures that one target can be repaired only once by exactly one SSC. Constraint (15) ensures the SSC can only start from the SSC initial nodes and cannot maneuver to any SSC initial nodes. Constraint (16) and (17) means that each SSC initial node can only send exactly one SSC with the same number as the initial node. Constraint (18) guarantees the continuity of each repairing route. Constraint (19) ensures that each SSC repairing route will terminate at the virtual endpoint. Constraint (20) indicates that the Δv cost of the edges connecting the endpoint are 0. Constraint (21) is a time constraint, it states that a SSC cannot arrive at node j before $(s_{ik} + t_{\mathrm{d}}^{i} + t_{\mathrm{r}}^{ij})$ if it travels from target i to j. M is a virtual variable that is set to infinity. $M\left(1 - x_{ijk}\right)$ is used to ensure that when the SSC does not pass through a certain route from i to j, the time constraint corresponding to the route is always satisfied. Constraint (22) guarantees that each repair task will end before the mission deadline. Constraint (23) indicates that the orbital maneuver time and Δv cost of any edges in A except those connecting virtual endpoint will be calculated by the nonlinear function f_{OM}. The maximum velocity increment constraints of each SSC are given in (24). Finally, constraint (25) and (26) give the restriction on decision variables.

3. Optimization method

3.1. Design of adaptive genetic algorithm

3.1.1. Encoding mechanism

In the genetic algorithm, the conversion method that converts the feasible solution of the problem from its solution space to the search space that the genetic algorithm can handle is called encoding (Mitchell, 1998; Holland, 1992). Therefore, the coded chromosome in GA procedure has the same meaning with original decision variables. The design of encoding rules plays a vital role in the solving efficiency of genetic algorithm. The sequence coding method is used in this paper for encoding the decision variable x_{ijk}. Assume there is an on-orbit repairing mission scenario consist of m targets and n SSC, to solve the mission planning by genetic algorithm, the chromosome length will first be fixed length to $m + n - 1$. Every chromosome will consist

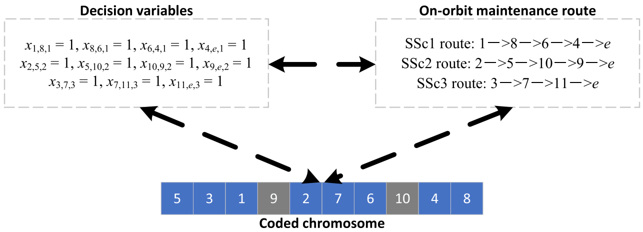

Decision variables
$x_{1,8,1}=1, x_{8,6,1}=1, x_{6,4,1}=1, x_{4,e,1}=1$
$x_{2,5,2}=1, x_{5,10,2}=1, x_{10,9,2}=1, x_{9,e,2}=1$
$x_{3,7,3}=1, x_{7,11,3}=1, x_{11,e,3}=1$
On-orbit maintenance route
SSc1 route: 1—>8—>6—>4—>e
SSc2 route: 2—>5—>10—>9—>e
SSc3 route: 3—>7—>11—>e
5 3 1 9 2 7 6 10 4 8
Coded chromosome

Fig. 6. Examples of coding mechanism.

of a random sequence from 1 to $m+n-1$, so each gene in the chromosome cannot be repeated and it represents the on-orbit repairing route of all SSC. Those genes numbered 1 to m represent the corresponding target number. The genes numbered $m+1$ to $m+n-1$ are split genes, which divide the entire target sequence into n fragments, and the sequence of each fragment corresponds to the on-orbit repairing route of the corresponding SSC. Based on the encoding method, each chromosome will uniquely correspond to a feasible solution of x_{ijk}, but a feasible solution may correspond to multiple coding chromosomes.

Compared the binary encoding mechanism which directly transform the decision variable x_{ijk} to the chromosome, the sequence encoding mechanism has several advantages: (i). The number of decision variable is decreased. (ii). The generated chromosome can always satisfy the route constraints. (iii). The generated chromosome has more specific meaning and is much easier to understand.

Fig. 6 give a feasible example of the coding mechanism. This example consists of 3 SSC and 8 targets, the SSC initial node number is 1 to 3, the target node number is 4 to 11. So, a chromosome of length 10 will be obtained through the encoding mechanism. Since the coded chromosome does not obviously contain the SSC initial node number, the target number in the chromosome will start from 1 to 8 with two split genes numbered 9 and 10. The chromosome is divided into 3 parts by the split genes: 5–3-1, 2–7-6, 4–8. It means that SSC1 will start from SSC initial node 1 and go to the target node 8, 6, 4 in order to perform an on-orbit repairing mission; SSC2 will start from SSC initial node 2 and go to the target node 5, 10, 9 successively; SSC3 will start from SSC initial node 3 and go to the target node 7, 11 in order. After complete the mission, all SSC will go to the parking orbit.

3.1.2. Fitness function

The fitness function is used to measure the degree of conformity of the generated solution, including the objective function value and the satisfaction of constraints. As we mentioned before, the feasible solution of decision variable generated by the sequence coding method will always satisfy the route constraints (14)–(19) and value constraint (25). Therefore, there is no need to measure those constraints in the fitness function. Constraint (20) is independent of any decision variable. It will be used as a known condition when calculating the fitness function. Only constraints (21)–(24) need to be measured by the fitness function. In this paper, the penalty function method is used to measure the satisfaction of constraints. Thus, the fitness function is given by

$$F=\sum_{k\in V}\sum_{i\in N}\sum_{j\in N}c_{ij}x_{ijk}+\varphi p_1(s_{ik})+\gamma p_2(x_{ijk},c_{ij}) \tag{27}$$

The fitness function consists of three parts: the total Δv cost of all SSC, that is, Eq. (13); the penalty function of constraint (22) with a constant coefficient φ; the penalty function of constraint (24) with a constant coefficient γ. p_1 and p_2 are defined as follows:

$$p_1(s_{ik})=\sum_{k\in V}\sum_{i\in N_t}\max\left\{(s_{ik}+t_{\mathrm{d}}^{i}-T_{\mathrm{end}}),0\right\} \tag{28}$$

$$p_2(x_{ijk},n_{ij})=\sum_{k\in V}\max\left\{\left(\sum_{i\in N}\sum_{j\in N}c_{ij}x_{ijk}-\Delta v_{\max}\right),0\right\} \tag{29}$$

Eq. (28) represents the sum of the duration of each mission completion time beyond the mission deadline. Eq. (29) represents the total Δv cost of each SSC that exceeds $\Delta v_{\max}$. Before calculating the fitness function described in Eq. (27)–(29), intermediate variables s_{ik} and c_{ij} need to be calculated by Eq. (21), (23) and the feasible solution decoded from the chromosome. Notice that before calculating the intermediate variables, another decision variable n_{ij} also need to be given. The method of determine the decision variable n_{ij} will be described in the next section.

3.1.3. Genetic operator

(1) selection operatorElite strategy is first used in the selection operator, which selects the chromosome with the highest fitness function to participate in the next genetic operator. Then, the roulette algorithm is used to select other chromosomes from the remaining population. In the roulette algorithm, the cumulative probability is used to judge whether the chromosome is selected, which is given by

$$P_i=\frac{F(i)}{\sum_{j=1}^{\rho}F(j)},\quad i=1,2,\ldots,\rho \tag{30}$$

$$Q_i=\sum_{j=1}^{i}P_i \tag{31}$$

where P_i is the proportion of fitness value of i-th chromosome in the population, ρ is the number of chromosomes in the remain population. Q_i is the cumulative selected probability from the first chromosome to i-th chromosome. The i-th chromosome is selected for the next genetic operator when the generated random number $r\in(0,1)$ satisfies $Q_{i-1}<r<Q_i$.

(2) crossover operatorIn crossover operator, the adaptive crossover probability, which can enhance the iterative efficiency, is used to determine whether the selected two chromosomes perform the crossover operation:

$$P_{\mathrm{c}} = \begin{cases} P_{\mathrm{c1}} - \frac{(P_{\mathrm{c1}} - P_{\mathrm{c2}})(F_i{}' - F_{\mathrm{avg}})}{F_{\max} - F_{\mathrm{avg}}}, & F_i' \geqslant F_{\mathrm{avg}} \\ P_{\mathrm{c1}}, & F_i' < F_{\mathrm{avg}} \end{cases} \tag{32}$$

where F_i' is the larger fitness value of the selected two chromosomes; $F_{\max}$ and F_{avg} denote the maximum fitness value and average fitness value of the population, respectively; P_{c1} and P_{c2} is the upper and lower bound of P_{c}. Partial-Mapped Crossover (Goldberg and Lingle, 1985) is used in the crossover operator. The method exchanges the genes between the two randomly chosen positions in the two selected chromosomes. After that, same genes will be existed in one chromosome, which is not permitted based on the encoding method. Three steps are performed to eliminate this phenomenon: (i). For the exchanged two chromosomes, find the genes in the unexchanged part which hold the same value as the exchanged parts and extract them; (ii). Match the genes extracted from different chromosome in turn; (iii). Exchange the matched genes one by one and put them into the original location. The whole procedure is presented in Fig. 7.

(3) mutation operatorAdaptive mutation probability P_{m} is used to determine whether the selected chromosome is mutated or not. The adaptability of mutation probability can prevent falling into local optima (Wang et al., 2019). P_{m} is given by

$$P_{\mathrm{m}} = \begin{cases} P_{\mathrm{m1}} - \frac{(P_{\mathrm{m1}} - P_{\mathrm{m2}})(F_{\max} - F_i)}{F_{\max} - F_{\mathrm{avg}}}, & F_i \geqslant F_{\mathrm{avg}} \\ P_{\mathrm{m1}}, & F_i < F_{\mathrm{avg}} \end{cases} \tag{33}$$

where $P_{\mathrm{m1}}, P_{\mathrm{m2}}$ is the upper bound and lower bound of P_{m}. As shown in Fig. 8, for the mutated chromosome, two gene sites are randomly selected and their genes are exchanged.

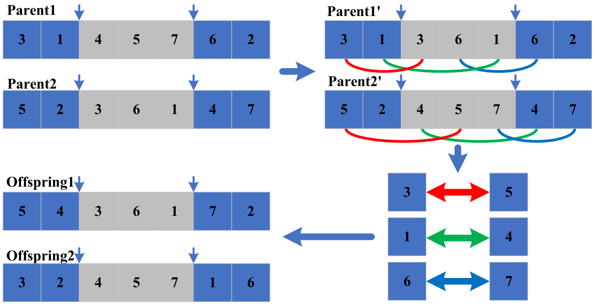


Fig. 7. Crossover operation.

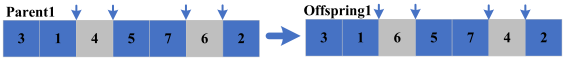


Fig. 8. Mutation operation.

3.2. Heuristics for determining the decision variable n_{ij}

As we can know from Section 2.2, if the repair route of the SSC is determined, the Δv cost for planar change is constant, but the Δv cost for phasing maneuver is time dependent and is determined by n_{ij} in this paper. The orbit information of each target is known and it does not change during the mission period. So, the k-th SSC's each maneuver point for orbital maneuver is fixed in space. Since the required repair time t_{d}^{i} for each target is also fixed, the SSC's each coast time for planar change can be calculated by Eq. (5). Therefore, the total phasing maneuver time for k-th SSC T_{phase}^{k} is given by

$$T_{\mathrm{phase}}^{k} = \sum t_{\mathrm{phase}}^{ij} = T_{\mathrm{end}} - \sum t_{\mathrm{d}}^{j} - \sum t_{\mathrm{coast}}^{ij} \tag{34}$$

where i, j satisfies $\forall i, j \in N,\ x_{ijk} = 1$. t_{phase}^{ij} and t_{coast}^{ij} denote the phasing maneuver time and the coast time before orbital plane change maneuver from target i to j. Once the upper-level algorithm determines the repair route, the determination of n_{ij} for the k-th SSC can be formulated as a nonlinear integer programming problem.

(1) Decision variables: $n_{ij},\ \forall i, j \in N,\ x_{ijk} = 1$.
(2) Objective function:

$$\min: \quad 2\sqrt{\mu} \sum_{\substack{i \in N, j \in N, \\ x_{ijk} = 1}} \left| \sqrt{\frac{2}{r_{\mathrm{GEO}}} - \frac{1}{r_{\mathrm{GEO}} \left[1 + \frac{\theta_{ij}}{2\pi n_{ij}}\right]^{\frac{2}{3}}}} - \sqrt{\frac{1}{r_{\mathrm{GEO}}}} \right| \tag{35}$$

(3) Constraints

$$\sum_{\substack{i \in N, j \in N, \\ x_{ijk} = 1}} \left[T_{\mathrm{GEO}} \left(n_{ij} + \frac{\theta_{ij}}{2\pi} \right) \right] \leqslant T_{\mathrm{phase}}^{k} \tag{36}$$

$$1 \leqslant n_{ij} < T_{\mathrm{phase}}^{k} / T_{\mathrm{GEO}},\ n_{ij} \in \mathbb{N}\ \forall i \in N, j \in N, x_{ijk} = 1 \tag{37}$$

θ_{ij} is the phase angle between node i and j. The objective function (35) minimizes the total phasing maneuver Δv cost of k-th SSC, which is obtained by substituting (8) into (9). Constraints (36) is the mission deadline constraint, it is a variant of (22). Constraints (37) is the value constraints of n_{ij}. Solving this NLIP problem can be complicated and time - consuming. In this paper, a simple and effective heuristic algorithm is proposed to quickly give a near optimal solution.

The pseudocode for the heuristic is shown in Algorithm 1. $card(\cdot)$ means the number of elements in a set. The main idea of the algorithm is to first divide the total phasing maneuver time T^k_{phase} equally according to the orbital period T_{GEO} into m pieces, then distribute those time pieces evenly to each phasing maneuver procedure, and finally allocate the extra time to the phasing maneuver procedure with a largest phase angle difference.

Algorithm 1. Heuristics for determining n_{ij}

```
input: $k$-th SSC repairing route set:$L_k$, phasing angle set:$\Theta_k$
Initialize decision variable $n_{ij}$ of SSC set: $N_l$
Calculate $T^k_{\text{phase}}$ by Eq. (34)
$m_{\text{r}} = \lfloor T^k_{\text{phase}}/T_{\text{GEO}} \rfloor$
for $n_{ij}$ in $N_l$ do
    $n_{ij} = \max\{\lfloor m_{\text{r}}/card(L_k) \rfloor, 1\}$
end for
Calculate SSC's mission end time $t^k_{\text{end}}$ using $L_k$ and $N_l$ by Section 2.2
if $t^k_{\text{end}} < T_{\text{end}}$ then
    $m_{\text{r}} = \lfloor (T_{\text{end}} - t^k_{\text{end}})/T_{\text{GEO}} \rfloor$
    Find target $a$ and $b$ in $L_k$ which have a minimum $\theta_{ab}$ in $\Theta_k$
    $n_{ab} = n_{ab} + m_{\text{r}}$
end if
return $N_l$
```

3.3. Design of large neighborhood search method

3.3.1. LNS framework

The large neighborhood search was first proposed by (Shaw, 1998), its main idea is to find a better solution in the search space by iteratively destroying and repairing the feasible solutions. In LNS algorithm, the neighborhood is implicitly defined by the *destroy* and *repair* method (Pisinger and Ropke, 2010). The pseudocode for the LNS algorithm is shown in Algorithm 2. There are three variables in the pseudocode. The variable x^b is the best solution of the current iteration, x is the feasible solution and x^t is the temporary feasible solution that could be accepted or discarded based on the acceptance rule. Function $destroy(\cdot)$ is used to destruct the feasible solution and return an infeasible solution x^d. Function $repair(\cdot)$ is the repair method that rebuild the destroyed solution, it usually returns a feasible solution built from x^d. More details about the destroy and repair method is proposed in Section 3.3.2 and Section 3.3.3, respectively. $c(x)$ denotes the objective function value of solution x. The acceptance criteria $accept(x^t, x)$ usually have many forms such as simulated annealing, hill-climber, threshold accepting, record-to-record travel. In this paper, the hill-climber method will be selected as the acceptance criteria, which only accept an improving solution. The hill-climber is also used in Shaw's original LNS paper.

Algorithm 2. Large Neighborhood Search

```
input: a feasible solution $x$
$x^b = x$
repeat
    $x^t = repair(destroy(x))$
    if $accept(x^t, x)$ then then
        $x = x^t$
    end if
    if $c(x^t) < c(x^b)$ then
        $x^b = x^t$
    end if
    until stop criterion is met
return $x^b$
```

3.3.2. Destroy method

The destroy method is used to remove a fixed number target from the feasible routes. We sequentially remove the targets which are similar to each other. The general idea is to delete some similar targets, because it can make similar targets mixed together, to create a new and perhaps better solution. Suppose the removed targets are very different from each other. In that case, we will not get a better feasible solution when re-inserting the targets, because those removed targets can only be inserted into their original positions or some wrong positions. A relatedness measure $R(i,j)$ is defined to evaluate the similarity of two targets i and j. The lower $R(i,j)$ is, the less related are the two targets. $R(i,j)$ is given by

$$R(i,j) = 1/(C\prime_{ij} + V_{ij}) \tag{38}$$

where V_{ij} is a parameter that evaluates whether target i and j are repaired by the same SSC, $V_{ij}$is given by

$$V_{ij} = \begin{cases} 0, & \sum\limits_{k \in V} x_{ijk} = 1 \\ 1, & \sum\limits_{k \in V} x_{ijk} = 0 \end{cases} \tag{39}$$

$C\prime_{ij}$ is a normalized parameter that approximately evaluates the cost of getting to target j from i. $C\prime_{ij}$is defined as follows

$$C\prime_{ij} = C_{ij}/\max\{C_{mn}\}, \ \forall m \in N_{\text{t}}, \ \ n \in N_{\text{t}} \tag{40}$$

In (Shaw, 1998), C_{ij} denotes the distance between the two nodes for the classical VRP problem. It approximately represents the transfer cost of the vehicle between two nodes. In this paper, the distance between the two target satellites changes over time and the transfer cost is also time-dependent according to Section 2.2, so we use the orbital difference to approximately evaluate the transfer cost between the two targets. C_{ij} is given by

$$C_{ij} = \beta|\alpha_{ij}| + (1-\beta)\theta_{ij}, \ \ 0 < \beta < 1 \tag{41}$$

where α_{ij} is the dihedral angle between targets i and j, θ_{ij} denotes the phase difference between the two targets. $\beta$is a determinism parameter used to adjust the weight of orbital plane difference and phase difference. Generally, the longer the mission period, the larger the parameter βset, because when the maneuver time increase the phasing maneuver Δv cost will be decreased, but the Δv cost of orbital plane change is time-independent.

The pseudocode for the remove procedure is shown in Algorithm 3. The procedure initially chooses a random target to remove. Then it runs into an iteration loop for selecting the targets similar to the latest removal target to remove until the removed target number exceeds the maximum remove rate. Determinism parameter $p \geqslant 1$ is used to add the randomness of the target removal process (Ropke and Pisinger, 2006) (a lower value of p corresponds to more randomness).

Algorithm 3. Remove Procedure

```
input: a feasible solution $x$, remove rate $q$
Initialize removal set: $x^r = \varnothing$, destroyed set:$x^d = x$
Randomly remove a target $t^r$ from $x^d$
$x^r = x^r \cup t^r$
repeat
    Calculate $R(t^r, t), \forall t \in x^d$ set: $L =$ sort descending $t$
      based on $R(t^r, t), \forall t \in x^d$
    Choose a random number $y$ from the interval $[0, 1)$
    $t^r$= select the $\lfloor y^p \rfloor$-th target from $L$
    $x^r = x^r \cup t^r$
until $card(x^r) \geqslant card(x) \cdot q$
return $x^r, x^d$
```

3.3.3. Repair method

We present the repair method based on the farthest insertion heuristic. The core idea of the algorithm is to first insert the removed targets with the highest "insertion cost". The "insertion cost" of the removed target i is defined as

$$c_i = \min\{\Delta f_{i,k}\}, \ \forall k \in V \tag{42}$$

where $\Delta f_{i,k}$ denotes the objective value change caused by inserting the removed target i into $k-$th SSC route at the position that increases the objective value the least. If all the constraints cannot be satisfied when inserting target i into $k-$th SSC route, the value of $\Delta f_{i,k}$ is set to infinity. As shown in Fig. 9, the "insertion cost" means the minimum objective value increment of inserting target i at its best position overall. We select the target with the largest insertion cost and insert it into the minimum cost position. Repeat the above operations until all the removed targets are inserted into the original broken routes. The pseudocode is shown in Algorithm 4.

Note that if a greedy strategy is adopted, the target with the lowest insertion cost should be inserted first, but this may cause the target with a larger insertion cost to be difficult to find a feasible insert position in the end of the insert procedure due to the violation of the mission deadline constraint or the SSC velocity increment constraint. That will reduce the efficiency of the re-insertion algorithm. Therefore, we present the farthest insertion heuristic to first insert the target that is more difficult to insert to ensure that the re-insertion algorithm can generate a feasible solution to the greatest extent.

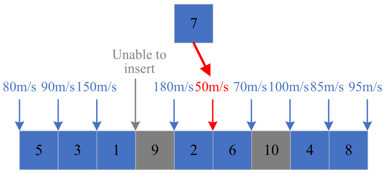


Fig. 9. Explanation of "insertion cost".

Algorithm 4. Re-insert procedure

```
input: removal set: $x^r$, destroyed set: $x^d$
Initialize repaired set: $x^t = x^d$
repeat
    for $t_i^r$ in $x^r$ do
        Calculate $\Delta f_{i,k}$ of $t_i^r$ based on $x^t$
        Find the minimum cost position $P_{i,k}$ of $t_i^r$ in $x^t$
    end for
    Find $t_i^r$ in $x^r$ with maximum $\Delta f_{i,k}$
    Insert $t_i^r$ into $P_{i,k}$ in $x^t$
    Remove $t_i^r$ from $x^r$
until $x^r == \varnothing$
return $x^t$
```

3.3.4. Framework of the LNS-AGA

The flowchart of the proposed LNS-AGA algorithm is presented in Fig. 10. We know that GA has a great performance on global search, but it does not employ a local search mechanism. In order to make up for this defect, a LNS procedure is added after the GA operator. In each iteration process of LNS-AGA, after performing the GA operation, the algorithm will select a fixed number of ϵ chromosomes with highest fitness value and input them into the LNS process. Those selected chromosomes will repeatedly perform the remove operator and re-insert operator until the fitness value is improved or the LNS iteration number exceeds the set maximum number of iterations. Based on the combination of GA and LNS, the algorithm will have not only great global search performance but also have strong local optimization ability, this will improve the algorithm's convergence speed to optimal and near optimal solutions.

4. Numerical simulations and discussion

The simulation experiment is divided into two parts. The first part creates a real-world scenario by choosing

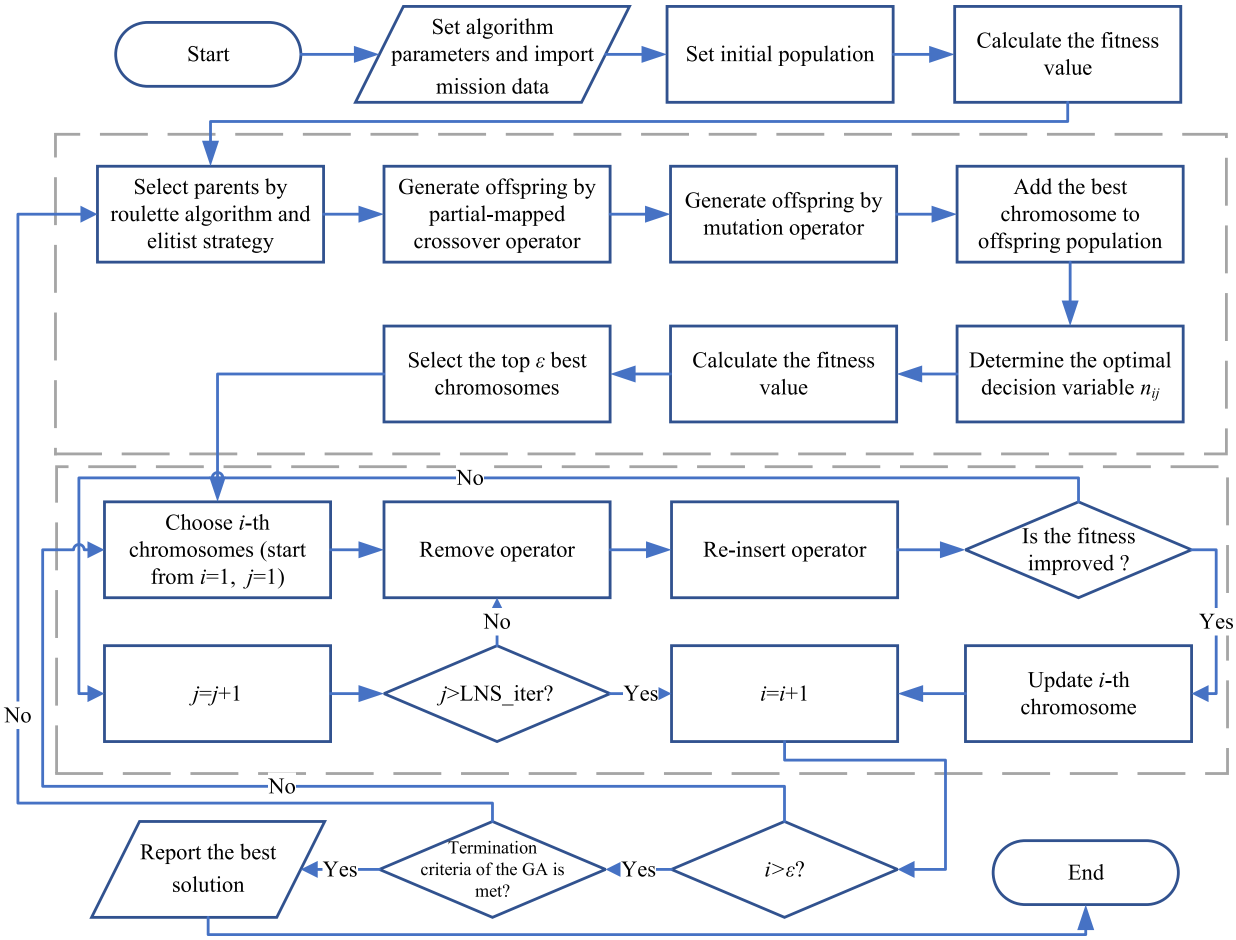

Start
Set algorithm parameters and import mission data
Set initial population
Calculate the fitness value
Select parents by roulette algorithm and elitist strategy
Generate offspring by partial-mapped crossover operator
Generate offspring by mutation operator
Add the best chromosome to offspring population
Determine the optimal decision variable n_{ij}
Calculate the fitness value
Select the top ε best chromosomes
No
Choose i-th chromosomes (start from i=1, j=1)
Remove operator
Re-insert operator
Is the fitness improved ?
Yes
No
j=j+1
j>LNS_iter?
Yes
i=i+1
Update i-th chromosome
No
No
Report the best solution
Yes
Termination criteria of the GA is met?
Yes
i>ε?
End

Fig. 10. Flowchart of the proposed LNS-AGA.

14 Chinese already launched GEO satellites as the target satellite and 2 SSC are randomly created. This scenario is used to verify the effectiveness of the proposed algorithm and determine the best parameters. The second part is used to further testing the performance of the LNS-AGA and make a comparison with other algorithms. In this part, several scenarios with different mission deadline settings and different problem scales are randomly generated. All the algorithms are coded in the MALTAB 2018b environment and tested on a PC with 2.6 GHz Intel Core i5 CPU and 8 GB RAM.

Table 2
Orbit parameter of SSC and target satellites.

ID	Name	Inclination	RAAN	True Anomaly
-1	SSC1	0	0	0
-2	SSC2	5°	0	160°
1	Beidou2_G7	1.60°	66.76°	278.27°
2	Beidou2_G8	0.30°	328.08°	156.03°
3	Beidou_G1	1.80°	45.11°	252.16°
4	Beidou_G2	7.77°	52.63°	328.00°
5	Beidou_G3	1.89°	52.10°	274.21°
6	Beidou_G4	1.06°	59.65°	144.68°
7	Beidou_G5	1.45°	67.40°	288.52°
8	Beidou_G6	1.86°	85.65°	319.30°
9	Chinasat_11	0.09°	103.25°	331.94°
10	Fengyun_2E	5.00°	68.04°	285.07°
11	Fengyun_2F	2.80°	83.11°	224.48°
12	Tianlian1_01	4.81°	71.74°	337.75°
13	Tianlian1_02	2.21°	74.98°	229.24°
14	Tianlian1_03	0.99°	98.18°	230.86°

4.1. Case study

A real-world scenario is firstly established in this part. This scenario chooses 14 already launched Chinese GEO satellites as the target satellite including the famous Beidou series, Fengyun series and Tianlian series. Those target satellites are all run on the GEO with different orbit plane and phase. The two SSC are set randomly with different inclination and true anomaly. The detailed orbit parameter is shown in Table 2. Every SSC can produce a maximum total speed increment of 1000 m/s. The mission scenario starts on 12 March 2021 04:00:00 UTCG, and the true anomaly of all SSC and targets in Table 2 is exactly related to this moment. Every targets' mission deadline is set to 11 April 2021 04:00:00 UTCG, so the mission duration is exactly 30 days. The on-orbit repairing time for each target is set to 20 h. The parameters for LNS-AGA are set as follows: the minimum iteration number is set to 100, the population size is set to 200, the crossover probability adaptively changes from 0.7 to 0.9, the mutation probability adaptively changes from 0.01 to 0.2, the remove rate for Destroy method is 30%, the iteration number of LNS procedure is fixed to 2. The penalty coefficient φ, γ are set to 100 and 10, respectively. The stopping criteria consist of two parts: the iteration number must exceed the minimum iteration number and the fitness value must have no change in the latest 50 iterations. Noticed that the parameter of the GA part is determined by conducting several experiments and the parameter of LNS part is determined based on (Ropke and Pisinger, 2006; Shaw, 1998).

Besides, the parameter ϵ connecting the AGA part and LNS part in Fig. 10 which represents the gap rate between GA and LNS needs to be determined further. We can find the best parameter value with different options of ϵ. Thus, we use the generated real-world to find the best parameter. Then, the computational results are obtained in 10 executions when different numbers of ϵ are tested. The results are reported in Fig. 11 with a box-plot form. From Fig. 11 we can see that when the gap rate between GA and LNS chooses 10% or 15% we can get better performance of the algorithm, by considering the computational time, we choose 10% as the parameter's best value. After all

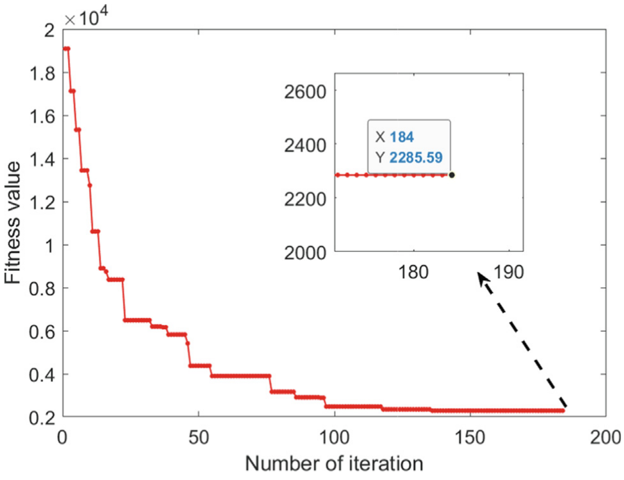


Fig. 13. Convergence curve of the general GA.

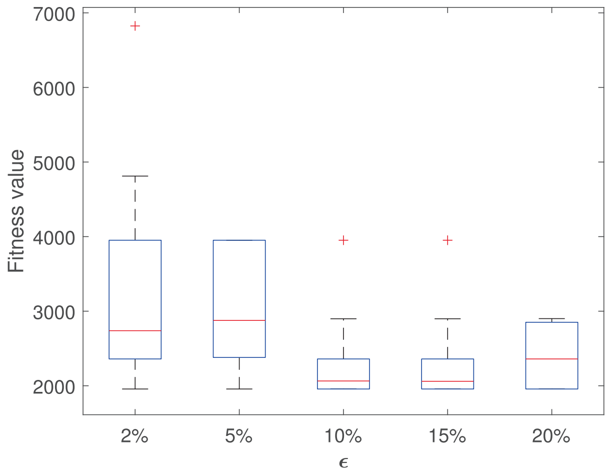
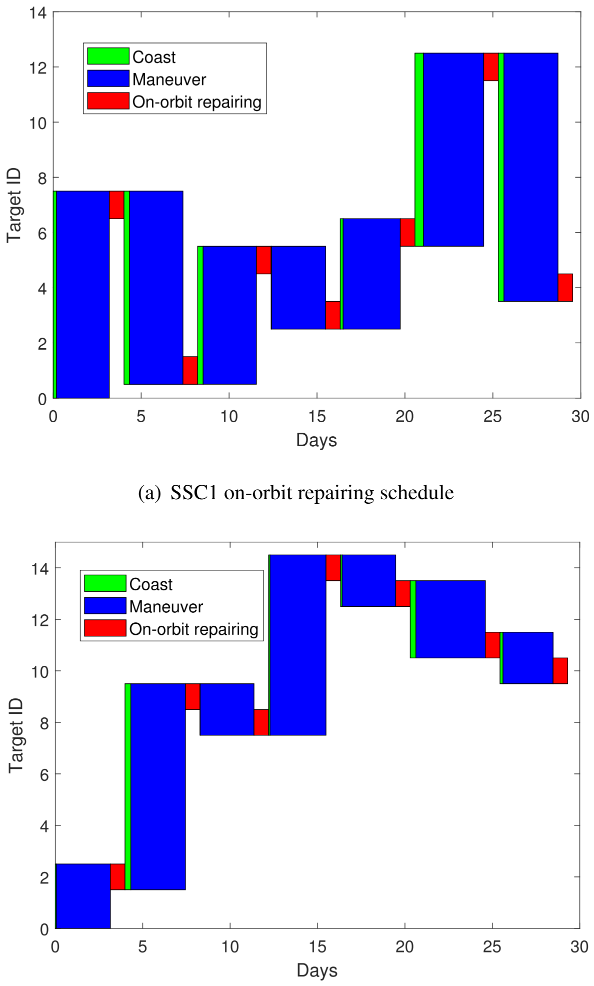


Fig. 11. The effect of gap rate between AGA and LNS.



(a) SSC1 on-orbit repairing schedule

(b) SSC2 on-orbit repairing schedule

Fig. 14. On-orbit repairing order obtained by LNS-AGA.

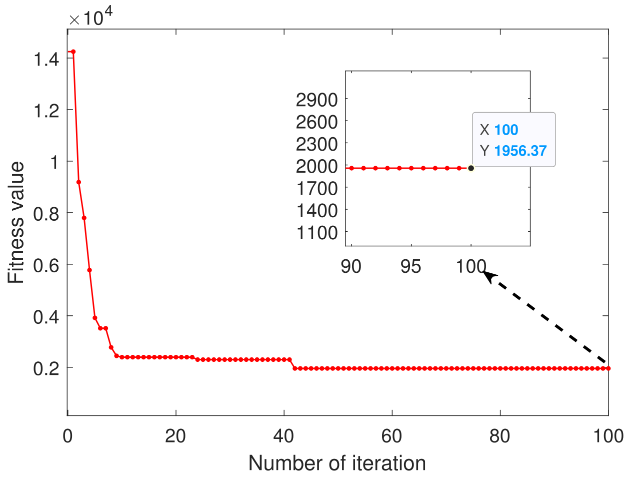


Fig. 12. Convergence curve of the LNS-AGA.

Table 3
Detailed mission planning results obtained by LNS-AGA.

SSC name	Target name	$\Delta \boldsymbol{v}_1$ (m/s)	$\Delta \boldsymbol{v}_2$ (m/s)	$\Delta \boldsymbol{v}_1$ epoch (MJD)	$\Delta \boldsymbol{v}_2$ epoch (MJD)	Coast time (h)	Maneuver time (h)	Δv cost (m/s)
SSC1	Beidou_G5	[−2.63,1.09,78.16]	[2.79,−1.16,−0.076]	2459285.8528	2459288.8563	4.4817	72.0792	81.25
	Beidou2_G7	[−11.25,6.42,8.30]	[11.88,−6.75,−0.38]	2459290.0076	2459293.0431	7.6356	72.8389	29.07
	Beidou_G3	[−0.294,−9.80,−28.79]	[−0.50,11.47,0.246]	2459294.1917	2459297.2229	7.5783	72.7472	41.90
	Beidou_G1	[−23.81,−11.2,−12.87]	[23.49,11.46,−0.26]	2459298.0826	2459301.1549	0.6244	73.7397	55.42
	Beidou_G4	[35.96,−72.94,42.45]	[−35.47,72.97,1.25]	2459302.1528	2459305.4021	3.9419	77.9881	172.89
	Tianlian1_01	[−113.50,29.0,213.32]	[133.54,−35.42,−11.6]	2459306.7229	2459310.1437	11.696	82.1136	382.04
	Beidou_G2	[−4.36,−17.1,−194.6]	[−15.29,30.90,4.21]	2459311.2972	2459314.3687	7.6733	73.7175	230.11
SSC2	Beidou2_G8	[0.763,−44.63,254.0]	[−1.18,34.74,0.15]	2459285.7271	2459288.8188	1.4597	74.1964	292.70
	Chinasat_11	[−30.81,−34.43,20.27]	[29.14,32.55,−0.057]	2459289.9854	2459293.1125	7.9978	75.06	94.15
	Beidou_G6	[−26.03,2.43,96.00]	[26.63,−2.44,−0.86]	2459293.9556	2459297.0312	0.2228	73.8189	126.26
	Tianlian1_03	[−63.08,20.20,−47.93]	[62.69,−20.47,−1.03]	2459297.9479	2459301.1507	2.0019	76.86	147.73
	Tianlian1_02	[−15.84,10.83,73.35]	[16.41,−10.23,−0.71]	2459302.0757	2459305.1361	2.2022	73.4617	95.18
	Fengyun_2F	[−7.37,−1.80,−36.85]	[3.44,1.25,−0.16]	2459306.2903	2459310.2701	7.6892	95.5161	41.30
	Fengyun_2E	[−34.71,24.28,−126.3]	[25.43,−21.05,−2.75]	2459311.2757	2459314.1389	4.1389	68.7189	166.36

the parameter is determined, the mission results for the real-world scenario are obtained by executing the algorithm 20 times. In addition, the problem is also solved by a general GA, which has the same parameter values and stopping criteria which LNS-AGA. The experimental results are shown in Fig. 11,12,12.

Figs. 12 and 13 show one of the best iteration curves in 20 executions of LNS-AGA and general GA, respectively. We can that LNS-AGA converges to the optimal fitness value 1958.36 in about 50 generations while the general GA takes nearly 140 generations converges to an infeasible solution with 2285.59 fitness value. Those two figures indicate that the LNS-AGA has a faster converge rate and can get to a better solution than the general GA. Fig. 14 shows the on-orbit repairing order of the two SSC, we can see that all the targets have been repaired. All the on-orbit repair mission is achieved within the mission deadline time. Table 3 gives detailed mission results obtained by LNS-AGA, including the on-orbit repair of each SSC, the impulse vector in ECI for each orbit maneuver that can directly send to the SSC for mission execution. As a comparison, the mission planning results in (Zhou et al., 2015b; Zhang et al., 2019) only gives a refueling route of each SSC, which needs to be detailed further for real SSC mission execution. From Table 3 we can also see that the SSC1 used total cost of 992.6817 m/s and the SSC2 used total cost of 963.6836 m/s, the two SSC both satisfies the maximum total velocity increment constraint. The above experimental results fully demonstrate the effectiveness of LNS-AGA in a real-world mission.

Table 4
Orbit parameter of 5 SSC.

ID	Name	Inclination	RAAN	True anomaly
1	SSC1	0	120°	30°
2	SSC2	2°	80°	80
3	SSC3	4°	50°	15°
4	SSC4	5°	0	0
5	SSC5	7°	240°	100°

4.2. Comparison with other algorithms

For further testing the performance of the LNS-AGA, firstly, we proposed a randomly generated scenarios. The scenarios consist of 60 targets and 5 SSC. The orbit inclination, RAAN and true anomaly of the target satellites are randomly generated in three certain ranges: 0–10°, 0–180° and 0–360°, respectively. The orbit parameters of 5 SSC are listed in Table 4 The maximum total speed increment constraint of each SSC increases to 2300 m/s and the on-orbit repairing time for each target is set to 20 h. Three algorithms are used to compare with LNS-AGA: conventional GA (Zhou et al., 2015a), Lambert-GA (a general GA combined the Lambert algorithm which is used in (Daneshjou et al., 2017) for calculating the orbit maneuver trajectory), ant colony algorithm with local search procedure (LS-ACO) (Zhang et al., 2019). Eq. (27) is selected as the cost function of the 4 algorithms. The parameters of LNS-AGA and other two GA are same as Section 4.1. The main parameter of LS-ACO is set as follows: number

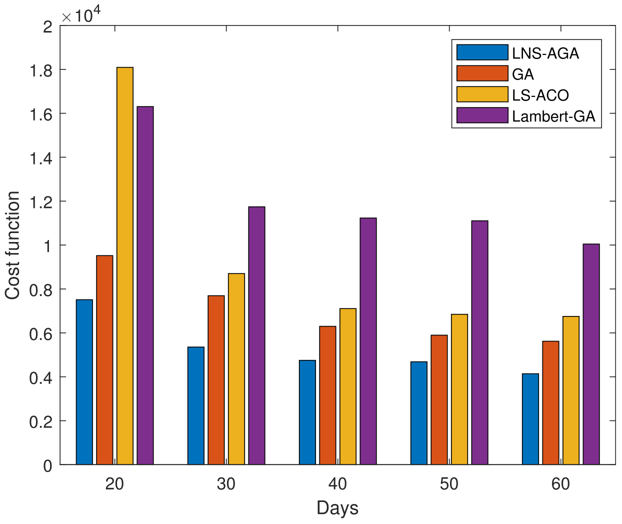
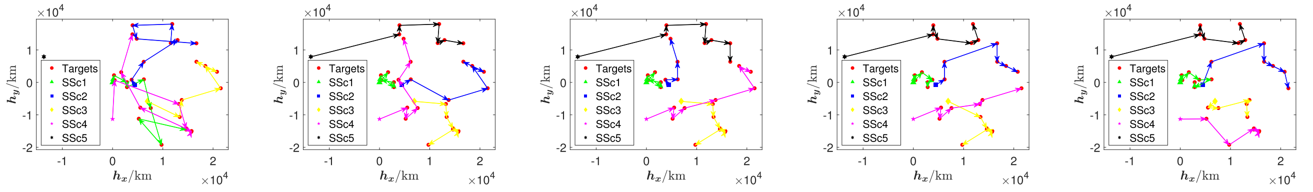


Fig. 15. Comparison of 4 algorithms' best solution in different deadline.



(a) 20 days (b) 30 days (c) 40 days (d) 50 days (e) 60 days

Fig. 16. Comparison of optimal repairing route of different deadlines.

Table 5
Computation results for the 4 algorithms with different deadline. σ represents the rate of feasible solution.

Algorithm	Index	Number of days				
		20	30	40	50	60
LNS-AGA	Ave.	7810.1	5705.1	4994.4	4922.0	4519.2
	Min.	7510.9	5354.3	4749.0	4684.0	4138.4
	Std.	293.5	189.2	180.4	177.5	257.6
	σ	100%	100%	100%	100%	100%
GA	Ave.	13389	9610.2	7252.9	7687.0	8473.0
	Min.	9519.2	7693.6	6298.5	5895.1	5619.0
	Std.	4618.4	1812.4	571.7	1448.9	2681.8
	σ	30%	65%	90%	60%	40%
LS-ACO	Ave.	20650	10313	8246.1	7912.4	8180.1
	Min.	18092	8701.4	7106.3	6845.4	6750.1
	Std.	2199.5	946.6	469.8	796.7	760.8
	σ	0%	20%	90%	65%	35%
Lambert- GA	Ave.	17997	16342	16875	14314	12981
	Min.	16306	11741	11231	11105	10042
	Std.	1577.0	3492.0	1876.8	2036.7	1920.2
	σ	0%	0%	0%	15%	15%

of ants is set to 200, minimum iteration number is 100, which holds the same calculation scale with other algorithms. The stopping criteria of the four algorithms are the same as those described in Section 4.1. Note that the second stopping criteria ensures that the iteration won't stop until it converged, . Each algorithm will be tested 20 times in each scenario.

Firstly, the influence of different mission deadlines for 4 algorithms will be tested. 30 randomly-chosen targets and all the 5 SSC will make up the test scenario. The mission deadline will verify from 20 days to 60 days with 10 days interval. The simulation results are shown in Fig. 15 and 16 and Table 5.

Fig. 15 shows that the Δv cost will be decreased slowly with the deadline increasing. In each scenario, the LNS-AGA always get a best solution among the 4 algorithms, this clearly demonstrate the superiority of proposed LNS-AGA. Lambert-GA have the largest cost value in most conditions, this also demonstrates the effectiveness of the mixed orbital maneuver strategy compared with the result of GA. Table 5 gives the more detailed statistical results, we can see that the LNS-AGA can always get a feasible solution and has the minimum standard deviation in each condition. GA and LS-ACO get a maximum rate of feasible solution when the deadlines is 40 days. The Lambert-GA only has several feasible solution when the mission deadlines is larger than 50, this is mainly because the large search space of maneuver time for each route and the instability of the iterative solving process of Lambert algorithm. Fig. 16 shows optimal repairing route in different deadlines obtained by LNS-AGA. Every point of the satellites represents the projection of its orbital angular momentum in the x-y plane of ECI. The distance between different points only reflects the difference of orbit plane (the semimajor axis for each satellite is set to be the same). That means that the greater the distance between two points, the greater Δv consumption for planar change between them. In Fig. 16 (a), the mission duration is set to 20 days, each maneuver leg has little time for phasing maneuver, the main Δv cost is taken by phasing maneuver. Therefore, the optimal route in the x-y plane of ECI is interlaced with others. When the mission duration increases, the Δv proportion of phasing maneuver is reduced, the optimization of the repairing route is mainly to reduce total planar change consumption. Then, the optimal route will have less crossover and looks more reasonable (see Fig. 16 (d), (e)).

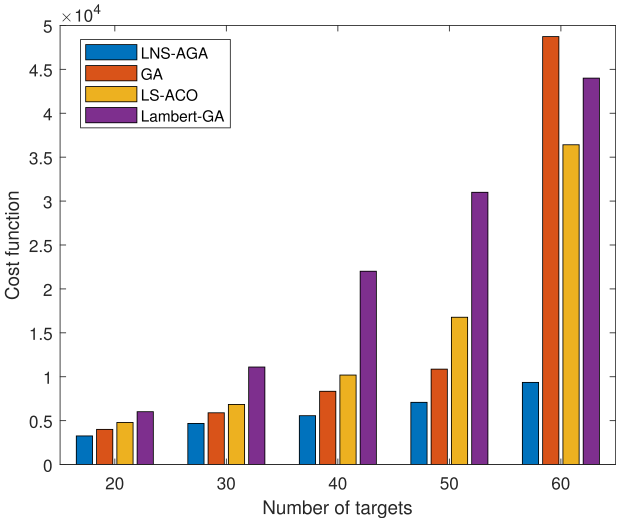


Fig. 17. Comparison of four algorithms' best solution in different target quantity.

Secondly, the influence of different target quantity for 4 algorithms will be tested. The mission deadline is fixed to

Table 6
Computation results for the 4 algorithms with different target number. σ represents the rate of feasible solution.

Algorithm	Index	Number of targets				
		20	30	40	50	60
LNS-AGA	Ave.	3546.4	4922.0	5797.0	7451.2	10094
	Min.	3260.0	4684.0	5565.9	7085.1	9355.6
	Std.	196.2	177.5	175.4	196.9	1089.2
	σ	100%	100%	100%	100%	80%
GA	Ave.	5422.8	7687.0	11225	24495	66782
	Min.	4005.7	5895.1	8342.7	10864	48728
	Std.	1412.1	1448.9	2674.0	9609.8	12721
	σ	40%	60%	50%	25%	0%
LS-ACO	Ave.	5849.9	7912.4	11625	24982	49036
	Min.	4792.4	6845.4	10191	16773	36412
	Std.	654.5	796.7	1005.7	3245.5	6320.2
	σ	70%	65%	20%	0%	0%
Lambert- GA	Ave.	6217.4	14314	26839	33818	50195
	Min.	6020.5	11105	22012	31002	44302
	Std.	1456.4	2036.7	3317.9	1991.7	2186.1
	σ	30%	15%	0%	0%	0%

50 days, and the number of target will verify from 20 to 60. The results are shown in Fig. 17 and Table 6. we can get the similar conclusion according to the simulation results. The LNS-AGA always can get the best solution and the rate of feasible is highest among 4 algorithms. This further illustrates the efficiency of LNS-AGA in solving the many-to-many on-orbit repairing optimization problems.

It is worth mention that the LNS-AGA usually spend more time than other algorithms during the iteration due to the complicated local search procedure. However, for a long-term on-orbit repairing mission, the increase in algorithm running time has little impact on task execution. A more feasible and better solution is usually significant. Because it reduces valuable fuel consumption and increase the service life of SSC. Thus, the LNS-AGA is useful in practice.

5. Conclusion

In this paper, a many-to-many on-orbit repairing mission planning problem is studied. The time-dependent mixed orbital rendezvous strategy combing planar change and phasing maneuver is designed for SSC rendezvous to the target satellite. Based on the orbital rendezvous strategy, the number of revolutions for each phasing maneuver is chosen as decision variables instead of the rendezvous time. Then, the mission planning model can be formulated as an integer programming model referring to the VRPTW model. A hybrid LNS-AGA algorithm combining a large neighborhood search algorithm and adaptive genetic algorithm is then designed to solve the problem. By conducting a real-world scenario consisting of 14 lunched Chinese GEO satellites and 2 SSC, the effectiveness of the algorithm is verified. Further, several comparative experiments indicate that the proposed LNS-AGA can usually get better solutions and has stronger search ability than the selected contrast algorithm, and the orbital rendezvous strategy for GEO spacecraft proposed in this paper is more efficient than the Lambert algorithm. The optimization of the SSC's initial orbit position will be studied and the maximum velocity increment constraint could be considered in the future work.

Declaration of Competing Interest

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

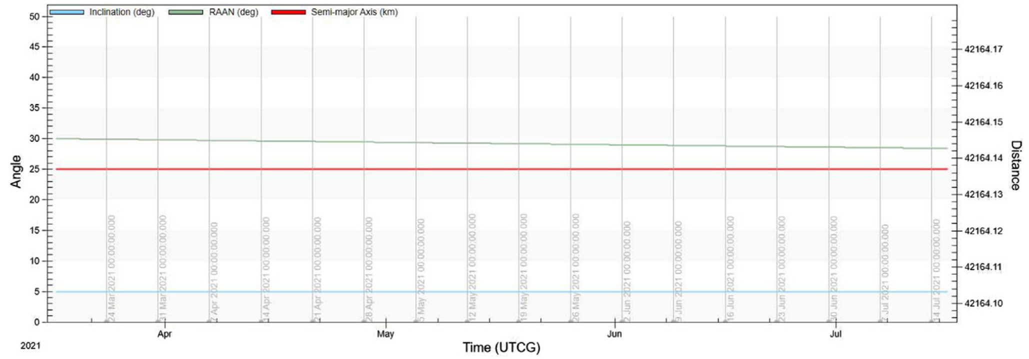
Acknowledgments

This work is supported by the National Natural Science Foundation of China, (No. 61973100, No. 61876050).

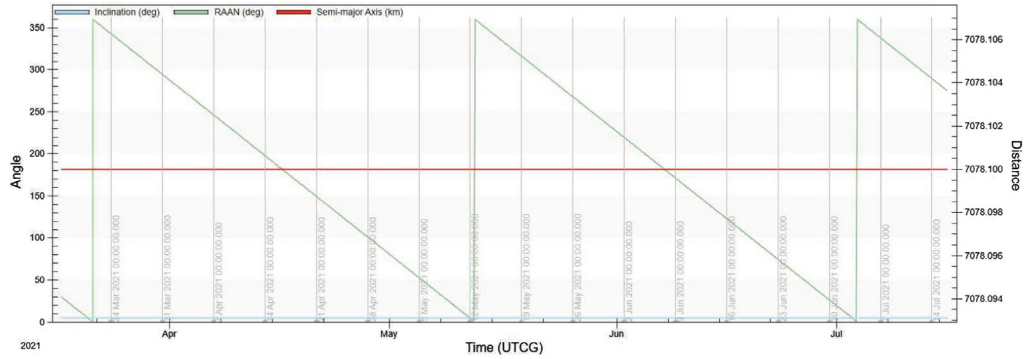
Appendix A. Rationality analysis of simplified two-body dynamics

In general, the satellite's orbit will be affected by Earth's non-spherical gravity, moon and sun gravitational perturbation and solar light pressure, and the most perturbation is caused by Earth's non-spherical gravity (Shi et al., 2010). We give comparative examples of orbit element evolution for GEO satellites and LEO satellites under 2-order Earth's non-spherical perturbation in 60 days in Fig. A.18. (In the simulation example of Section 4, the maximum mission duration is set to 60 days).

We can see from the figure that the semimajor axis and inclination hold constantly under J2 perturbation, the J2 perturbation only effect the RAAN. In 60 days, the RAAN of exampled GEO satellite only decreased 1.6146°, but the RAAN of exampled LEO satellite varies significantly between 0–360°. For a 5 days transfer leg, the RAAN dif-



(a) GEO satellite (semimajor axis=42164.1km; inclination=5°; RAAN=30°; true anomaly=30°)



(b) LEO satellite (semimajor axis=7078.1km; inclination=5°; RAAN=30°; true anomaly=30°)

Fig. A.18. Two examples of orbital element evolution.

ference between two body dynamics and J2 perturbated model is 0.1345°. to fulfill the RAAN difference, the cost will be 6.291 m/s according to Eq. (1) and Eq. (4). That will have a little impact on the final result. As a comparison, the RAAN decrease of LEO satellite is almost 30°, which will need 338.6 m/s to fulfill the gap.

Thus, for GEO on-orbit repairing mission, using the two-body dynamics to calculate the Δv cost of each maneuver leg is reasonable.

References

Bang, J., Ahn, J., 2019. Multitarget rendezvous for active debris removal using multiple spacecraft. J. Spacecr. Rockets 56 (4), 1237–1247.

Baniamerian, A., Bashiri, M., Tavakkoli-Moghaddam, R., 2019. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. Appl. Soft Comput. 75, 441–460.

Chen, H., Ho, K., 2018. Integrated space logistics mission planning and spacecraft design with mixed-integer nonlinear programming. J. Spacecr. Rockets 55 (2), 365–381.

Chen, X.-Q., Yu, J., 2017. Optimal mission planning of geo on-orbit refueling in mixed strategy. Acta Astronaut. 133, 63–72.

Daneshjou, K., Mohammadi-Dehabadi, A., Bakhtiari, M., 2017. Mission planning for on-orbit servicing through multiple servicing satellites: A new approach. Adv. Space Res. 60 (6), 1148–1162.

Danilo, Z., Lorenzo, F., Alessandro, Z., 2021. Preliminary design of multi-chaser active debris removal missions with evolutionary algorithms. In: AAS/AIAA Astrodynamics Specialist Conference, pp. 1–18.

Davis, J.P., Mayberry, J.P., Penn, J.P., 2019. On-orbit servicing: Inspection repair refuel upgrade and assembly of satellites in space. The Aerospace Corporation, report.

Du, B.-X., Zhao, Y., Dutta, A., et al., 2015. Optimal scheduling of multispacecraft refueling based on cooperative maneuver. Adv. Space Res. 55 (12), 2808–2819.

Federici, L., Zavoli, A., Colasurdo, G., 2021. Evolutionary optimization of multirendezvous impulsive trajectories. Int. J. Aerosp. Eng., 21

Friesen, L.J., Jackson IV, A.A., Zook, H.A., et al., 1992. Analysis of orbital perturbations acting on objects in orbits near geosynchronous earth orbit. J. Geophys. Res. 97 (E3), 3845–3863.

Goldberg, D.E., Lingle, R., 1985. Alleles, loci, and the traveling salesman problem. In: Proceedings of an International Conference on Genetic Algorithms and their Applications, vol. 154, Carnegie-Mellon University Pittsburgh, PA, pp. 154–159.

Hastings, D.E., Joppin, C., 2006. On-orbit upgrade and repair: The hubble space telescope example. J. Spacecr. Rockets 43 (3), 614–625.

He, G.-W., Melton, R.G., 2019. Multiple small-satellite salvage mission sequence planning for debris mitigation. Adv. Astronaut. Sci. 171, 4229–4244.

Holland, J.H., 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Sarton du Jonchay, T., Chen, H., Gunasekara, O., et al., 2021. Framework for modeling and optimization of on-orbit servicing operations under demand uncertainties. J. Spacecr. Rockets, 1–17.

Li, C.-Z., Xu, B., 2020. Optimal scheduling of multiple sun-synchronous orbit satellites refueling. Adv. Space Res. 66 (2), 345–358.

Li, H.-Y., Baoyin, H., 2019. Optimization of multiple debris removal missions using an evolving elitist club algorithm. IEEE Trans. Aerosp. Electron. Syst. 56 (1), 773–784.

Li, W.-J., Cheng, D.-Y., Liu, X.-G., et al., 2019. On-orbit service (oos) of spacecraft: A review of engineering developments. Prog. Aerosp. Sci. 108, 32–120.

Mitchell, M., 1998. An introduction to genetic algorithms. MIT press.

Ombuki, B., Ross, B.J., Hanshar, F., 2006. Multi-objective genetic algorithms for vehicle routing problem with time windows. Appl. Intell. 24 (1), 17–30.

Pisinger, D., Ropke, S., 2010. Large neighborhood search. In Handbook of metaheuristics. Springer, pp. 399–419.

Qi, R., Shi, A.-R., Misra, A.K., et al., 2019. Coulomb tether double-pyramid formation, a potential configuration for geostationary satellite collocation. Aerosp. Sci. Technol. 84, 1131–1140.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Sci. 40 (4), 455–472.

Rousso, P., Samsam, S., Chhabra, R., 2021. A mission architecture for on-orbit servicing industrialization. In: 2021 IEEE Aerospace Conference, 50100, IEEE, pp. 1–14.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: International Conference on Principles and Practice of Constraint Programming. Springer, pp. 417–431.

Shen, H.-J., Tsiotras, P., 2002. Optimal scheduling for servicing multiple satellites in a circular constellation. In: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, p. 4907.

Shen, H.-X., Zhang, T.-J., Casalino, L., et al., 2018. Optimization of active debris removal missions with multiple targets. J. Spacecr. Rockets 55 (1), 181–189.

Shi, H.-L., Han, Y.-B., Ma, L.-H. et al., 2010. Beyond life-cycle utilization of geostationary communication satellites in end-of-life. Satellite Communications, Nazzareno Diodato (Ed), Intech, pp. 323–365.

Wang, S., Zhao, L., Cheng, J.-H., et al., 2019. Task scheduling and attitude planning for agile earth observation satellite with intensive tasks. Aerosp. Sci. Technol. 90, 23–33.

Yu, J., Chen, X.-Q., Chen, L.-H., et al., 2014. Optimal scheduling of geo debris removing based on hybrid optimal control theory. Acta Astronaut. 93, 400–409.

Zhang, J., Parks, G.T., Luo, Y.-Z., et al., 2014. Multispacecraft refueling optimization considering the j 2 perturbation and window constraints. J. Guid. Control Dyn. 37 (1), 111–122.

Zhang, T.-J., Yang, Y.-K., Wang, B.-H., et al., 2019. Optimal scheduling for location geosynchronous satellites refueling problem. Acta Astronaut. 163, 264–271.

Zhao, Z., Zhang, J., Li, H.-Y., et al., 2017. Leo cooperative multi-spacecraft refueling mission optimization considering j2 perturbation and target's surplus propellant constraint. Adv. Space Res. 59 (1), 252–262.

Zhou, Y., Yan, Y., Huang, X., et al., 2015a. Mission planning optimization for multiple geosynchronous satellites refueling. Adv. Space Res. 56 (11), 2612–2625.

Zhou, Y., Yan, Y., Huang, X., et al., 2015b. Optimal scheduling of multiple geosynchronous satellites refueling based on a hybrid particle swarm optimizer. Aerosp. Sci. Technol. 47, 125–134.