The Aeronautical Journal (2025), 1–29
doi:10.1017/aer.2024.142

RESEARCH ARTICLE

Optimal scheduling for many-to-many on-orbit service manoeuvers considering variances in target accessibility

J. Zhang[1], H. Xia[1] and L. Li[2]

[1]Space Control and Inertial Technology Research Center, Harbin Institute of Technology, Harbin, China
[2]College of New Energy, Harbin Institute of Technology, Weihai, China
Corresponding author: H. Xia; Email: hxia@hit.edu.cn

Received: 27 March 2024; **Revised:** 20 October 2024; **Accepted:** 14 November 2024

Keywords: optimal scheduling problem; many-to-many on-orbit service; variances in target accessibility; time-dependent coloured traveling salesman problem; enhanced firefly algorithm

Abstract

In this article, we delve into the optimal scheduling challenge for many-to-many on-orbit services, taking into account variations in target accessibility. The scenario assumes that each servicing satellite is equipped with singular or multiple service capabilities, tasked with providing on-orbit services to multiple targets, each characterised by distinct service requirements. The mission's primary objective is to determine the optimal service sequence, orbital transfer duration and on-orbit service time for each servicing satellite, with the ultimate goal of minimising the overall cost. We frame the optimal scheduling dilemma as a time-related colored travelling salesman problem (TRCTSP) and propose an enhanced firefly algorithm (EFA) to address it. Finally, experimental results across various scenarios validate the effectiveness and superiority of the proposed algorithm. The principal contribution of this work lies in the modeling and resolution of the many-to-many on-orbit service challenge, considering accessibility variations — a domain that has, until now, remained unexplored.

Nomenclature

$x = a, e, \Omega, i, \omega, \theta$	the orbital element vector
$t_{i,p}^{initial}, t_{i,p}^{end}$	the initial and end time of the *p*th sub-mission for the *i*th satellite
$t_{i,p}^{lambert}, t_{i,back}^{lambert}$	time required for satellite *i* to transfer orbit
$t_i^{parking}, t_{i,p,k}^{service}$	time required of satellite *i* for parking and on-orbit service
t_i^{end}	the time for servicing satellite *i* to return to the space station
$\delta v_{i,p}$	the change in velocity vector for satellite *i* during the *p*th manoeuver
$\mathbf{S} = \sum_{i=1}^{m} \left\{ \mathbf{S}_i^{trans}, \mathbf{S}_i^{time} \right\}$	optimisation variables
$\mathbf{C}$	total cost
$a_{q,w}^{i}$	the binary variable that elucidate the transfer path of satellite *i*
$\beta_0, \gamma, \bar{\alpha}, N, N_d, M_p$	firefly algorithm parameters
s_i^k, s_j^k	individual fireflies

Abbreviations

OOS	on-orbit service
NP-hard	non-deterministic polynomial hard
TSP	travelling salesman problem
VRP	vehicle routing problem
GA	genetic algorithm
FA	firefly algorithm
ACO	ant colony optimisation

PSO	particle swarm optimisation
LNS	large neighbourhood search
TRCTSP	time-related coloured travelling salesman problem
CTSP	coloured travelling salesman problem
EFA	enhanced firefly algorithm
PSMS	positional substitution mutation strategy
DES	differential evolution strategy
T-PSO	Taguchi-based particle swarm optimisation

1.0 Introduction

In recent years, on-orbit service (OOS) has emerged as a focal point of interest within the aerospace domain, driven largely by its significant economic implications and the promising applications it heralds. Routine on-orbit servicing operations encompass visual inspections, refueling, maintenance and debris removal, among sundry others. Regardless of the OOS mission's type, the servicing satellite must initially traverse from a parking orbit to effectuate a rendezvous with the target. Subsequently, it must sustain proximity to the target until proceeding to the subsequent target or returning as requisite. As space technology advances, the paradigm of OOS has shifted from a one-to-one to a one-to-many and many-to-many configuration. This evolution entails deploying one or more servicing satellites to sequentially deliver services to multiple targets. In the context of one-to-many and many-to-many service modalities, the prudent planning of service sequences can significantly reduce the economic costs associated with missions. Consequently, exploring mission planning and orbital manoeuver strategies under these OOS configurations becomes critically important. Indeed, whether the issue pertains to visual inspections, refueling, maintenance, debris removal or any other form of on-orbit service scheduling optimisation, it fundamentally revolves around optimising the sequence of routes to the servicing satellites. Hence, this investigation transcends the confines of any specific on-orbit service category, delving into the essence of route sequence optimisation for servicing satellites, which constitutes the fundamental challenge in achieving optimal scheduling for on-orbit services.

This optimal scheduling conundrum for route sequences epitomises a quintessential non-deterministic polynomial (NP) hard problem, amenable to modelling as the travelling salesman problem (TSP) [1] and the vehicle routing problem (VRP) [2], among various other related variants. Deterministic methodologies are only feasible for addressing small-scale NP-hard problems. As the complexity of these problems increases, deterministic approaches falter in delivering optimal solutions within limited time. A multitude of meta-heuristic algorithms are employed to address this class of challenges, including the genetic algorithm (GA) [3], the firefly algorithm (FA) [4], particle swarm optimisation (PSO) [5], ant colony optimisation (ACO) [6] and their various adaptations, among others. In the study by Yang [7], an economical manoeuvering strategy for inspecting multiple geosynchronous satellites was proposed, rooted in GA-based optimisation. The research presented in Ref. [8] framed the optimisation problem of multi-orbit routing and scheduling for refuelable space robots in on-orbit servicing as a variant of the VRP, adeptly addressed through the arc creation algorithm. Zhang and Li [9, 10] offered solutions to the challenges of multi-satellite refueling planning in near-circular low-Earth orbit and Sun-synchronous orbit, respectively, employing GA and combinatorial heuristic algorithms. In Ref. [11], a sophisticated two-stage optimisation framework leveraging GA was introduced to concurrently address orbit design and mission scheduling for an on-orbit refueling system operating in sun-synchronous orbit. References [12] and [13] both focused on refining GA to determine the optimal sequence for active debris removal by servicing satellites. Yu [14], tackled the complexities of mission scheduling for active debris removal in low-Earth orbit using PSO, meticulously accounting for communication time-window constraints, terminal state constraints and time distribution constraints. Shen [15], reformulated the optimisation problem of debris swarm removal as an extended TSP and resolved it using ACO. Bang introduced a sophisticated two-stage framework in Ref. [16] to address the multi-objective optimisation problem associated with active debris removal tasks. In Ref. [17], a large neighbourhood search (LNS) adaptive

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

GA was introduced to strategise on-orbit repair tasks, with explicit attention to task duration constraints for many-to-many geosynchronous orbit satellites. Additionally, some research endeavors, such as those in Refs [18, 19], focus on orchestrating the manoeuvers of servicing satellites during OOS operations, rather than optimising scheduling for a specific OOS mission. However, constrained by the choice of orbital manoeuvering strategies, the aforementioned findings limit the flexibility to dictate the temporal expenditure of on-orbit services. While approaches like Hohmann-based orbital manoeuvers excel in fuel efficiency, they restrict the flexibility in configuring the duration of orbital manoeuvers, which can be critical for missions with stringent time constraints.

In the studies conducted by Bakhtiari [20] and Daneshjou [21], both researchers addressed the challenge of scheduling on-orbit service missions while allowing for free determination of orbit change timing. Their approaches involved optimisation frameworks based on PSO and Taguchi-based PSO (T-PSO) combined with the Lambert-based orbit change strategy. It is noteworthy that the models founded on the Lambert manoeuver strategy in Refs. [20, 21] optimise the temporal variable, thereby placing greater demands on the optimisation algorithms, especially when dealing with a large number of service objectives. Unfortunately, the number of service targets in both cases does not exceed ten. Moreover, Refs [20, 21] do not incorporate additional constraints within the optimal scheduling process of on-orbit services.

The prevailing limitation in optimising on-orbit service scheduling encompasses constraints such as fuel limitations, time window constraints and J2 perturbations, as thoroughly examined in Refs [9, 10, 15, 17]. While significant research has been devoted to the optimal scheduling of OOS, we argue that certain critical limitations, such as differences in target access capabilities, have yet to be addressed. Current studies typically assume that each servicing satellite can universally access every target, implying an omnipresent capacity to serve any target. However, this assumption is unrealistic. For example, servicing satellites designed primarily for refueling may lack the capability to perform maintenance services. Even within the same category of on-orbit servicing missions, such as maintenance, varying specific maintenance requirements can lead to scenarios where only particular servicing satellites are equipped to handle specific targets. As space science and technology advance, a likely future scenario involves deploying a constellation of servicing satellites, each with one or more service capabilities, tasked with providing on-orbit services to a diverse array of targets with varying demands. In this complex environment, conventional scheduling models that ignore variations in access capabilities may become inadequate. This scenario highlights the urgent need for more sophisticated scheduling models capable of managing the specific needs of diverse targets while considering the heterogeneous access capabilities of each satellite. Unfortunately, to our knowledge, no existing literature explores the optimal scheduling of on-orbit services with consideration for variations in target access capabilities.

To address this research gap, we present a novel approach for modeling the optimal scheduling of on-orbit services, carefully incorporating variations in target accessibility. This is framed as a time-related coloured travelling salesman problem (TRCTSP), and we propose an enhanced firefly algorithm (EFA) to methodically tackle this complex issue. The key contributions of this paper are succinctly outlined as follows:

(1) The pursuit of optimal scheduling for on-orbit services, accounting for varying target accessibility, is innovatively reframed as a TRCTSP. Unlike the coloured traveling salesman problem (CTSP) [22], the TRCTSP incorporates both time-bounded and time-dependent cost functions.

(2) The TRCTSP model developed in this study significantly surpasses the complexity of the traditional CTSP model, due to the nuances of orbital dynamics and the constraints imposed by target accessibility. To address this challenge, a novel EFA is introduced. This advanced algorithm combines the positional substitution mutation strategy (PSMS) with the differential evolution strategy (DES). Experimental results demonstrate that EFA outperforms GA [3], LNS-GA [17], PSO [5], T-PSO [21] and FA [4] in the given problem scenario.

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

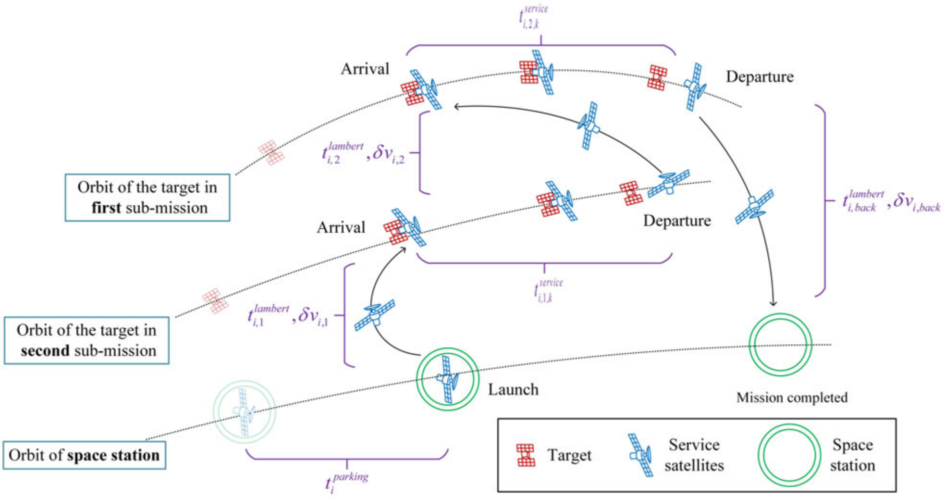
$t_{i,2,k}^{service}$
Arrival
Departure
$t_{i,2}^{lambert}, \delta v_{i,2}$
Orbit of the target in **first** sub-mission
Arrival
Departure
$t_{i,back}^{lambert}, \delta v_{i,back}$
$t_{i,1,k}^{service}$
Orbit of the target in **second** sub-mission
$t_{i,1}^{lambert}, \delta v_{i,1}$
Launch
Mission completed
Orbit of **space station**
$t_i^{parking}$
Target
Service satellites
Space station

Figure 1. *The overview of the on-orbit service mission process.*

The structure of this article is as follows: Section 2 outlines the necessary prerequisites. Section 3 details the optimisation model, while Section 4 explores the algorithmic design. Section 5 presents numerical simulations to validate the proposed approach. Finally, Section 6 offers the conclusion.

2.0 Problem scenario and on-orbit service strategy

In the envisioned mission scenario, m satellites are deployed to perform on-orbit servicing operations at n designated targets. The service categories include visual inspection, refueling, maintenance and debris removal, numbered sequentially as 1, 2, 3 and 4, respectively. All servicing satellites are equipped with the capability for visual inspection. For each servicing satellite i (where $i \leq m$), it possesses one or more of the following service functions in addition to visual inspection: refueling, maintenance or debris removal. Alternatively, servicing satellite i may be equipped solely with visual inspection capabilities. Each designated target j (where $j \leq n$) has a single specific service requirement. Typically, the on-orbit services are carried out in the following phases: The servicing satellite departs from the space station and enters a transfer orbit. It then exits the transfer orbit to rendezvous with the designated target at the servicing point, where it performs the required tasks. Afterward, the satellite proceeds to the next target, completes all servicing operations and eventually returns to the space station.

In Fig. 1, we provide a detailed illustration of the entire mission process, showcasing two on-orbit service operations carried out by satellite i. Satellite i is assigned to perform on-orbit services for P_i targets ($\sum_{i=1}^{m} P_i = n$), necessitating P_i manoeuvers and corresponding service operations. Thus, satellite i is responsible for P_i sub-missions in total. In the context of Fig. 1, $P_i = 2$. The ith servicing satellite co-orbits with the space station within the parking orbit for a time denoted as $t_i^{parking}$. Subsequently, it begins its first sub-mission by manoeuvering to and servicing its initial target. During this phase, the manoeuver time is represented as $t_{i,1}^{lambert}$, the change in velocity vector from the manoeuver is $\delta v_{i,1}$, and the service time for the first target is denoted as $t_{i,1,k}^{service}$, where $k \in \{1, 2, 3, 4\}$ indicates the type of on-orbit service. After completing the first sub-mission, satellite i allocates a time period of $t_{i,2}^{lambert}$ to achieve rendezvous with the next service target. The changes in the velocity vector during this maneuver are represented as $\delta v_{i,2}$. The duration for servicing the second target is denoted as $t_{i,2,k}^{service}$. Once all designated targets have been serviced, satellite i requires $t_{i,back}^{lambert}$ to return to the space station, with velocity vector adjustments documented as $\delta v_{i,back}$. The temporal, orbital state and mass relationships throughout the on-orbit service for the multi-satellite mission are depicted below.

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

2.1 Time relationship

The time instants $t_{i,p}^{initial}$ and $t_{i,p}^{end}$ marking the initial and end of the pth sub-mission for the ith servicing satellite $(1 \leq p \leq P_i)$ are determined by

$$t_{i,p}^{initial} = \begin{cases} t_i^{parking}(p = 1) \\ t_{i,p-1}^{end}(1 < p \leq P_i) \end{cases} \qquad (1)$$
$$t_{i,p}^{end} = t_{i,p}^{initial} + t_{i,p}^{lambert} + t_{i,p,k}^{service}$$

The time instant, t_i^{end}, for servicing satellite i to return to the space station after accomplishing all on-orbit services can be formulated as

$$t_i^{end} = t_{i,P_i}^{end} + t_{i,back}^{lambert}. \qquad (2)$$

2.2 Orbital element vector of a target after time Δt

The dynamics of an object in space can be delineated through the orbital element vector $x = [a, e, \Omega, i, \omega, \theta]$, where a represents the semimajor axis, e signifies the eccentricity, Ω represents the right ascension of the ascending node, i stands for the inclination, ω denotes the argument of perigee, and θ represents the true anomaly. When targets revolve in a specific Keplerian orbit, it is primarily the true anomaly that undergoes variations. Given a target state $x\,(t_0)$ at t_0 and a time interval Δt, we can calculate $\boldsymbol{x}(t_0 + \Delta t)$ as:

$$\boldsymbol{x}(t_0 + \Delta t) = \mathcal{G}(\boldsymbol{x}\,(t_0)\,, \Delta t) \qquad (3)$$

For a detailed derivation, see the Appendix 6.

2.3 Lambert-based manoeuver

Each servicing satellite needs to perform orbital transfers during the mission to rendezvous with the service target. In this study, servicing satellites undergo orbital transfer procedures based on Lambert's theorem. Two manoeuvers are taken into account for each orbital transfer. Provided with the initial position vector $\boldsymbol{r}_1$, final position vector $\boldsymbol{r}_2$, and the transfer time Δt of the transfer orbit, the necessary velocity changes for the two manoeuvers can be computed as:

$$\boldsymbol{v}_1 = \mathcal{L}_1(\boldsymbol{r}_1, \boldsymbol{r}_2, \Delta t)\,, \boldsymbol{v}_2 = \mathcal{L}_2(\boldsymbol{r}_1, \boldsymbol{r}_2, \Delta t) \qquad (4)$$

For a detailed derivation, see the Appendix.

In addition, if the orbital element x of an object at a specific point along an orbit are known, it allows for the determination of the object's position and velocity vectors $\boldsymbol{r}$ and $\boldsymbol{v}$ at that point, and vice versa. We represent this functional relationship as:

$$\boldsymbol{x} = \mathcal{T}_x(\boldsymbol{r}, \boldsymbol{v}), \boldsymbol{r} = \mathcal{T}_r(\boldsymbol{x}), \boldsymbol{v} = \mathcal{T}_v(\boldsymbol{x}). \qquad (5)$$

Suppose that the pth sub-mission of the ith servicing satellite requires a manoeuver to rendezvous with the jth target. The initiation time of this sub-mission is denoted as $t_{i,p}^{initial}$, and the duration of the Lambert-based manoeuver is $t_{i,p}^{lambert}$. The orbital elements for the ith servicing satellite and the jth target at $t_{i,p}^{initial}$ are known and designated as $\boldsymbol{x}_i\left(t_{i,p}^{initial}\right)$ and $\boldsymbol{x}_j\left(t_{i,p}^{initial}\right)$. Combining the previously derived information, the change in velocity vector $\delta v_{i,p}$ required for ith servicing satellite to manoeuver during the pth sub-mission is

$$\begin{aligned} v_{i,p} = {} & \| \mathcal{L}_2\left(\mathcal{T}_r\left(\boldsymbol{x}_i\left(t_{i,p}^{initial}\right)\right), \mathcal{T}_r\left(\mathcal{G}\left(\boldsymbol{x}_j\left(t_{i,p}^{initial}\right), t_{i,p}^{lambert}\right)\right), \Delta t\right) \\ & - \mathcal{T}_v\left(\mathcal{G}\left(\boldsymbol{x}_j\left(t_{i,p}^{initial}\right), t_{i,p}^{lambert}\right)\right) \| + \| \mathcal{T}_v\left(\boldsymbol{x}_i\left(t_{i,p}^{initial}\right)\right) \\ & - \mathcal{L}_1\left(\mathcal{T}_r\left(\boldsymbol{x}_i\left(t_{i,p}^{initial}\right)\right), \mathcal{T}_r\left(\mathcal{G}\left(\boldsymbol{x}_j\left(t_{i,p}^{initial}\right), t_{i,p}^{lambert}\right)\right), \Delta t\right) \| \end{aligned} \qquad (6)$$

Given that the initial mass of the ith servicing spacecraft is $m_{i,0}$ and its mass after completing the pth sub-mission is $m_{i,p}$, the mass of fuel consumed by the ith servicing spacecraft in the pth sub-mission is

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

$$\delta m_{i,p} = m_{i,p-1} - m_{i,p-1} \exp\left(\frac{-\delta v_{i,p}}{g_0 I_{sp}}\right) \tag{7}$$

where g_0 is the sea-level standard acceleration of gravity, and I_{sp} is the specific impulse of thrusters.

3.0 Optimisation model

In alignment with the on-orbit service strategy outlined in the previous section, the multi-objective on-orbit service problem can be framed as a TRCTSP, which falls under the category of NP-hard combinatorial optimisation problems. The coloured travelling salesman problem (CTSP) is defined as follows: each salesman and city is associated with one or more colours, and a city can be visited only once by a salesman of the same colour. The primary objective is to determine the shortest Hamiltonian loop for all salesmen while strictly adhering to the colour constraints. In this analogy, colour represents the type of on-orbit service being performed, salesman signifies the servicing satellite, and city denotes the target being serviced. This section focuses on defining the design variables, outlining the objective function and specifying the constraint conditions.

3.1 Design variables

As previously mentioned, the completed on-orbit service operation comprises m independent service sub-missions, each corresponding to a target sequence $\mathbf{S}_i$, where i represents the serial number of the sub-mission and also the serial number of the corresponding servicing satellite. Each sequence $\mathbf{S}_i$ encompasses two pieces of information, comprising the transfer sequence

$$\mathbf{S}_i^{trans} = \left[s_{i,1}^{trans}, s_{i,2}^{trans}, \cdots, s_{i,p}^{trans}, \cdots, s_{i,P_i}^{trans}\right]$$

and the time sequence

$$\mathbf{S}_i^{time} = \left[t_i^{parking}, t_{i,1}^{lambert}, \cdots, t_{i,P_i}^{lambert}, t_{i,1,k}^{service}, \cdots, t_{i,P_i,k}^{service}, t_{i,back}^{lambert}\right].$$

where $s_{i,p}^{trans}$ signifies the serial number of the target for servicing satellite i in the pth sub-mission. The explanations for all other variables have been provided in the preceding section. Consequently, the optimisation variables for the multi-objective on-orbit service problem can be expressed as:

$$\mathbf{S} = \sum_{i=1}^{m} \mathbf{S}_i = \sum_{i=1}^{m} \left\{\mathbf{S}_i^{trans}, \mathbf{S}_i^{time}\right\}. \tag{8}$$

3.2 Constraint conditions

The transfer sequence $\mathbf{S}_i^{trans}$ is insufficient for detailing the constraints on transfer paths in the TRCTSP model. Consequently, we employ the binary variable $a_{q,w}^i \in \{0, 1\}$ to elucidate the transfer path of the ith servicing satellite and provide the associated constraints. In this context, $q, w \in \{0, 1, 2, \ldots, n\}$, where $\{0\}$ denotes the space station, and $\{1, 2, \ldots, n\}$ denote the serial numbers of the targets.$a_{q,w}^i = 1$ if the ith servicing satellite possesses a direct manoeuvering path between target q and target w; and otherwise, $a_{q,w}^i = 0$. As an illustrative example to elucidate this concept, let's consider the transfer sequence of the 1st servicing satellite $\mathbf{S}_1^{trans} = [2, 1, 5]$. In this case, we have: $a_{0,2}^1 = 1$, $a_{2,1}^1 = 1$, $a_{1,5}^1 = 1$, $a_{5,0}^1 = 1$, and all other relevant binary variables are 0. For the sake of simplicity in our description, we denote $\{1, 2, \cdots, n\}$ as $\langle n\rangle$. The multi-objective on-orbit service problem is governed by the following constraint conditions:

(a) All servicing satellites are required to embark from and return to the space station.

$$\sum_{w=1}^{n} a_{0,w}^i = 1, \forall i \in \langle m\rangle, \forall w \in \langle n\rangle \tag{9}$$

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

$$\sum_{q=1}^{n} a_{q,0}^{i} = 1, \forall i \in \langle m \rangle, \forall q \in \langle n \rangle \tag{10}$$

Allow the service capacity set of the satellite and the service requirement set of the target to be denoted as $c_i^{satellite}$ and c_q^{target}, respectively, with $c_i^{satellite}, c_q^{target} \in \{1, 2, 3, 4\}$. In accordance with the problem scenario outlined in Section 2, each target has a solitary requirement, while each satellite can possess either one or two service capabilities. If $c_q^{target} \in c_i^{satellite}$, it signifies that servicing satellite i possesses the capability to service target q. Subsequently, we can derive the serial number set of targets accessible by the ith satellite as $V_i = \left\{q{:}c_q^{target} \in c_i^{satellite}, q \in 1, 2, \cdots, n\right\}$.

(b) Servicing satellite i cannot access targets for which it lacks the capability to provide service:

$$\sum_{q=1}^{n} \sum_{w=1}^{n} a_{q,w}^{i} = 0, \forall i \in \langle m \rangle, \forall q \in \langle n \rangle \backslash V_i, \forall w \in \langle n \rangle \tag{11}$$

$$\sum_{q=1}^{n} \sum_{w=1}^{n} a_{q,w}^{i} = 0, \forall i \in m, \forall q \in n, \forall w \in n \backslash V_i. \tag{12}$$

(c) Each target must be accessed once and only once:

$$\sum_{q=1}^{n} \sum_{i=1}^{m} a_{q,w}^{i} = 1, \forall i \in \langle m \rangle, \forall q \in \langle n \rangle, \forall w \in \langle n \rangle, q \neq w \tag{13}$$

$$\sum_{=1}^{n} \sum_{i=1}^{m} a_{q,w}^{i} = 1, \forall i \in \langle m \rangle, \forall q \in \langle n \rangle, \forall w \in \langle n \rangle, q \neq w \tag{14}$$

$$\sum_{q=1}^{n} a_{qw}^{i} = \sum_{l=1}^{n} a_{wl}^{i}, q \neq w \neq l, \forall w, q, l \in \langle n \rangle, \forall i \in \langle m \rangle \tag{15}$$

(d) The incorporation of subloops in the transfer path of servicing satellite manoeuvers is strictly forbidden:

$$u_q^i - u_w^i + n \times a_{q,w}^{i} \leq n - 1, \forall q, w \in n, q \neq w, \forall i \in m, \tag{16}$$

where u_q^i represents the count of targets serviced by satellite i from the station to target q.

(e) Considering the restrictions on the fuel capacity of each servicing satellite, the total fuel consumption during manoeuvers will not surpass a maximum value $\delta m_{i,max}$:

$$\delta m_{i,back} + \sum_{p=1}^{P_i} \delta m_{i,p} \leq \delta m_{i,max}, \tag{17}$$

(f) The parking time, manoeuver time, and on-orbit service time for each servicing spacecraft are restricted:

$$\begin{array}{ccc} t_{min}^{final} \leq t_{max} \leq & & t_{max}^{final} \\ t_{i,min}^{lambert} \leq t_{i,p}^{lambert} \leq & & t_{i,max}^{lambert} \\ t_{i,min}^{lambert} \leq t_{i,back}^{lambert} \leq & & t_{i,max}^{lambert} \\ t_{i,min}^{parking} \leq t_{i}^{parking} \leq & & t_{i,max}^{parking} \\ t_{i,p,k,min}^{service} \leq t_{i,p,k}^{service} \leq & & t_{i,p,k,max}^{service} \end{array} \tag{18}$$

where t^{final}_{min}, t^{final}_{max}, $t^{lambert}_{i,min}$, $t^{lambert}_{i,max}$, $t^{rking}_{i,min}$, $t^{parking}_{i,max}$, $t^{service}_{i,p,k,min}$, and $t^{service}_{i,p,k,max}$ are known constants. If $t^{service}_{i,p,k}$ exceeds $t^{service}_{i,p,k,min}$, it indicates that the ith satellite will park in the target orbit upon accomplishing its ith service target sub-mission, with a parking duration of $t^{service}_{i,p,k} - t^{service}_{i,p,k,min}$.

3.3 The cost function

This multi-objective optimisation function is normalised prior to summation. The total cost **C** of the campaign is defined as:

$$\mathbf{C} = \kappa \frac{t_{max} - t^{final}_{min}}{t^{final}_{max} - t^{final}_{min}} + (1-\kappa) \frac{\sum_{i=1}^{m} \left(\delta m_{i,back} + \sum_{p=1}^{P_i} \delta m_{i,p} \right)}{m \delta m_{i,max}} + \varepsilon n_{violate} \tag{19}$$

where $t_{max} = max\left(t^{end}_{1,P_1}, t^{end}_{2,P_2}, \cdots, t^{end}_{i,P_i}, \cdots, t^{end}_{m,P_m}\right)$ represents the duration required to accomplish the entire task, κ denotes the weight parameter, ε is the penalty coefficient, and $n_{violate}$ represents the number of targets in the planned path that violate the reachability variances restriction. The penalty coefficient ε should take a value much larger than the normalised multi-objective function to ensure that the relevant constraints are achieved. δm_{min} is the minimum mass consumption of the task estimated by advance computation. Our optimisation objective encompasses two facets: minimissing mission completion time and reducing fuel consumption for servicing satellite manoeuvers. Unfortunately, these objectives are inherently in conflict; shorter mission completion times typically result in higher fuel consumption, necessitating a trade-off between the two. The weighting parameter κ captures this trade-off, where a higher value of κ indicates a greater emphasis on swift mission completion at the cost of increased fuel consumption. We will discuss the effect of the value of the weighting parameter κ on the experimental results in the subsequent section.

Remark 1. *This study introduces a pioneering approach that integrates the constraints of variances in target accessibility for optimising the scheduling of on-orbit services, conceptualising it as TRCTSP. To our knowledge, this original concept has not been previously proposed. TRCTSP distinguishes itself from CTSP* [22] *through the incorporation of a time-bounded and time-dependent cost function. Condition a defines the mathematical expression of the constraint on variances in target accessibility. Failure to satisfy this constraint incurs a significant total cost (19), due to the penalty term $\varepsilon n_{violate}$.*

Remark 2. *Indeed, the constraint of variances in target accessibility demands a higher level of optimisation expertise from the meta-heuristic algorithm compared to the time window restriction and fuel constraint* [9, 10, 15, 17]. *Moreover, the optimisation challenge is further intensified by the temporal optimisation sequence S^{time}_i. Traditional meta-heuristic algorithms such as GA, PSO, FA, etc., often fall short in producing optimal outcomes or even feasible solutions that comply with the constraints, especially as the number of serviced satellites increases. This shortcoming will be highlighted in subsequent experiments. Consequently, there is an urgent need to refine conventional meta-heuristic algorithms and enhance their optimisation capabilities to meet the model-solving requirements outlined in this study. In this context, FA will undergo refinement, as elaborated in the following section.*

4.0 Algorithm design

In this section, we will employ EFA to address the multi-objective on-orbit servicing problem delineated in the preceding section. The traditional firefly algorithm draws inspiration from the collective behaviour of fireflies during their aggregation and can be formulated as follows [4]:

(1) The light intensity of the ith firefly, represented by the solution s_i, is inversely proportional to the value of the cost function **C**.

(2) The attractiveness β_{ij} between firefly i and firefly j is mathematically defined as follows:

$$\beta_{ij} = \beta_0 \mathrm{e}^{-\gamma r_{ij}^2} \tag{20}$$

where β_0 represents the attractiveness when $r_{ij} = 0$, e denotes Euler's number, and γ signifies the light absorption coefficient. The Euclidean distance between two fireflies s_i and s_j, denoted as r_{ij}, is expressed as:

$$r_{ij} = \left\| s_i - s_j \right\| = \sqrt{\sum_{d=1}^{N} \left(s_{i,d} - s_{j,d} \right)^2} \tag{21}$$

where N characterises the dimensionality of the optimisation problem.

(3) The ith firefly shifts towards another more attractive jth firefly, and s_i is updated as follows:

$$s_i^k = s_i^{k-1} + \beta_{ij} \cdot \left(s_j^{k-1} - s_i^{k-1} \right) + \alpha \varepsilon_i \tag{22}$$

where k represents the iteration number of the algorithm, α stands for the step factor, and ε_i is an N-dimensional random number following either a uniform or Gaussian distribution. Scholars have primarily adopted a dynamic step-size strategy to enhance the stochastic aspect of the algorithm. In mathematical terms, this is represented as:

$$\alpha^{k+1} = \bar{\alpha} \alpha^k \tag{23}$$

where $\bar{\alpha}$ represents the dynamic step coefficient. With this adjustment, (22] transforms as follows:

$$s_i^k = s_i^{k-1} + \beta_{ij} \cdot \left(s_j^{k-1} - s_i^{k-1} \right) + \alpha^k \varepsilon_i \tag{24}$$

The multi-objective on-orbit servicing problem poses a formidable optimisation challenge with a multitude of local optimal solutions. The inherent complexity can result in the firefly algorithm experiencing premature convergence, characterised by a swift decline in population diversity, thereby impeding the optimisation process. Moreover, the pronounced attraction of the population to local optima may induce continuous oscillations, hindering the algorithm from achieving convergence. In response to these challenges, this study introduces PSMS and DES meticulously designed to augment algorithmic performance.

4.1 Positional substitution mutation strategy

During the update process of firefly individuals, we introduce PSMS. Let the solution of firefly individual i after the kth iteration be denoted as s_i^k. We randomly select two elements within s_i^k and exchange their positions, resulting in $\bar{s}_i^k$. Subsequently, the values of the consumption functions, denoted as $\mathbf{C}\left(s_i^k\right)$ and $\mathbf{C}\left(\bar{s}_i^k\right)$, are calculated. If $\mathbf{C}\left(\bar{s}_i^k\right) < \mathbf{C}\left(s_i^k\right)$, we replace the solution s_i^k with $\bar{s}_i^k$. This positional substitution operation is repeated N_s times.

On one hand, given that this operation occurs subsequent to individual position updates and is nested within the population update process, the value of N_s must not be overly large. Excessive values would augment the computational intricacy of the algorithm, consequently diminishing its convergence speed. On the other hand, if the frequency of these operations is too scant, the alterations in individual positions will be negligible, rendering the escape from local optima unattainable and impairing the precision of the optimisation search. Therefore, it is recommended that N_s be confined within the range $2 \leq N_s \leq 5$.

4.2 Differential evolution strategy

To maximise the utilisation of population-wide optimal information and enhance the performance of the algorithm, DES is incorporated into the population updating process. After k iterations, the best solution s_{best}^k within the firefly population is recorded. Subsequently, two solutions of fireflies, s_i^k and s_j^k,

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Algorithm 1: The enhanced firefly algorithm

```
Initialize $M_P$,$M_I$, $\alpha$, $\beta_0$, and $\gamma$;
Initialize the firefly population and get $\left\{s_1^0, \cdots, s_{M_P}^0\right\}$;
Calculate $\left\{\mathbf{C}(s_1^0), \cdots, \mathbf{C}(s_{M_P}^0)\right\}$;
while $k \le M_I$ do
    $\alpha^k \leftarrow \overline{\alpha}\alpha^{k-1}$;
    for $i \leftarrow 1$ to $M_P$ do
        for $j \leftarrow 1$ to $M_P$ do
            if $\mathbf{C}(s_i^{k-1}) > \mathbf{C}(s_j^{k-1})$ then
                $\beta_{ij} \leftarrow \beta_0 e^{-\gamma\|s_i - s_j\|^2}$;
                $s_i^k \leftarrow s_i^{k-1} + \beta_{ij} \cdot \left(s_j^{k-1} - s_i^{k-1}\right) + \alpha^k \boldsymbol{\varepsilon}_i$;
            end
        end
        $s_i^k \leftarrow$ SubstitutionMutation$\left(s_i^k\right)$;
    end
    $s_{best}^k \leftarrow$ DifferentialEvolution$\left(\left\{s_1^k, \cdots, s_{M_P}^k\right\}\right)$ ;
end
return $s_{best}^{M_I}$
```

are randomly selected from the population to compute $\bar{s}_{best}^k$ using the following formula:

$$\bar{s}_{best}^k = s_{best}^k + \eta \left(s_i^k - s_j^k\right) \tag{25}$$

In this equation, η is a variable following a normal distribution. Subsequently, the values of the consumption functions, denoted as $\mathbf{C}\left(s_{best}^k\right)$ and $\mathbf{C}\left(\bar{s}_{best}^k\right)$, are calculated. If $\mathbf{C}\left(\bar{s}_{best}^k\right) < \mathbf{C}\left(s_{best}^k\right)$, we replace the solution s_{best}^k with $\bar{s}_{best}^k$. The differential evolution operation is repeated N_d times.

As there is no circular nesting in this operation, N_d can be chosen to be a relatively larger value, enabling the utilisation of historical optimal information to guide the search process and enhance the algorithm's convergence accuracy. Taking cues from the differential evolution algorithm, the value of N_d can be set equal to the population size, denoted as M_P.

The pseudo-code of EFA is depicted in Algorithm 1. DES is outlined in Algorithm 2, and PSMS is detailed in Algorithm 3. Here, the function **RandomNaturalNumber** (M_P) denotes the generation of random natural numbers that are less than or equal to M_P.

Remark 3. *Inspired by the methodology introduced in LNS* [17]*, we introduce PSMS as an enhancement to FA. However, PSMS, as proposed in this study, offers two distinct advantages over LNS: (a) While LNS executes at the conclusion of each iteration of the algorithm, PSMS operates during each update of the population individuals within the algorithm. Specifically, for each optimisation instance, LNS executes M_I times, whereas PSMS executes M_IM_P times. This discrepancy implies that the extensive exploration capacity of PSMS far exceeds that of LNS. (b) In contrast to the intricate destruction and repair operations featured in LNS, PSMS employs a simpler and more efficacious positional substitution operation. This operation is equally adept at enhancing local search capabilities. Additionally, PSMS incorporates an acceptance criterion akin to a hill-climber for new solutions, wherein solutions are updated only when superior results are achieved, thereby averting the generation of detrimental new solutions. This enhancement effectively amplifies the algorithm's capacity for extensive exploration and facilitates the generation of solutions that adhere to imposed constraints.*

Remark 4. *DES draws inspiration from the principles of the differential evolution algorithm (DEA) outlined in* [23]*, operating at the conclusion of each iteration of EFA. Given that DES aims to augment*

Algorithm 2: Function of **DifferentialEvolution** $\left(\left\{s_1^k, \cdots, s_N^k\right\}\right)$

```
Input: $\left\{\mathbf{s}_1^k, \cdots, \mathbf{s}_N^k\right\}$
Initialize $\eta$;
Initialize $\mathbf{s}_{best}^k = \mathbf{s}_1^k$;
for $i \leftarrow 1$ to $M_P$ do
    if C($\mathbf{s}_{best}^k$) > C($\mathbf{s}_i^k$) then
        $\mathbf{s}_{best}^k \leftarrow \mathbf{s}_i^k$;
    end
end
$i$ = RadomNaturalNumber($M_P$);
$j$ = RadomNaturalNumber($M_P$);
for $q \leftarrow 1$ to $N_d$ do
    $\overline{\mathbf{s}}_{best}^k \leftarrow \mathbf{s}_{best}^k + \eta(\mathbf{s}_i^k - \mathbf{s}_j^k)$;
    if C($\mathbf{s}_{best}^k$) > C($\overline{\mathbf{s}}_{best}^k$) then
        $\mathbf{s}_{best}^k \leftarrow \overline{\mathbf{s}}_{best}^k$;
    end
end
return $\mathbf{s}_{best}^k$
```

Algorithm 3: Function of **SubstitutionMutation** $\left(s_i^k\right)$

```
Input: $s_i^k$
for $p \leftarrow 1$ to $N_s$ do
    $q$ = RadomNaturalNumber($N$);
    $w$ = RadomNaturalNumber($N$);
    $\overline{\mathbf{s}}_i^k \leftarrow \mathbf{s}_i^k$;
    $\overline{s}_{i,q}^k \leftarrow s_{i,w}^k$;
    $\overline{s}_{i,w}^k \leftarrow s_{i,q}^k$;
    if C($\mathbf{s}_i^k$) > C($\overline{\mathbf{s}}_i^k$) then
        $\mathbf{s}_i^k \leftarrow \overline{\mathbf{s}}_i^k$;
    end
end
return $s_i^k$
```

FA's search prowess and escape local optima, it incorporates an acceptance criterion akin to a hill-climber for new solutions, a feature not present in DEA. Consequently, the algorithm's local search capability is effectively bolstered.

5.0 Experiments and results

In this section, we will conduct three scenarios with a total of nine cases of experiments.

5.1 Experimental settings

Each servicing satellite has a total mass of 1,000 kg and carries 500 kg of chemical fuel, with the thruster parameter $g_0 I_{sp}$ set at 3,000 m/s [9]. The initial solutions are generated randomly, and the parameter

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 1. *Parameter settings*

Parameter	Value	Parameter	Value
Weight parameter κ	0.05	population size M_P	100
Maximum iteration M_I	200	initial attractiveness β_0	2
Light absorption coefficient γ	1	initial step factor α	0.5
Dynamic step coefficient $\bar{\alpha}$	0.98	maximum fuel consumption $\delta m_{i,max}$	500 kg
Minimal transfer time $t^{lambert}_{i,min}$	6h	maximum transfer time $t^{lambert}_{i,max}$	40 h
Minimal parking time $t^{parking}_{i,min}$	6h	maximum parking time $t^{parking}_{i,max}$	40 h
Minimal service time $t^{service}_{i,p,k,min}$	6h	maximum service time $t^{service}_{i,p,k,max}$	40 h
Minimal total timet^{final}_{min}	48h	maximum total timet^{final}_{max}	480 h
Penalty coefficient ε	2		

Table 2. *Servicing capabilities of servicing satellites*

ID	Case 1	Case 2	Case 3
1	[c]refueling/inspection	[c]maintenance/inspection	[c]debris removal/inspection
2	[c]refueling/inspection	[c]maintenance/inspection	[c]debris removal/inspection
3	[c]refueling/inspection	[c]maintenance/inspection	[c]debris removal/inspection
4	–	inspection	inspection
5	–	–	inspection
6	–	–	inspection

settings used during the simulation are detailed in Table 1. The experimental setup includes an Intel(R) Core(TM) i7-13700 CPU.

In scenario 1, the space station, all servicing satellites, and all targets are operating in geostationary Earth orbit (GEO). Drawing inspiration from the GEO orbital element settings in the literature [17, 24], the corresponding orbital elements are randomly generated within the following ranges: $a \in [42,164,\ 42,166]$ km, $e \in [0, 0.0004]$, $\Omega \in [0, 360]^\circ$, $i \in [0, 0.05]^\circ$, $\omega \in [0, 360]^\circ$, $\theta \in [0, 360]^\circ$. Experiments will be conducted under scenario 1 with three cases: 3 satellites serving 8 targets, 4 satellites serving 15 targets, and 6 satellites serving 20 targets. Table 2 outlines the service capabilities of the servicing satellites in these three cases, where ID represents the number of the servicing satellite. The initial orbital parameters for the three cases in scenario 1, along with the specific on-orbit service requirements, are presented in Tables 7, 10 and 11 in the Appendix, where ID denotes the number of each target, and ID 0 represents the space station.

In scenario 2, the space station, all servicing satellites, all targets are operating on randomly generated orbits. The corresponding orbital elements are randomly generated in the following range: $a \in [6,800,\ 42,000]$ km, $e \in [0, 0.2]$, $\Omega \in [0, 360]^\circ$, $i \in [0, 180]^\circ$, $\omega \in [0, 360]^\circ$, $\theta \in [0, 360]^\circ$. We will conduct experiments under scenario 2 with 3 satellites serving 8 targets, 4 satellites serving 15 targets, and 6 satellites serving 20 targets, for a total of three cases. Table 2 outlines the service capabilities of the servicing satellites in three cases under scenario 2, where ID is the number of the serving satellite. The initial orbital parameters for the three cases under scenario 2 and the specific on-orbit service requirements are presented in Tables 8, 12 and 13 in the Appendix, where ID is the number of each target and ID 0 represents the space station.

In scenario 3, the space station, all servicing satellites, all targets are operating on low Earth orbits. The corresponding orbital elements are randomly generated in the following range: $a \in [7,380,\ 7,400]$ km, $e \in [0, 0.0004]$, $\Omega \in [99, 101]^\circ$, $i \in [81, 82]^\circ$, $\omega \in [0, 0]^\circ$, $\theta \in [0, 360]^\circ$ [14]. We will conduct experiments under scenario 3 with 3 satellites serving 8 targets, 4 satellites serving 15 targets, and 6 satellites serving 20 targets, for a total of three cases. Table 2 outlines the service capabilities of the servicing satellites in three cases under scenario 3, where ID is the number of the serving satellite. The initial orbital parameters for the three cases under scenario 3 and the specific on-orbit service requirements are presented in Tables 9, 14 and 15 in the Appendix, where ID is the number of each target and ID 0 represents the space station.

5.2 Optimisation results

Tables 16, 17 and 18 give the optimisation results for the three cases of scenario 1, including transfer sequence, time of arrival, time of departure, velocity change and total fuel mass consumption.

Here, we will use the information of servicing satellite 1 from the optimisation results under case 1, scenario 1, as an example to elucidate the significance of the optimisation outcome.

Transfer sequence 0-7-4-1-0 indicates that servicing satellite 1 departs from the space station, sequentially provides on-orbit services to targets 7, 4 and 1, and ultimately returns to the space station.

Time of arrival refers to the moment when servicing satellite 1 reaches the target corresponding to the transfer sequence mentioned above. Specifically, servicing satellite 1 is stationed at the initial moment of 0. The subsequent arrival times at targets 7, 4 and 1 are 47.97, 86.37 and 127.96 h, respectively. Finally, the satellite returns to the station at 166.16 h.

Time of departure denotes the moment when servicing satellite 1 departs from the target corresponding to the transfer sequence mentioned above. Specifically, satellite 1 left the space station at 22.47 h and departed from targets 7, 4 and 1 at 63.97, 106.66 and 143.96 h, respectively.

It is noteworthy that the time of arrival and time of departure can be used to calculate both the duration required for each orbital transfer and the time spent on each on-orbit service. For instance, the time spent servicing target 7 by satellite 1 is determined by subtracting the time of arrival at target 7 from the time of departure from target 7, yielding 16 h. The time required for the orbital transfer from the station to target 7 is calculated by subtracting the time of departure from the station from the time of arrival at target 7, resulting in 25 h.

Additionally, we provide the velocity change required for each orbital transfer based on Lambert's theory. For example, the velocity change required to maneuver from the station to target 7 is 178.04 m/s. Finally, the table details the fuel consumption of each servicing satellite upon completing its mission.

Tables 19, 20 and 21 present the optimisation results for the three cases in scenario 2. Tables 22, 23 and 24 present the optimisation results for the three cases in scenario 3. These results are displayed in the same manner as those for scenario 1.

It should be noted that scenario 2 is purely a mathematical simulation and does not account for practical engineering applications. When parameterising the target initial orbits for scenario 2, we did not reference values from commonly used orbits. The purpose of this setup is twofold: to verify the validity and superiority of the proposed algorithm from another perspective, and to lay the theoretical groundwork for potential spatial applications. The results indicate that the velocity change required for orbital transfer is substantial under the scenario 2 settings. Conventional chemical fuels would be insufficient to meet such demands. Therefore, in scenario 2, we assume that the servicing satellite is equipped with high-power ion thrusters, offering a specific impulse of up to 70,000 m/s [25].

5.3 Superiority experimental results

In this subsection, we conduct a comparative analysis between GA [3], LNS-GA [17], PSO [5], T-PSO [21] and FA [4] as benchmark algorithms against EFA. Each algorithm undergoes execution on 10 randomly generated instances to comprehensively assess their optimisation capabilities. The rules for generating orbital elements for each instance are the same as for scenario 1 and scenario 2. Detailed parameters for the compared GA, and PSO algorithms are provided in Table 3, while parameters for the compared LNS-GA and T-PSO algorithms are elucidated in Table 4.

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

***Table 3.** The parameters of the compared GA, PSO*

GA		PSO	
Population size	200	Population size	300
Maximum iteration	1,000	Maximum iteration	500
Crossover probability	0.6	Inertia weight	0.8
Mutation probability	0.01	Maximum velocity	1
		Individual learning factor	1.5
		Group learning factor	1.5

***Table 4.** The parameters of the compared LNG-GA, T-PSO*

LNS-GA		T-PSO	
Population size	200	Population size	300
Maximum iteration	1,000	Maximum iteration	200
Crossover probability	0.6	Inertia weight	0.8
Mutation probability	0.01	Maximum velocity	1
Proportion of elite chromosomes in LNS	5%	Individual learning factor	1.5
Maximum iteration of LNS	10	Group learning factor	1.5
Percentage of LNS individuals removed	30%	Number of repository	100
		Number of grid size	20

***Table 5.** Average cost for different algorithms*

	EFA	FA	PSO	T-PSO	GA	LNS-GA
Case 1 scenario 1	0.5245	0.5326	0.5473	0.5308	0.6293	0.5259
Case 2 scenario 1	0.5625	0.6029	0.5946	0.5861	0.7822	0.5827
Case 3 scenario 1	0.5852	0.7392	–	–	0.7459	0.6895
Case 1 scenario 2	0.6953	0.7189	0.7294	0.7031	0.7246	0.7003
Case 2 scenario 2	0.7170	0.7725	0.7630	0.7593	0.8039	0.7493
Case 3 scenario 2	0.7189	0.8493	–	–	0.8991	0.7982
Case 1 scenario 3	0.6383	0.7138	0.7091	0.6937	0.7497	0.6628
Case 2 scenario 3	0.6412	0.7331	0.7207	0.7238	0.7737	0.6901
Case 3 scenario 3	0.6609	0.7422	–	–	0.7966	0.7168

Table 5 presents a comparison of the average cost achieved by various algorithms. In case 1, marked by lower problem complexity, the differences in solutions generated by the three algorithms are subtle. However, in case 3, characterised by higher problem complexity, the EFA proposed herein demonstrates a significantly superior solution quality compared to the other algorithms. Additionally, the average number of iterations failing to meet the constraints provides further insight into the algorithms' ability to explore a broad solution space, as shown in Table 6. Notably, both PSO and TPSO fail to produce solutions that adhere to the constraints in case 3. The reasons for this outcome will be elaborated upon in the following subsection.

5.4 Hyperparametric sensitivity experiments

In this subsection, we will analyse the effects of the weight parameter κ and the penalty coefficient ε on the optimisation results, using case 1 of scenario 1 as an example.

We conducted experiments by varying the weight parameter κ from 0 to 0.5 in intervals of 0.05. To mitigate the impact of random factors, each experiment was repeated 10 times. The experimental results

Table 6. *Average number of the iterations unsatisfying the constraints for different algorithms*

	EFA	FA	PSO	T-PSO	GA	LNS-GA
Case 1 scenario 1	1.1	1.3	4.0	3.6	1.2	1.1
Case 2 scenario 1	1.2	1.4	19.6	17.3	13.2	11.6
Case 3 scenario 1	3.3	6.1	496.7	495.2	31.6	29.8
Case 1 scenario 2	0.9	1.2	4.1	3.3	1.1	0.9
Case 2 scenario 2	1.0	1.4	19.5	16.8	13.4	11.5
Case 3 scenario 2	3.6	5.8	497.2	494.1	33.9	31.5
Case 1 scenario 3	0.9	1.2	4.2	3.5	1.1	0.9
Case 2 scenario 3	0.9	1.4	18.7	17.1	13.5	12.4
Case 3 scenario 3	3.3	5.6	493.4	496.1	32.7	30.4

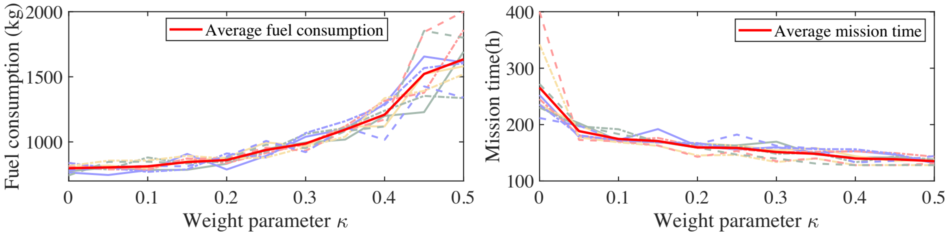

Figure 2. *Optimisation results under different weight parameters κ.*

are depicted in Fig. 2, where the dashed curves represent the results of the 10 individual experiments, and the red curve represents the average value across these experiments. As observed, an increase in the weight parameter κ indicates that the optimisation algorithm is more inclined to select routes that minimise time at the expense of higher fuel consumption.

For a single orbital transfer, it is possible to achieve a more time-efficient and fuel-efficient re-orbiting based on Lambert's theory. However, as Fig. 2 illustrates, increasing κ does not necessarily reduce mission time while increasing fuel consumption. Nonetheless, from a broader perspective, by analysing the average data from multiple experiments, we can draw some conclusions. If the priority is to conserve fuel while accepting longer mission durations, κ should be decreased. Conversely, if a shorter mission time is preferred, even at the cost of higher fuel consumption, κ should be increased.

We conducted experiments by varying the penalty coefficient ε from 0.2 to 2 in intervals of 0.2. To mitigate the effects of random factors, each experiment was repeated 10 times. The experimental results are displayed in Fig. 3, where the dashed curves represent the results of the 10 individual experiments, and the red curve represents the average value across these experiments. When ε is set to 0.2 and 0.4, the algorithm fails to produce optimisation results that meet the constraints. As ε gradually increases, the optimization algorithm becomes more inclined to generate results that satisfy the constraints. Once ε reaches a certain threshold, the algorithm consistently produces results that almost always satisfy the constraints. Therefore, we recommend choosing a larger value for ε. Given that the objective function is normalised, we have set $\varepsilon = 2$ in this study.

5.5 Discussion of results

Based on the experimental outcomes presented in the preceding subsection, it is clear that in cases with fewer optimisation variables — indicating a reduced number of service satellites — and lower complexity, the differences among the algorithms are minimal. Conversely, in cases of increased complexity, the

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

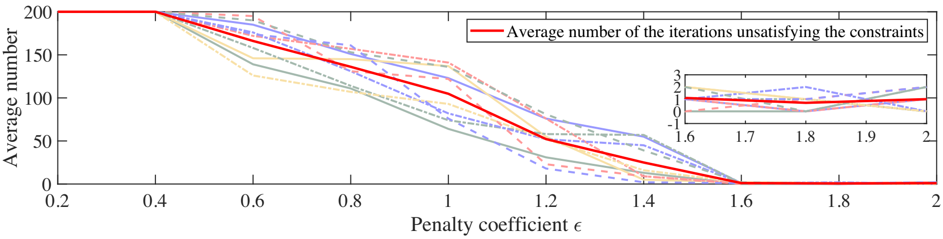

Figure 3. *Optimisation results under different penalty coefficients* ε*.*

proposed EFA demonstrates notably superior performance. Below, we provide a detailed analysis of the results for each algorithm.

The particles in PSO possess both velocity and position attributes, granting PSO robust local search capabilities. Consequently, PSO and T-PSO perform well in scenario 1, which is characterised by low model complexity. However, PSO lacks the stochastic elements found in GA, such as mutation and crossover operations, as well as the stochastic term $\alpha\varepsilon_i$ in FA's update mechanism. This absence limits PSO's ability to explore large solution spaces thoroughly. As a result, PSO and T-PSO struggle to generate solutions that adhere to constraints in scenario 3, which is of high complexity. Notably, T-PSO, optimised solely with hyper-parameters, does not significantly enhance PSO's exploration or local search capabilities, leading to only a modest improvement over PSO.

GA exhibits a strong capacity for exploring expansive solution spaces due to its use of crossover and mutation operations. However, GA is known for its limited local search capabilities and tendency towards premature convergence [26]. Although LNS-GA was introduced by [17] to improve GA's local search performance, this enhancement does not fully address GA's limitations. Consequently, both GA and LNS-GA underperform relative to FA. Nevertheless, GA's ability to explore vast solution spaces allows it to find solutions that conform to constraints even in scenario 3, characterised by the highest complexity — unlike PSO, where exploration efforts are insufficient.

In contrast to PSO and GA, FA achieves a better balance between extensive exploration and local search capabilities. The incorporation of PSMS enhances FA's ability to explore large solution spaces, while DES improves its local search performance, effectively addressing the challenges posed by TRCTSP. Notably, in scenario 3, the EFA demonstrates markedly superior performance compared to other meta-heuristics. We believe that EFA will continue to outperform other algorithms in cases with elevated complexity.

6.0 Conclusion

This article explores the complexities of optimal scheduling for many-to-many on-orbit services, incorporating detailed considerations of variations in target accessibility. In addressing the inherently challenging multi-objective optimal scheduling problem, a paradigm within the NP-hard realm of combinatorial optimisation, we introduce the EFA. The simulation results across various scenarios decisively demonstrate the efficacy and superiority of the proposed algorithm. Future research will focus on refining and advancing intelligent algorithms to tackle the real-time optimal scheduling challenges inherent to on-orbit services.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (grant number: 61304108).

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

Competing interests. 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 article.

References

1. Dorigo, M. and Gambardella, L.M. Ant colonies for the travelling salesman problem, *Biosystems*, 1997, **43**, (2), pp 73–81.
2. Baker, B.M. and Ayechew, M.A. A genetic algorithm for the vehicle routing problem, *Comput. Oper. Res.*, 2003, **30**, (5), pp 787–800.
3. Mirjalili, S. *Genetic Algorithm*, pp 43–55. Springer International Publishing, 2019.
4. Fister, I., Fister, I., Yang, X.-S. and Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.*, 2013, **13**, pp 34–46.
5. Kennedy, J. and Eberhart, R. Particle swarm optimization, In *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp 1942–1948, 1995.
6. Dorigo, M., Birattari, M. and Stutzle, T. Ant colony optimization, *IEEE Comput. Intell. Mag.*, 2006, **1**, (4), pp 28–39.
7. Yang, Z., Ye, Y., Xu, H., Yueneng, Y. and Hua, Z. Mission planning optimization for the visual inspection of multiple geosynchronous satellites, *Eng. Optim.*, 2015, **47**, (11), pp 1543–1563.
8. Sorenson, S.E. and Nurre Pinkley, S.G. Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots, *Comput. Ind. Eng.*, 2023, **176**, p 108852.
9. Zhang, J., Parks, G.T., Luo, Y. and Tang, G. Multispacecraft refueling optimization considering the j2 perturbation and window constraints, *J. Guid. Control Dyn.*, **37**(1):111–122, 2014.
10. Li, C. and Xu, B. Optimal scheduling of multiple sun-synchronous orbit satellites refueling, *Adv. Space Res.*, 2020, **66**, (2), pp 345–358.
11. Han, P., Guo, Y., Wang, P., Li, C. and Pedrycz, W. Optimal orbit design and mission scheduling for sun-synchronous orbit on-orbit refueling system, *IEEE Trans. Aerosp. Electron. Syst.*, 2023, **59**, (5), pp 4968–4983.
12. Rodrigues Neto, J.B. and Ramos, G.D.O. An interpolated approach for active debris removal, In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pp 1–6, 2022.
13. Zona, D., Zavoli, A., Federici, L. and Avanzini, G. Evolutionary optimization for active debris removal mission planning, *IEEE Access*, 2023, **11**, pp 41019–41033.
14. Jing, Y., Xiaoqian, C. and Lihu, C. Optimal planning of Leo active debris removal based on hybrid optimal control theory, *Adv. Space Res.*, 2015, **55**, (11), pp 2628–2640.
15. Shen, H.X., Zhang, T.J., Casalino, L. and Pastrone, D. Optimization of active debris removal missions with multiple targets, *J. Spacecr. Rockets*, 2018, **55**, (1), pp 181–189.
16. Bang, J. and Ahn, J.. Multitarget rendezvous for active debris removal using multiple spacecraft, *J. Spacecr. Rockets*, **56**(4):1237–1247, 2019.
17. Han, P., Guo, Y., Li, C., Zhi, H. and Lv, Y. Multiple geo satellites on-orbit repairing mission planning using large neighborhood search-adaptive genetic algorithm, *Adv. Space Res.*, 2022, **70**, (2), pp 286–302.
18. Chen, H., Gardner, B.M., Grogan, P.T. and Ho, K. Flexibility management for space logistics via decision rules, *J. Spacecr. Rockets*, 2021, **58**, (5) pp 1314–1324.
19. Sarton, T., Chen, H., Gunasekara, O. and Ho, K. Framework for modeling and optimization of on-orbit servicing operations under demand uncertainties, *J. Spacecr. Rockets*, 2021, **58**, (4), pp. 1157–1173.
20. Bakhtiari, M., Daneshjou, K. and Mohammadi-Dehabadi, A.A. The effects of parking orbit elements on designing of on-orbit servicing missions. *Proc. Inst. Mech. Eng. G: J. Aerosp. Eng.*, 2019, **233**, (3), pp 793–810.
21. Daneshjou, K., Mohammadi-Dehabadi, A.A. and Bakhtiari, M. Mission planning for on-orbit servicing through multiple servicing satellites: A new approach, *Adv. Space Res.*, 2017, **60**, (6), pp 1148–1162.
22. Li, J., Zhou, M.C., Sun, Q., Dai, X. and Yu, X. Colored traveling salesman problem, *IEEE Trans. Cybern.*, 2015, **45**, (11), pp 2390–2401.
23. Das, S. and Suganthan, P.N. Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.*, 2011, **15**, (1), pp 4–31.
24. Yan, H., Guo, Y., Li, C. and Song, B. Neural-network-assisted optimization of a close-range multi-spacecraft rendezvous mission based on a multi-impulse maneuvering strategy, *Adv. Space Res.*, 2023, **72**, (5), pp 1829–1843.
25. Koroteev, A.S., Lovtsov, A.S., Vyacheslav, A., Muravlev, M.Y., Andrey, S. and Shagayda, A. Development of ion thruster it-500, *Eur. Phys. J. D*, 2017, **71**, (120), pp 1–10.
26. Liu, Y., Guo, C., and Weng, Y. Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance, *IEEE Access*, 2019, **7**, pp 146301–146308.
27. Curtis, H.D. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2013.

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Appendix

A.1 Initial states of targets in the experiments

Table 7. *Initial states of targets for case 1 under scenario 1*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	42,164.0	0.0000	1.7058	0.0005	3.7037	2.3942	–
1	42,164.0	0.0003	3.2898	0.0003	3.5455	0.9424	refueling
2	42,164.8	0.0002	0.7524	0.0003	0.5940	4.7981	maintenance
3	42,164.9	0.0004	0.6567	0.0008	5.2180	3.9838	debris removal
4	42,165.0	0.0002	0.1495	0.0007	2.7429	4.3405	inspection
5	42,165.6	0.0000	0.2207	0.0002	5.8959	3.7696	inspection
6	42,165.4	0.0000	3.5175	0.0008	4.3624	0.2257	inspection
7	42,165.7	0.0001	0.4006	0.0006	3.1557	3.6618	inspection
8	42,164.2	0.0001	0.6512	0.0007	1.8542	3.9623	inspection

Table 8. *Initial states of targets for case 1 under scenario 2*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	16,135.4	0.0894	0.5257	2.0007	4.6034	1.1702	–
1	25,076.6	0.1566	4.9237	0.3947	3.0758	5.7967	refueling
2	39,312.1	0.0637	1.0540	2.9154	4.5576	6.1763	maintenance
3	23,741.7	0.1056	1.9343	1.7716	1.6739	4.1415	debris removal
4	24,887.3	0.0808	5.3589	0.0084	2.8596	3.6299	inspection
5	17,896.7	0.0106	1.4860	3.1178	0.1810	2.4507	inspection
6	18,001.7	0.1264	2.1591	2.3055	2.0869	3.7917	inspection
7	23,624.2	0.1200	0.5668	2.9625	1.5340	2.5593	inspection
8	11,738.0	0.0427	2.8913	0.6473	3.2807	0.3367	inspection

Table 9. *Initial states of targets for case 1 under scenario 3*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	7,384.7	0.0000	1.7464	1.4181	0.0000	2.5536	–
1	7,391.1	0.0003	1.7338	1.4195	0.0000	2.8414	refueling
2	7,392.6	0.0003	1.7339	1.4299	0.0000	0.4483	maintenance
3	7,391.9	0.0003	1.7538	1.4219	0.0000	2.5916	debris removal
4	7,392.4	0.0004	1.7410	1.4218	0.0000	5.7121	inspection
5	7,399.0	0.0002	1.7540	1.4296	0.0000	1.9326	inspection
6	7,393.9	0.0000	1.7404	1.4144	0.0000	1.3444	inspection
7	7,396.5	0.0003	1.7489	1.4167	0.0000	2.5081	inspection
8	7,388.6	0.0003	1.7316	1.4194	0.0000	3.3447	inspection

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 10. *Initial states of targets for case 2 under scenario 1*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	42,164.0	0.0000	2.0787	0.0006	2.8590	5.6994	–
1	42,164.3	0.0001	1.5570	0.0002	3.0678	3.7583	refueling
2	42,165.4	0.0000	0.9676	0.0007	3.9923	2.3108	maintenance
3	42,165.8	0.0003	1.3160	0.0001	0.6807	3.6536	debris removal
4	42,164.5	0.0004	1.7927	0.0004	4.1930	4.4842	inspection
5	42,164.4	0.0001	3.1010	0.0007	2.4538	5.1072	inspection
6	42,164.0	0.0000	2.5755	0.0008	3.7668	2.9707	inspection
7	42,164.2	0.0002	4.8201	0.0000	4.9949	5.4380	inspection
8	42,164.4	0.0001	3.7595	0.0005	5.6134	4.8034	inspection
9	42,164.9	0.0003	2.2350	0.0004	5.0645	3.5919	inspection
10	42,165.5	0.0001	0.0242	0.0006	1.5905	6.0879	inspection
11	42,165.6	0.0002	5.1638	0.0008	1.3862	0.2744	refueling
12	42,165.0	0.0000	4.8333	0.0007	5.3075	5.3121	maintenance
13	42,165.0	0.0001	2.8778	0.0004	0.6897	5.8202	debris removal
14	42,164.5	0.0001	2.8737	0.0005	3.5417	1.4580	inspection
16	42,164.0	0.0002	4.5925	0.0008	5.8385	1.2789	inspection

Table 11. *Initial states of targets for case 3 under scenario 1*

ID	a [km]	e	Ω [rad]	[rad]	ω [rad]	θ [rad]	required service
0	42,164.0	0.0000	5.2907	0.0000	0.6038	6.0505	–
1	42,165.2	0.0001	5.6384	0.0005	0.2003	5.1996	refueling
2	42,164.6	0.0001	2.8147	0.0006	0.1068	3.8796	maintenance
3	42,164.1	0.0002	1.3625	0.0008	0.5010	2.0426	debris removal
4	42,165.2	0.0000	1.9892	0.0005	3.6017	6.0685	inspection
5	42,165.0	0.0001	5.6697	0.0004	1.0207	0.0258	inspection
6	42,164.7	0.0003	1.7578	0.0004	6.1647	5.8373	inspection
7	42,165.2	0.0004	5.0752	0.0008	4.6879	1.3635	inspection
8	42,164.6	0.0003	6.1512	0.0001	0.0663	5.6538	inspection
9	42,165.5	0.0004	1.8632	0.0002	2.4215	3.9798	inspection
10	42,165.0	0.0002	4.9116	0.0008	2.0708	3.9930	inspection
11	42,164.9	0.0001	3.0714	0.0001	1.1874	2.2029	refueling
12	42,164.6	0.0004	4.7874	0.0006	3.0441	0.8490	maintenance
13	42,164.4	0.0003	3.4757	0.0002	4.6555	1.5517	debris removal
14	42,164.3	0.0000	0.3318	0.0003	3.9019	4.3703	inspection
15	42,164.3	0.0003	5.9307	0.0005	0.6161	5.3227	inspection
16	42,164.7	0.0002	3.4963	0.0001	5.5216	4.6049	inspection
17	42,165.7	0.0001	0.0978	0.0004	0.0654	4.8759	inspection
18	42,164.4	0.0001	0.8480	0.0006	6.0143	1.7793	inspection
19	42,164.4	0.0002	5.8500	0.0005	2.5926	5.7597	inspection
20	42,165.3	0.0004	4.3205	0.0008	1.4077	3.1285	inspection

Table 12. *Initial states of targets for case 2 under scenario 2*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	32,349.0	0.0800	4.4186	1.6426	2.1298	2.5422	–
1	16,528.0	0.1272	3.3809	3.1268	1.4633	0.9744	refueling
2	21,743.4	0.1831	2.0564	0.0945	4.8581	1.1613	maintenance
3	24,962.2	0.1162	2.9842	1.9939	0.3311	4.4228	debris removal
4	37,326.0	0.0781	0.7776	1.4795	2.5047	0.9258	inspection
5	19,597.9	0.0666	5.1364	2.5469	0.6775	0.5338	inspection
6	40,226.6	0.1096	4.9026	0.6090	1.5037	2.1889	inspection
7	29,690.0	0.1113	6.2369	3.0312	3.8077	2.6829	inspection
8	17,323.6	0.0343	5.6955	1.3287	5.8520	4.9637	inspection
9	39,820.5	0.1246	0.1465	3.1308	1.6712	0.353	inspection
10	41,232.6	0.0665	4.3681	0.4918	4.3699	4.6217	inspection
11	28,305.3	0.1327	5.6321	2.9972	4.3882	4.1173	refueling
12	26,297.1	0.1273	3.9983	1.4328	4.8198	1.7731	maintenance
13	30,824.1	0.1271	6.0958	2.5749	4.4030	5.4935	debris removal
14	15,621.8	0.1613	1.7257	1.1296	6.1768	5.4371	inspection
15	25,603.0	0.0163	3.6410	1.8565	3.9882	1.0988	inspection

Table 13. *Initial states of targets for case 3 under scenario 2*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
1	35,575.5	0.0792	0.3542	0.0254	2.6057	2.2282	–
2	41,509.5	0.1914	4.4379	1.9455	5.1448	1.6144	refueling
3	29,541.2	0.1293	3.5779	1.3323	6.1324	4.9055	maintenance
4	26,203.0	0.0604	5.1818	1.1609	0.7539	5.6288	debris removal
5	17,788.1	0.0414	4.3486	0.3152	1.0793	5.4373	inspection
6	13,507.7	0.0769	0.7773	0.7281	1.7491	3.8874	inspection
7	16,472.1	0.1795	4.9559	0.3466	0.7818	0.1792	inspection
8	17,726.8	0.0796	5.4837	2.5575	0.5911	1.6282	inspection
9	16,607.2	0.0173	5.3244	0.1680	4.6359	4.5053	inspection
10	12,706.3	0.1650	2.7365	1.6766	1.7691	1.6036	inspection
11	12,639.8	0.1431	5.0481	2.7222	3.2868	3.0098	inspection
12	13,474.9	0.1274	1.5462	0.5920	0.8853	5.7875	refueling
13	26,188.6	0.1590	1.9355	1.9821	3.4268	4.7601	maintenance
14	39,302.4	0.0993	3.8420	1.2998	5.7027	2.1747	debris removal
15	36,351.1	0.1648	1.3511	2.4524	4.6223	3.5101	inspection
16	36,974.1	0.0188	5.7283	1.9600	4.4744	2.366	inspection
17	20,866.0	0.0986	0.3563	2.9771	4.7818	1.2463	inspection
18	19,979.1	0.0032	5.9669	2.7641	2.4774	5.6024	inspection
19	22,573.8	0.0150	5.2209	2.7656	1.4891	4.0333	inspection
20	33,393.0	0.0939	5.7515	0.0863	0.9627	2.6612	inspection
21	9,197.20	0.0833	2.6816	2.1972	4.4693	2.3044	inspection

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 14. *Initial states of targets for case 2 under scenario 3*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	7,388.5	0.0000	1.7469	1.4241	0.0000	4.2119	–
1	7,391.2	0.0001	1.7502	1.4248	0.0000	2.7974	refueling
2	7,389.1	0.0003	1.7404	1.4288	0.0000	2.7254	maintenance
3	7,388.4	0.0003	1.7542	1.4166	0.0000	4.4086	debris removal
4	7,387.5	0.0003	1.7582	1.4269	0.0000	4.2240	inspection
5	7,392.6	0.0002	1.7586	1.4200	0.0000	2.9425	inspection
6	7,385.0	0.0001	1.7548	1.4256	0.0000	5.8146	inspection
7	7,391.5	0.0003	1.7397	1.4171	0.0000	4.0104	inspection
8	7,399.2	0.0001	1.7598	1.4156	0.0000	0.2401	inspection
9	7,394.4	0.0001	1.7311	1.4297	0.0000	0.1304	inspection
10	7,387.0	0.0000	1.7620	1.4163	0.0000	4.6993	inspection
11	7,396.7	0.0004	1.7346	1.4142	0.0000	3.8831	refueling
12	7,380.8	0.0002	1.7417	1.4191	0.0000	5.1202	maintenance
13	7,392.5	0.0001	1.7449	1.4287	0.0000	1.8991	debris removal
14	7,398.4	0.0000	1.7595	1.4204	0.0000	2.5560	inspection
15	7,388.7	0.0000	1.7453	1.4280	0.0000	4.5640	inspection

Table 15. *Initial states of targets for case 3 under scenario 3*

ID	a [km]	e	Ω [rad]	i [rad]	ω [rad]	θ [rad]	required service
0	7,389.4	0.0000	1.7378	1.4170	0.0000	0.2817	–
1	7,394.3	0.0002	1.758	1.4183	0.0000	5.3328	refueling
2	7,389.7	0.0001	1.7474	1.4252	0.0000	2.1603	maintenance
3	7,393.2	0.0002	1.7409	1.4264	0.0000	5.9693	debris removal
4	7,380.4	0.0001	1.7342	1.4287	0.0000	5.1246	inspection
5	7,399.3	0.0001	1.7326	1.4191	0.0000	0.5256	inspection
6	7,383.4	0.0003	1.7381	1.4247	0.0000	4.9442	inspection
7	7,382.5	0.0001	1.7568	1.4248	0.0000	2.2955	inspection
8	7,394.1	0.0002	1.7432	1.4250	0.0000	2.2540	inspection
9	7,385.4	0.0004	1.7354	1.4184	0.0000	5.2520	inspection
10	7,397.8	0.0004	1.7475	1.4236	0.0000	4.6847	inspection
11	7,388.9	0.0002	1.7453	1.4218	0.0000	0.7531	refueling
12	7,395.1	0.0001	1.7490	1.4183	0.0000	5.2362	maintenance
13	7,394.6	0.0002	1.7365	1.4187	0.0000	6.0598	debris removal
14	7,398.4	0.0002	1.7356	1.4176	0.0000	2.0813	inspection
15	7,395.6	0.0001	1.7413	1.4167	0.0000	3.3283	inspection
16	7,385.9	0.0002	1.7494	1.4273	0.0000	2.2089	inspection
17	7,388.8	0.0003	1.7473	1.4160	0.0000	0.8351	inspection
18	7,392.9	0.0003	1.7517	1.4311	0.0000	0.7725	inspection
19	7,395.4	0.0002	1.7535	1.4150	0.0000	0.5326	inspection
20	7,394.9	0.0003	1.7590	1.4260	0.0000	2.6404	inspection

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

A.2 Optimisation results

Table 16. *Optimal solution for case 1 under scenario 1*

Servicing satellite 1					
Transfer sequence	0	7	4	1	0
Time of arrival (h)	0	47.97	86.37	127.96	166.16
Time of departure (h)	22.47	63.97	106.66	143.96	–
Velocity change (m/s)	–	178.04	5.58	196.76	10.18
Fuel consumption (kg)	122.07				
Servicing satellite 2					
Transfer sequence	0	2	8	0	
Time of arrival (h)	0	62.25	100.67	147.38	
Time of departure (h)	32.35	78.57	128.88	–	
Velocity change (m/s)	–	431.08	112.74	554.80	
Fuel consumption (kg)	306.64				
Servicing satellite 3					
Transfer sequence	0	6	3	5	0
Time of arrival (h)	0	51.08	83.98	122.01	169.71
Time of departure (h)	29.28	67.08	100.51	138.01	–
Velocity change (m/s)	–	29.28	67.08	100.5	138.01
Fuel consumption (kg)	378.83				

Table 17. *Optimal solution for case 2 under scenario 1*

Servicing satellite 1								
Transfer sequence	0	4	6	7	1	11	0	
Time of arrival (h)	0	54.80	100.35	160.53	226.43	294.35	367.15	
Time of departure (h)	31.00	72.45	136.23	200.53	264.85	334.35	–	
Velocity change (m/s)	–	54.46	321.36	110.30	175.44	411.20	587.84	
Fuel consumption (kg)	425.09							
Servicing satellite 2								
Transfer sequence	0	12	2	5	0			
Time of arrival (h)	0	46.59	114.98	181.92	228.95			
Time of departure (h)	17.59	84.38	147.42	207.65	–			
Velocity change (m/s)	–	392.88	483.69	656.85	10.69			
Fuel consumption (kg)	402.32							
Servicing satellite 3								
Transfer sequence	0	13	8	14	10	3	15	0
Time of arrival (h)	0	57.55	107.35	155.71	218.71	279.02	324.82	391.65
Time of departure (h)	29.25	77.95	132.61	195.21	247.52	301.22	364.35	–
Velocity change (m/s)	–	342.32	394.78	7.09	57.94	507.94	73.11	305.24
Fuel consumption (kg)	430.39							

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 17. *(Continued)*

Servicing satellite 4			
Transfer sequence	0	9	0
Time of arrival (h)	0	44.13	106.84
Time of departure (h)	21.93	82.64	–
Velocity change (m/s)	–	88.17	81.06
Fuel consumption (kg)	54.84		

Table 18. *Optimal solution for case 3 under scenario 1*

Servicing satellite 1								
Transfer sequence	0	11	16	1	7	8	0	
Time of arrival (h)	0	37.75	96.15	162.76	201.66	239.37	278.49	
Time of departure (h)	17.15	75.85	129.26	178.76	218.77	255.89	–	
Velocity change (m/s)	–	298.59	333.14	603.38	29.69	278.28	25.94	
Fuel consumption (kg)	407.27							
Servicing satellite 2								
Transfer sequence	0	20	12	18	14	6	2	0
Time of arrival (h)	0	62.85	112.05	162.66	207.76	262.21	324.90	382.66
Time of departure (h)	27.45	87.85	140.06	185.06	234.41	298.90	354.66	–
Velocity change (m/s)	–	684.13	56.17	14.12	13.39	314.5	200.83	315.33
Fuel consumption (kg)	413.06							
Servicing satellite 3								
Transfer sequence	0	13	3	10	4	0		
Time of arrival (h)	0	58.55	119.78	157.35	205.64	246.46		
Time of departure (h)	26.25	98.18	136.95	185.04	224.16	–		
Velocity change (m/s)	–	546.18	180.71	295.66	254.22	99.61		
Fuel consumption (kg)	367.95							
Servicing satellite 4								
Transfer sequence	0	15	0					
Time of arrival (h)	0	49.63	96.44					
Time of departure (h)	26.53	73.84	–					
Velocity change (m/s)	–	25.65	25.99					
Fuel consumption (kg)	17.06							
Servicing satellite 5								
Transfer sequence	0	9	19	5	0			
Time of arrival (h)	0	64.03	127.79	176.56	230.14			
Time of departure (h)	26.33	103.39	148.36	202.54	–			
Velocity change (m/s)	–	766.85	109.85	330.91	295.4			
Fuel consumption (kg)	394.08							
Servicing satellite 6								
Transfer sequence	0	17	0					
Time of arrival (h)	0	57.82	107.85					
Time of departure (h)	32.02	86.65	–					
Velocity change (m/s)	–	186.3	226.42					
Fuel consumption (kg)	128.53							

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

***Table 19.** Optimal solution for case 1 under scenario 2*

Servicing satellite 1					
Transfer sequence	0	5	1	0	
Time of arrival (h)	0	24.26	59.54	95.16	
Time of departure (h)	14.16	53.54	85.36	–	
Velocity change (m/s)	–	11,092.96	10,984.6	10,861.39	
Fuel consumption (kg)	375.35				
Servicing satellite 2					
Transfer sequence	0	2	8	4	0
Time of arrival (h)	0	19.00	40.12	69.64	102.76
Time of departure (h)	6.00	33.22	61.64	92.76	–
Velocity change (m/s)	–	6,995.89	6,444.35	6,205.07	10,262.04
Fuel consumption (kg)	347.7				
Servicing satellite 3					
Transfer sequence	0	7	3	6	0
Time of arrival (h)	0	25.23	39.63	56.85	81.52
Time of departure (h)	18.53	31.23	50.15	70.62	–
Velocity change (m/s)	–	9,509.72	6,542.47	5,511.77	10,343.43
Fuel consumption (kg)	366.07				

***Table 20.** Optimal solution for case 2 under scenario 2*

Servicing satellite 1							
Transfer sequence	0	1	4	11	0		
Time of arrival (h)	0	38.32	63.9	104.15	161.34		
Time of departure (h)	30.42	54.10	84.55	140.74	–		
Velocity change (m/s)	–	8,425.28	10,701.47	6,070.85	7,040.24		
Fuel consumption (kg)	369.06						
Servicing satellite 2							
Transfer sequence	0	5	8	2	12	0	
Time of arrival (h)	0	41.17	72.70	107.35	119.95	152.87	
Time of departure (h)	34.07	63.40	101.35	113.95	146.47	–	
Velocity change (m/s)	–	5,456.75	8,719.15	5,818.38	7,713.32	2,986.1	
Fuel consumption (kg)	354.98						
Servicing satellite 3							
Transfer sequence	0	13	14	3	0		
Time of arrival (h)	0	32.07	63.92	85.07	128.69		
Time of departure (h)	20.77	56.62	74.97	110.99	–		
Velocity change (m/s)	–	7,137.76	8,966.72	8,660.46	6,487.38		
Fuel consumption (kg)	360.11						
Servicing satellite 4							
Transfer sequence	0	7	9	10	6	15	0
Time of arrival (h)	0	31.24	51.84	67.64	104.01	136.82	169.17
Time of departure (h)	24.94	37.24	57.84	93.81	130.02	154.37	–
Velocity change (m/s)	–	6,874.56	9,831.94	6,318.16	2,216.61	6,346.57	3,905.85
Fuel consumption (kg)	397.73						

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 21. *Optimal solution for case 3 under scenario 2*

Servicing satellite 1						
Transfer sequence	0	11	1	7	19	0
Time of arrival (h)	0	37.44	60.42	86.91	121.16	148.66
Time of departure (h)	28.44	53.22	80.01	113.66	127.96	–
Velocity change (m/s)	–	4,861.11	6,354.46	7,817.83	11,004.69	1,455.12
Fuel consumption (kg)	362.31					
Servicing satellite 2						
Transfer sequence	0	12	9	20	2	0
Time of arrival (h)	0	32.65	86.15	106.98	127.38	172.08
Time of departure (h)	26.65	72.65	94.58	112.98	157.28	–
Velocity change (m/s)	–	7,514.49	10,560.13	11,020.25	10,631.24	5,140.64
Fuel consumption (kg)	473.21					
Servicing satellite 3						
Transfer sequence	0	13	3	5	8	0
Time of arrival (h)	0	46.66	77.92	95.52	117.15	155.27
Time of departure (h)	25.06	66.72	86.72	109.55	148.77	–
Velocity change (m/s)	–	4,613.19	4,410.92	7,511.27	5,444.49	3,059.29
Fuel consumption (kg)	300.72					
Servicing satellite 4						
Transfer sequence	0	16	18	0		
Time of arrival (h)	0	22.30	45.97	86.05		
Time of departure (h)	16.3	33.77	76.75	–		
Velocity change (m/s)	–	10,281.83	12,652.83	9,739.67		
Fuel consumption (kg)	372.98					
Servicing satellite 5						
Transfer sequence	0	17	14	0		
Time of arrival (h)	0	36.71	97.21	137.35		
Time of departure (h)	28.91	76.71	127.25	–		
Velocity change (m/s)	–	10,107.8	11,147.5	6,330.91		
Fuel consumption (kg)	325.71					
Servicing satellite 6						
Transfer sequence	0	6	4	15	10	0
Time of arrival (h)	0	51.10	65.51	96.87	133.16	174.97
Time of departure (h)	40.00	58.61	90.77	127.16	165.47	–
Velocity change (m/s)	–	4,323.86	2,632.99	6,961.52	9,636.04	11,393.2
Fuel consumption (kg)	393.02					

Table 22. *Optimal solution for case 1 under scenario 3*

Servicing satellite 1					
Transfer sequence	0	1	8	4	0
Time of arrival (h)	0	11.88	25.12	37.28	48.36
Time of departure (h)	6.78	20.02	30.98	42.36	–
Velocity change (m/s)	–	440.82	422.66	484.97	472.00
Fuel consumption (kg)	454.92				
Servicing satellite 2					
Transfer sequence	0	2	6	0	
Time of arrival (h)	0	12.66	26.79	38.43	
Time of departure (h)	6.86	21.79	31.93	–	
Velocity change (m/s)	–	465.55	420.80	504.47	
Fuel consumption (kg)	370.99				
Servicing satellite 3					
Transfer sequence	0	5	3	7	0
Time of arrival (h)	0	13.73	25.28	39.99	54.63
Time of departure (h)	8.33	20.28	34.69	49.53	–
Velocity change (m/s)	–	447.12	428.10	431.75	426.64
Fuel consumption (kg)	438.91				

Table 23. *Optimal solution for case 2 under scenario 3*

Servicing satellite 1						
Transfer sequence	0	11	15	1	8	0
Time of arrival (h)	0	14.98	26.06	40.90	55.16	68.66
Time of departure (h)	9.68	21.06	35.20	49.16	62.96	–
Velocity change (m/s)	–	355.59	336.66	368.50	375.58	367.99
Fuel consumption (kg)	451.98					
Servicing satellite 2						
Transfer sequence	0	2	5	12	9	0
Time of arrival (h)	0	15.60	28.67	40.20	51.62	67.32
Time of departure (h)	10.00	23.47	33.90	46.62	61.62	–
Velocity change (m/s)	–	365.88	348.08	377.43	336.90	382.68
Fuel consumption (kg)	453.19					
Servicing satellite 3						
Transfer sequence	0	3	13	14	0	
Time of arrival (h)	0	11.72	27.14	39.33	52.96	
Time of departure (h)	6.52	21.24	34.23	46.56	–	
Velocity change (m/s)	–	339.98	380.64	348.07	385.76	
Fuel consumption (kg)	384.19					
Servicing satellite 4						
Transfer sequence	0	7	6	4	10	0
Time of arrival (h)	0	12.36	28.76	43.75	54.75	69.50
Time of departure (h)	7.06	22.36	38.05	49.65	64.10	–
Velocity change (m/s)	–	348.88	397.81	359.76	337.53	368.69
Fuel consumption (kg)	453.50					

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

Table 24. *Optimal solution for case 3 under scenario 3*

Servicing satellite 1						
Transfer sequence	0	1	9	11	18	0
Time of arrival (h)	0	15.60	29.63	43.25	54.71	65.29
Time of departure (h)	10.00	24.43	36.75	49.51	59.99	–
Velocity change (m/s)	–	355.61	344.67	385.97	361.14	363.35
Fuel consumption (kg)	453.15					
Servicing satellite 2						
Transfer sequence	0	10	2	12	15	0
Time of arrival (h)	0	11.31	25.96	42.06	57.57	71.91
Time of departure (h)	5.51	20.06	35.96	51.87	65.91	–
Velocity change (m/s)	–	367.24	365.99	389.73	378.21	370.61
Fuel consumption (kg)	464.17					
Servicing satellite 3						
Transfer sequence	0	13	14	3	0	
Time of arrival (h)	0	10.68	22.92	35.47	48.57	
Time of departure (h)	5.28	16.62	29.67	43.57	–	
Velocity change (m/s)	–	349.76	391.90	380.10	345.11	
Fuel consumption (kg)	386.73					
Servicing satellite 4						
Transfer sequence	0	4	7	20	0	
Time of arrival (h)	0	14.24	27.10	39.67	50.92	
Time of departure (h)	8.64	21.10	34.47	45.12	–	
Velocity change (m/s)	–	357.77	378.53	347.56	381.47	
Fuel consumption (kg)	386.42					
Servicing satellite 5						
Transfer sequence	0	5	8	16	0	
Time of arrival (h)	0	14.29	28.63	42.79	55.50	
Time of departure (h)	9.09	22.13	37.59	49.70	–	
Velocity change (m/s)	–	345.08	380.31	346.75	364.61	
Fuel consumption (kg)	380.55					
Servicing satellite 6						
Transfer sequence	0	19	17	6	0	
Time of arrival (h)	0	14.83	28.32	43.55	56.89	
Time of departure (h)	9.73	23.22	37.85	50.29	–	
Velocity change (m/s)	–	362.67	340.88	385.87	385.32	
Fuel consumption (kg)	388.34					

A.3 Orbital element vector of an target after time Δt

Given a target state $x\left(t_0\right)$ at t_0 and a time interval Δt, we can calculate $x\left(t_0+\Delta t\right)$ using the following procedure [27]. Given the known true anomaly $\theta\ \left(t_0\right)$ at the moment t_0 and the equation $\tan\frac{E}{2}=\sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}$, we can derive the eccentric anomaly $E\left(t_0\right)$ as follows:

$$E\left(t_0\right)=2\tan^{-1}\left(\sqrt{\frac{1-e}{1+e}}\tan\frac{\theta\ \left(t_0\right)}{2}\right) \tag{A1}$$

Following Kepler's equation, we can express the mean anomaly at the moment t_0 as

$$M_e\left(t_0\right)=E\left(t_0\right)-e\sin E\left(t_0\right) \tag{A2}$$

The period of an elliptical or circular orbit can be represented as

$$T=\frac{2\pi}{\sqrt{\mu}}a^{3/2} \tag{A3}$$

where $\mu=398,600\,\mathrm{km}^3/\mathrm{s}^2$ represents the gravitational parameter for earth, and a signifies either the semimajor axis of an elliptical orbit or the radius of a circular orbit. Therefore, the mean anomaly at $t_0+\Delta t$ can be expressed as

$$M_e\left(t_0+\Delta t\right)=M_e\left(t_0\right)+\Delta t\frac{2\pi}{\sqrt{\mu}}a^{3/2} \tag{A4}$$

Given $M_e\left(t_0+\Delta t\right)$, and relying on the equality $M_e=E-e\sin E$, we can employ an iterative method to solve for $E\left(t_0+\Delta t\right)$. Ultimately, we can determine $\theta\ \left(t_0+\Delta t\right)$ from equation $\tan\frac{E}{2}=\sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}$ as follows:

$$\theta\ \left(t_0+\Delta t\right)=2\tan^{-1}\left(\sqrt{\frac{1+e}{1-e}}\tan\frac{E\left(t_0+\Delta t\right)}{2}\right) \tag{A5}$$

The equation above necessitates that no manoeuvers take place within the Keplerian orbit, ensuring that no orbital elements other than the true anomaly undergo changes. We denote the functional relationship between $x\left(t_0\right)$ and $x\left(t_0+\Delta t\right)$ as

$$x\left(t_0+\Delta t\right)=\mathcal{G}\left(x\left(t_0\right),\Delta t\right) \tag{A6}$$

A.4 Lambert-based manoeuver

Provided with the initial position vector $\boldsymbol{r}_1$, final position vector $\boldsymbol{r}_2$, and the transfer time Δt of the transfer orbit, the necessary velocity changes for the two manoeuvers can be computed using the following procedure [27].

With knowledge of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, the change in the true anomaly $\Delta\theta$ formed by the initial and final position vectors on the transfer orbit satisfies $\cos\Delta\theta=\frac{\boldsymbol{r}_1\cdot\boldsymbol{r}_2}{r_1r_2}$, where r_1 and r_2 represent the magnitudes of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, respectively. Two scenarios must be taken into account: prograde trajectories ($0^\circ<i<90^\circ$) and retrograde trajectories ($90^\circ<i<180^\circ$), where i denotes the inclination of the transfer orbit. Hence, the solution for $\Delta\theta$ can be formulated as

$$\Delta\theta=\begin{cases}\cos^{-1}\left(\frac{\boldsymbol{r}_1\cdot\boldsymbol{r}_2}{r_1r_2}\right) & \text{if } \sin\Delta\theta\cos i\geq 0 \quad \text{prograde}\\ 360^\circ-\cos^{-1}\left(\frac{\boldsymbol{r}_1\cdot\boldsymbol{r}_2}{r_1r_2}\right) & \text{if } \sin\Delta\theta\cos i<0 \quad \text{prograde}\\ \cos^{-1}\left(\frac{\boldsymbol{r}_1\cdot\boldsymbol{r}_2}{r_1r_2}\right) & \text{if } \sin\Delta\theta\cos i<0 \quad \text{retrograde}\\ 360^\circ-\cos^{-1}\left(\frac{\boldsymbol{r}_1\cdot\boldsymbol{r}_2}{r_1r_2}\right) & \text{if } \sin\Delta\theta\cos i\geq 0 \quad \text{retrograde}\end{cases} \tag{A7}$$

https://doi.org/10.1017/aer.2024.142 Published online by Cambridge University Press

By defining the velocity vectors at the initial and final positions on the transfer orbit as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, respectively, the following relationship holds:

$$\begin{aligned} \boldsymbol{r}_2 &= f\boldsymbol{r}_1 + g\boldsymbol{v}_1 \\ \boldsymbol{v}_2 &= \dot{f}\boldsymbol{r}_1 + \dot{g}\boldsymbol{v}_1 \end{aligned} \tag{A8}$$

where Lagrange coefficients f, g and their derivatives $\dot{f}$, $\dot{g}$ are given by

$$\begin{aligned} &f = 1 - \tfrac{\chi^2}{r_1} C\,(z) \\ &g = \Delta t - \tfrac{1}{\sqrt{\mu}} \chi^3 S\,(z) \\ &\dot{f} = \tfrac{\sqrt{\mu}}{r_1 r_2} \chi\,(zS\,(z) - 1) \\ &\dot{g} = 1 - \tfrac{\chi^2}{r_2} C\,(z) \end{aligned} \tag{A9}$$

In the equations above, z is represented as $z = \alpha\chi^2$, where α and χ denote the reciprocal of the semimajor axis and the universal anomaly of the transfer orbit, respectively. The Stumpff function $S\,(z)$ and $C\,(z)$ are defined as

$$S\,(z) = \begin{cases} \frac{\sqrt{z} - \sin\sqrt{z}}{(\sqrt{z})^3}(z > 0) \\ \frac{\sinh\sqrt{-z} - \sqrt{-z}}{(\sqrt{z})^3}(z < 0) \\ \frac{1}{6}(z = 0) \end{cases}, \quad C\,(z) = \begin{cases} \frac{1 - \cos\sqrt{z}}{z}(z > 0) \\ \frac{\cosh\sqrt{-z} - 1}{-z}(z < 0) \\ \frac{1}{2}(z = 0) \end{cases} \tag{A10}$$

To determine $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, it is essential to calculate the unknown variables z and χ in (A9). This have been previously solved in the Ref. [27] using analytical relationships related to orbital mechanics and recursive methods. Their derivation will not be reiterated here. Given $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and Δt, Lambert's theorem is applied to determine $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. This functional relationship is expressed as:

$$\boldsymbol{v}_1 = \mathcal{L}_1\,(\boldsymbol{r}_1, \boldsymbol{r}_2, \Delta t)\,, \boldsymbol{v}_2 = \mathcal{L}_2\,(\boldsymbol{r}_1, \boldsymbol{r}_2, \Delta t) \tag{A11}$$

Cite this article: Zhang J., Xia H. and Li L. Optimal scheduling for many-to-many on-orbit service manoeuvers considering variances in target accessibility. *The Aeronautical Journal*, https://doi.org/10.1017/aer.2024.142