Fuel-Efficient on-Orbit Service Vehicle Allocation Based on an Improved Discrete Particle Swarm Optimization Algorithm

Wu Jian, Liu Qingguo, Liu Xinxue, and Li Yaxiong

Xi'an High-tech Institute, Xi'an 710025, China

Correspondence should be addressed to Liu Qingguo; teamalpha@163.com

Received 1 October 2019; Revised 1 March 2020; Accepted 20 March 2020; Published 15 April 2020

Copyright © 2020 Wu Jian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Given the limited fuel capacity of an on-orbit service vehicle (OSV), proper OSV allocation to satellites during each service mission is critical for economic fuel consumption. This allocation problem can be formulated as an optimization problem with many continuous and discrete design variables of wide domains. This problem can be effectively handled through the proposed approach that combines the tabu search with the discrete particle swarm optimization algorithm (DPSO-TS). First of all, Pontryagin’s minimum principle and genetic algorithm (GA) are exploited to find the most fuel-efficient transfer trajectory. This fuel efficiency maximization can then serve as the performance index of the OSV allocation optimization model problem. In particular, the maximization of the minimum residual fuel over individual OSVs is proposed as a performance index for OSV allocation optimization. The optimization problem is numerically solved through the proposed DPSO-TS algorithm. Finally, the simulation results demonstrate that the DPSO-TS algorithm has a higher accuracy compared to the DPSO, the DPSO-PDM and the DPSO-CSA algorithms in the premise that these four algorithms have the basically same computational time. The DPSO-TS algorithm can effectively solve the OSV allocation optimization problem.

1. Introduction

As humans explore space deeper, more and more diverse satellites are deployed into space. On-orbit service of satellites is a type of space operations performed by astronauts, robots, or both to extend the lives of satellites and improve their abilities to perform various tasks [1]. On-orbit service vehicles (OSV) are particularly useful as an effective tool to extend lives of satellites. These vehicles perform several tasks including on-orbit assembly, maintenance, and logistic support [2]. Many important problems of the OSV are investigated in the existing literature. The study of transfer trajectory is the basic work of allocation of OSVs. Direct methods [3–5], indirect methods [6, 7], and hybrid methods [8, 9] are used to study the optimization of transfer trajectories of different types of space vehicles. The hybrid methods, one of which is adopted in this paper to optimize the transfer trajectories, combine the advantages of the direct and indirect methods.

In general, multiple satellites need to be maintained in one service mission. The allocation of OSVs to satellites is an important problem that has the characteristics of a general resource allocation problem. However, the OSV allocation problem is distinguished by two aspects: (a) there are many optimization variables with a wide search scope for each, and (b) the optimization variables are of both the continuous and discrete types. Allocation optimization problem is a complex combinatorial optimization problem and known to be an NP hard, and discrete particle swarm optimization (DPSO) is a commonly used algorithm to solve allocation optimization problems. The traditional DPSO algorithm [10–12] is easy to be trapped in the local optimum, and hence some improved DPSO algorithms are studied. Inertia weight of DPSO is used to balance the global and local search capacity, and then the way of designing the adjustment strategy of inertia weight to solve the problem of falling into local optimum is widely studied. Li et al. [13] propose DPSO-PDM algorithm by the hybridizing adjustment strategy based on new particle
2. Transfer Trajectory Optimization Based on a Hybrid Method

In this section, we provide the differential equations of motion of an OSV in an Earth-centred inertial (ECI) coordinate system. Then, an optimization model of the OSV transfer trajectory is established.

2.1. OSV Equations of Motion. The differential equations of motion [25] are

\[
\begin{align*}
\dot{r} &= v, \\
\dot{\psi} &= -\frac{\mu}{|r|^3} r + \frac{F}{m} \alpha, \\
\dot{m} &= -\frac{F}{(g_0 I_{sp})},
\end{align*}
\]

where \( r \) is the position vector, \( v \) is the velocity vector, \( F \) is the thrust, \( m \) is the OSV mass, \( I_{sp} \) is the specific impulse of the engine, and \( \alpha = [\alpha_x, \alpha_y, \alpha_z]^T \) is the unit vector of the thrust direction.

2.2. Transfer Trajectory Optimization. Assuming that the maneuvering time of each OSV is known or has been already determined, an optimization model for the transfer trajectory between each pair of an OSV and a satellite is established based on a hybrid method, which exploits Pontryagin’s minimum principle [26, 27] and genetic algorithm (GA) [28]. The more fuel one OSV has retained, the longer it can stay in the orbit and the more satellites it can serve. Therefore, the mass of the remaining fuel can be taken as the optimization index to be maximized, namely,

\[
\max f = m(t_f),
\]

where \( t_f \) is the terminal time of each transfer trajectory and \( m \) is a function of the change in the OSV fuel mass.

Based on the OSV equations of motion in formula (1), the Hamilton function can be derived as

\[
H = \lambda_r v + \lambda_\psi \left( \frac{\mu}{|r|^3} r + \frac{F}{m} \alpha \right) - \lambda_m \frac{F}{(g_0 I_{sp})},
\]

where \( \lambda_r \) and \( \lambda_\psi \) are the adjoint variables of the state variables and \( \lambda_m \) is the adjoint variable of the OSV mass \( m \).

According to Pontryagin’s minimum principle, the optimal thrust direction \( \alpha^* \) is obtained by solving \( ||\alpha|| = 1 \) and \( \partial H / \partial \alpha = 0 \). Then, the optimal thrust direction is

\[
\alpha^* = \frac{\lambda_\psi}{||\lambda_\psi||}
\]

The partial derivatives of the Hamiltonian with respect to the adjoint variables are thus obtained. Consequently, the adjoint variables can be shown to satisfy the following differential equations:
Table 1: Nomenclature of variables.

| Variable | Explanation |
|----------|-------------|
| $\mathbf{r}$ | Position vector |
| $\mathbf{v}$ | Velocity vector |
| $F$ | Thrust |
| $m$ | OSV mass |
| $I_{sp}$ | Specific impulse |
| $\alpha$ | Thrust direction |
| $t_f$ | Terminal time of transfer trajectory |
| $\lambda_t$ | Adjoint variables of position vector |
| $\lambda_v$ | Adjoint variables of velocity vector |
| $\lambda_m$ | Adjoint variable of the OSV mass |
| $\alpha^*$ | Optimal thrust direction |
| $F_{\max}$ | Maximum of thrust |
| $t_{on}$ | Switching-on time of the 2nd thrust segment |
| $t_{off}$ | Switching-off time of the 2nd thrust segment |
| $\beta$ | Design parameters of transfer trajectory |
| $P_T$ | Population size of GA |
| $P_c$ | Crossover probability of GA |
| $P_m$ | Mutation probability of GA |
| $N_f$ | Maximum iterations of GA |
| $N$ | Count of OSVs |
| $M$ | Count of satellites |
| $S$ | Denotation of OSV |
| $\mathbf{X}$ | Allocation matrix |
| $t_m$ | Maneuvering time |
| $m_{\text{remain}}$ | Remaining fuel mass of $O_k$ after maintaining |
| $M_G$ | Population size of DPSO |
| $K$ | Number of satellites needed to be maintained |
| $T_D$ | Maximum iterations of DPSO |
| $\omega$ | Inertia weight |
| $c_1$ | Cognitive coefficient |
| $c_2$ | Social coefficient |
| $h$ | Iteration counter |
| $\delta$ | Time step |
| $t_{\max}$ | Mission duration |
| $p_i^h$ | Individual fitness extreme |
| $p_i^g$ | Global fitness extreme |
| $l_T$ | Length of the tabu table |
| $M_T$ | Size of the candidate set |
| $T_T$ | Maximum iterations of TS |
| Openlist | Candidate set |
| Tabulist | Tabu table |

\[
\dot{\lambda}_t = \frac{\mu}{|\mathbf{r}|^3} \lambda_t - \frac{3 \mu \cdot \lambda_t}{|\mathbf{r}|^5} \mathbf{r}, \\
\dot{\lambda}_v = -\lambda_v, \\
\dot{\lambda}_m = -\frac{F}{m^2} \|\lambda\|.
\]  

Initial boundary conditions are
\[
\begin{align*}
\mathbf{r}(t_0) &= r_0, \\
\mathbf{v}(t_0) &= v_0, \\
m(t_0) &= m_0.
\end{align*}
\]  

and the terminal boundary constraints are
\[
\begin{align*}
\mathbf{r}(t_f) &= r_f, \\
\mathbf{v}(t_f) &= v_f.
\end{align*}
\]

The optimal control under finite-thrust conditions is the switching mode of the engine strategy (bang-bang control). In this paper, the adopted switching mode is the on-off-on mode of the engine working sequence, and the switching moments are taken as the optimization variables. In the switching-on segment, the thrust is set to $F_{\max}$. In the switching-off segment, the thrust is set to zero.

For the GA-based hybrid method, the orbital optimization problem is converted into an optimization problem with parameters, including the initial values of the adjoint variables, the switching-off time $t_{off}$ of the first thrust segment, the switching-on time $t_{on}$ of the second thrust segment, and the switching-off time $t_{off}^2$ of the second thrust segment. For maximizing the mass of the remaining OSV fuel under the given constraints, seventeen parameters $\beta = [\lambda_1(t_0), \lambda_2(t_0), t_{on}^1, t_{off}^1, t_{on}^2, t_{off}^2]$ are taken as optimization variables. The solution of this optimization problem using genetic algorithm is carried out as follows:

Step 1: the design parameters $\beta = [\lambda_1(t_0), \lambda_2(t_0), t_{on}^1, t_{off}^1, t_{on}^2, t_{off}^2]$ are encoded. Thirty chromosomes are generated randomly to form the initial population. The population size $P_T$, the crossover probability $P_c$, the mutation probability $P_m$, and the maximum number of iterations $N_f$ are set.

Step 2: for each chromosome, the equations are solved to get the optimal thrust direction under the conditions (6) and (7). Then, the equations of motion (1) are solved to obtain the transfer trajectory. The fitness values of all chromosomes in the current generation are computed based on the fitness function in formula (2).

Step 3: for the next generation, chromosomes are selected using the roulette-wheel selection method. Some genes on two different chromosomes reciprocally cross according to the crossover probability and mutate according to the mutation probability. The execution of the selection, crossover, and mutation operations produces the next generation population.

Step 4: steps 2 and 3 are repeated until the GA fitness function values go below $\epsilon = 10^{-6}$ or the maximum number of iterations $N$ is reached.

The hybrid method results in the OSV transfer trajectory with the maximum mass of the remaining fuel at a given maneuvering moment.

### 3. OSV Allocation Optimization Based on the DPSO-TS Algorithm

The OSV allocation optimization problem can be solved through the DPSO algorithm. The inertia weight $\omega$ is an important parameter of the DPSO algorithm. The larger the inertia weight $\omega$ is, the stronger the global search capability is. TS algorithm is a hill-climbing algorithm with good local
search performance, and it improves the efficiency of DPSO algorithm by exploiting the neighborhood of the results obtained by DPSO algorithm. In order to keep the balance between exploration and exploitation in DPSO-TS algorithm, we set the inertia weight $\omega$ to a larger value to get strong exploration ability. We propose to use TS algorithm in each iteration of DPSO algorithm to improve local search capacity on conditions that the exploitation ability is limited by setting the appropriate length of the tabu table.

### 3.1. Mathematical Optimization Model

Assume there are $N$ OSVs denoted by $O = \{O_1, O_2, \ldots, O_N\}$, and $M$ satellites denoted by $S = \{S_1, S_2, \ldots, S_M\}$. Each OSV serves at most a satellite in one mission, and each satellite can be served at most once. The allocation matrix $X$ can be written as

$$
x_{ij}(t_m) = \begin{cases} 
1, & O_i \text{ serving } S_j \text{ with manoeuvring time } t_m, \\
0, & O_i \not\text{ serving } S_j \text{ with manoeuvring time } t_m,
\end{cases}
$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$ and $t_m$ is manoeuvring time. $X$ includes the allocation plan of OSVs and their manoeuvring times. The mathematical model of OSV allocation problem can be written as

$$
\text{max } f(X) = \text{max} \left\{ \min \{m^i_{\text{remain}}(X), m^i_{\text{remain}}(X), \ldots, m^i_{\text{remain}}(X)\} \right\},
$$

$$
\begin{align*}
\sum_{j=1}^{M} x_{ij}(t_m) & \leq 1, \quad \forall i = 1, 2, \ldots, N, \\
0 & < \sum_{j=1}^{N} x_{ij}(t_m) \leq 1, \quad \forall j = 1, 2, \ldots, M \\
m^k_{\text{remain}} & \geq m_{\text{keep}}, \quad \forall k = 1, 2, \ldots, N,
\end{align*}
$$

where $m^k_{\text{remain}}$ is the remaining fuel mass of the $k^{th}$ OSV after maintaining, and $m_{\text{keep}}$ is the mass of the fuel used to keep the orbit. The inner min operator $\min \{m^i_{\text{remain}}(X), m^i_{\text{remain}}(X), \ldots, m^i_{\text{remain}}(X)\}$ returns the minimum mass of the remaining fuel among individual OSVs used to maintain satellites according to $X$. The outer max operator $\max \{\min \{m^i_{\text{remain}}(X), m^i_{\text{remain}}(X), \ldots, m^i_{\text{remain}}(X)\}\}$ searches for the allocation plan $X$ that maximizes the minimum mass value. The optimization index reflects the overall service capability of OSVs.

### 3.2. Optimization Problem Solution Based on the DPSO-TS Algorithm

The OSV allocation optimization problem can be solved using the proposed DPSO-TS algorithm. Details of the DPSO-TS algorithm are given for two parts: the implementation steps of DPSO-TS algorithm and DPSO algorithm and some steps and strategies of DPSO-TS algorithm.

#### 3.2.1. The Implementation Steps of DPSO-TS Algorithm

The implementation steps of DPSO-TS algorithm are

1. **Step 1:** set the DPSO-TS parameters including the population size $M_B$ of the particle swarm, the OSV count $N$, the number of satellites $K$ need to be maintained, the maximum number of iterations $T_D$, the inertia weight $\omega$, the cognitive coefficient $c_1$, the social coefficient $c_2$, and the iteration counter $h = 1$. Select a certain time step $\delta$ to discretize the mission duration $t_{\text{max}}$.

2. **Step 2:** initialize the OSV particle swarm population. Set parameter values for each particle including an OSV serial number and a randomly generated maneuvering moment. For computational efficiency, maneuvering moments and OSV numbers of OSVs are separately used for the DPSO update by formula (12). The fitness function values $f(P_i)$ of all particles are calculated, and the individual fitness extreme $P_i^h$ and the global fitness extreme $P^h$ are recorded.

3. **Step 3:** the particle positions are updated by formula (12). The fitness function values $f(P_i)$ of all particles are calculated. The individual fitness extreme $P_i^h$ and the global fitness extreme $P^h$ are updated.

4. **Step 4:** the parameters of TS algorithm are set including the length $l_T$ of the tabu table, the size $M_T$ of the candidate set, and the maximum number of iterations $T_T$.

5. **Step 5:** set the initial solution $P_T = P^h$ and the optimal solution $P_{T_{\text{best}}} = P^h$ of the TS algorithm.

6. **Step 6:** according to the centrality and diversity search strategies, generate the neighborhood of the current solution, and select $N_T$ particles from the neighborhood as the candidate set Openlist($Y$).

7. **Step 7:** according to the fitness function value $f(Y)$, judge whether the amnesty criterion is met. If so, the corresponding candidate solution $Y_i$ will be placed in the tabu table Tabulist($Y$) and the solution in the tabu table will be released. Otherwise, keep the tabu table intact.

8. **Step 8:** check whether the TS terminal conditions are satisfied. If so, output $P^{h_{\text{opt}}} = P_{T_{\text{best}}}$. Otherwise, go to Step 6.

9. **Step 9:** increase the iteration number, $h = h + 1$. Check whether the DPSO termination conditions are satisfied. If so, output $P^{h_{\text{opt}}}$ Otherwise, go to Step 3.

The flow chart of DPSO-TS algorithm is shown in Figure 1.

#### 3.2.2. Some Steps and Strategies of DPSO-TS Algorithm

(1) **Coding of Particles.** Particles of the OSV allocation problem are encoded in a decimal system where each particle has a length of $2 \times N$. Each particle is encoded as shown in Figure 2, where $O_{i_1}, \ldots, O_{i_N}$ are $N$ OSVs, and $t_{s_1}, \ldots, t_{s_N}$ are the corresponding maneuvering moments. The value of $O_i$ indicates the number of satellites served by the $i^{th}$ OSV. This value belongs to the set $\{0, 1, 2, \ldots, M\}$, where 0 means no satellite is served. We set the duration of the mission to $[0, t_{\text{max}}]$. If $O_i = 0$, set $t_i = -1$. 

$P_i^h = X_i^h$, calculate $P_g^h$

Terminal conditions?

Generate random numbers $r_1$, $r_2$ and $r_3$

$r_1 \geq \omega$, $r_2 \geq c_1$, $r_3 \geq c_2$

Update $X_i^{h+1} = c_2 \otimes F_3[c_1 \otimes F_3[w \otimes F_1(X_i^h), P_i^h], P_g^h]$

If $fit(X_i^{h+1}) < fit(P_i^h)$, $P_i^{h+1} = X_i^{h+1}$; otherwise $P_i^{h+1} = P_i^h$

If $fit(X_i^{h+1}) < fit(P_g^h)$, $P_g^{h+1} = X_i^{h+1}$; otherwise $P_g^{h+1} = P_g^h$

Set the TS parameters, $i = 1$

Set initial solution $PT = P_g^h$ and the optimal solution $PT_{best} = P_g^h$

Terminal conditions of TS?

Generate neighborhood and select $N_T$ particles as the candidate set $Openlist(Y)$

Calculate $fit_T(Y_i)$

Amnesty criterion?

Corresponding candidate solution $Y_i$ is placed in the tabu table

Output $PT_{best}$ and update tabu table, $i = i + 1$

$h = h + 1$

Output $P_g^h$

End

Start

Figure 1: Flow chart of DPSO-TS.

Figure 2: Coding of a particle in the OSV allocation problem.
(2) Calculation of the Fitness Function. Objective function (9) is taken as the fitness function for the particles

\[
\max f(P_s) = \max \left[ \min \left\{ m_{\text{remain}}(P_s), m_{\text{remain}}'(P_s), \ldots, m_{\text{remain}}'(P_s) \right\} \right], \quad (s \leq N),
\]

where \( P_s \) is the \( i \)th particle.

(3) Particle Position and Velocity Update. The position of a particle is determined by the particle velocity, the individual fitness extreme, and the global fitness extreme [29]. The particle position and velocity in a PSO algorithm are updated as

\[
\begin{align*}
v_{h+1}^i &= \omega \cdot v_i^h + c_1 \cdot r_1 \cdot (P_i^h - x_i^h) + c_2 \cdot r_2 \cdot (P_g^h - x_i^h), \\
\xi_{h+1}^i &= \xi_i^h + v_{h+1}^i,
\end{align*}
\]

where \( P_i^h \) is the individual fitness extreme of the \( i \)th particle in the \( h \)th iteration; \( P_g^h \) is the global fitness extreme in the \( h \)th iteration; \( \xi_i^h \) is the fitness function value of the \( i \)th particle in the \( h \)th iteration; \( v_{h+1}^i \) is the fitness function value of the \( i \)th particle in the \( (h+1) \)th iteration; \( v_i^h \) is the particle velocity value of the \( i \)th particle in the \( h \)th iteration; \( \omega \) is the inertia weight which reflects the extent to which a particle keeps the current velocity; \( c_1 \) is the cognitive coefficient, which measures how far a particle gets close to the individual fitness extreme; \( c_2 \) is the social coefficient, which measures how far a particle gets close to the global fitness extreme; and both \( r_1 \) and \( r_2 \) are random numbers in the interval [0, 1].

On the basis of the PSO, we propose the following DPSO update rules for the particle position and velocity:

\[
P_{s+1} = c_2 \otimes F_1(c_1 \otimes F_2(\omega \otimes F_1(P_i^h), P_i^h), P_g^h)_{\xi_i^h},
\]

where \( P_i^h \) and \( P_{s+1} \) are the fitness function values of the \( i \)th particle in the \( h \)th and \( (h+1) \)th iterations, respectively. Formula (12) includes three formulas which are written as formulas (13)–(15).

\[
\Psi_i^h = \omega \otimes F_1(P_i^h) = \begin{cases} F_1(P_i^h), & \text{rand} < \omega, \\ P_i^h, & \text{rand} \geq \omega, \end{cases} \quad (13)
\]

\[
\Phi_i^h = c_1 \otimes F_2(\psi_i, P_i^h) = \begin{cases} F_2(\psi_i, P_i^h), & \text{rand} < c_1, \\ \psi_i, & \text{rand} \geq c_1, \end{cases} \quad (14)
\]

\[
P_{s+1}^{h+1} = c_2 \otimes F_3(\Phi_i^h, P_g^h) = \begin{cases} F_3(\Phi_i^h, P_g^h), & \text{rand} < c_2, \\ \Phi_i^h, & \text{rand} \geq c_2. \end{cases} \quad (15)
\]

In formula (13), \( \Psi_i^h \) is the value of the \( i \)th particle in the \( h \)th iteration and is obtained via the product \( \omega \otimes F_1(P_i^h) \). The function rand generates a random number from a uniform distribution on [0, 1]. If rand < \omega, the function \( F_1(P_i^h) \) is computed leading to the random generation of two integers \( a \) and \( b \), from the set \( \{0, 1, 2, \ldots, M\} \), and hence the interchange of the values of the \( i \)th particle at positions \( a \) and \( b \). If rand \geq \omega, the output is \( P_i^h \), which means that the coding values of the \( i \)th particle are kept unchanged.

In formula (14), \( \Phi_i^h \) is the value of the \( i \)th particle in the \( h \)th iteration, and is obtained via the product \( c_1 \otimes F_2(\psi_i, P_i^h) \), where \( P_i^h \) is the individual fitness extreme position in the \( h \)th iteration and is obtained via the product \( \omega \otimes F_1(P_i^h) \). In this paper, the initial fitness function is computed leading to the random generation of two integers \( a \) and \( b \), from the set \( \{0, 1, 2, \ldots, M\} \), and hence the interchange of the values of \( \Psi_i^h \) and \( P_{s+1}^{h+1} \) at positions \( a \) and \( b \). If rand \geq c_1, the output is \( \Phi_i^h \), which means that the coding values of \( \Psi_i^h \) are kept unchanged.

In formula (15), \( P_{s+1}^{h+1} \) is the global fitness extreme position in the \( h \)th iteration. If rand < \( c_2 \), function \( F_3(\Phi_i^h, P_g^h) \) is computed leading to the random generation of two integers \( a \) and \( b \), from the set \( \{0, 1, 2, \ldots, M\} \), and hence the interchange of the values of \( \Phi_i^h \) and \( P_g^h \) at positions \( a \) and \( b \). If rand \geq c_2, the output is \( \Phi_i^h \), which means that the coding values of \( \Phi_i^h \) are kept unchanged.

(4) Some Strategies of TS Algorithm. In this paper, the initial solution of the TS algorithm is the global extremum. \( P_i \) in each iteration of the DPSO algorithm. The fitness function is written as in formula (10). TS algorithm includes some strategies: if the solution is superior to the current optimal solution, the optimal solution should be amnesty; tabu tables which are first-in first-out (FIFO) queue are used to prevent repeated search for visited solutions and to avoid falling into local loops; the centralized search strategy refers to strengthening search in the neighborhood of good solutions. A diversity search strategy looks for solutions in unknown or highly unexplored regions.

4. Simulations and Analyses of Results

4.1. Simulations. The parking orbit elements of satellites (at 0:00 on July 1, 2099) in need of service are shown in Table 2. The parking orbit elements of the OSVs (at 0:00 on July 1, 2099) are shown in Table 3. The maintaining time for a satellite is an hour, and the on-orbit service mission should be achieved in 24 hours (from 0:00 on July 1, 2099 to 0:00 on July 2, 2099), which means \( t_m + t_{tra} \) (\( t_{tra} \) is the duration of OSV and satellite from maneuvering time to rendezvous time) is in [0, 23 × 3600 s]. The mass of each OSV is 1000 kg, each OSV carries 600 kg fuel the engine thrust is 490 N, and the gas jet velocity of 3000 m/s. \( m_{\text{eop}} \) is 100 kg. The population size \( P_T \) is 20, the crossover probability \( P_c \) is 0.7, the mutation probability \( P_m \) is 0.1, and the maximum number of iterations \( N_i \) is 30. The population size \( M_g \) is 20, maximum
number of iterations $T_F$ is 30, the inertia weight $\omega$ is 0.8, the cognitive coefficient $c_1$ is 0.6, and the social coefficient $c_2$ is 0.6. The length $l_T$ of the tabu table is 2, the size $M_T$ of candidate is 5, and the maximum number of iterations $T_F$ is 8. Time step $\delta$ is 1 s, and $I = t_m/\delta$. The simulation performs ten independent experiments. In order to reduce computational time, each trajectory is computed by one independent blade of blade servers.

We design comparison experiments with DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm. The parameters of DPSO algorithm are the same as those of DPSO-TS algorithm. The parameters of DPSO algorithm, part of DPSO-PDM algorithm, are the same as those of DPSO-TS algorithm, and the other parameters are the same as those in the article written by Li et al. [13]. The parameters of the DPSO algorithm part of CHIDPSO algorithm are the same as those of DPSO-TS algorithm, and the other parameters are the same as those in the article written by Vairam et al. [17].

4.2. Analyses of Results. Results including allocation plans, computational time, and the mass of the remaining fuel obtained by DPSO-TS algorithm, DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm in ten independent experiments are shown in Tables 4–7, separately. The maneuvering time of OSV selected from the integer from the duration of on-orbit service mission is $[0, 23 \times 3600]$ s. The maneuvering time $I$ obtained by these four algorithms in different experiments are shown in Tables 4–7, and the best results can be obtained at these maneuvering times. It can be seen from Tables 4–7 that the average mass of the remaining fuel obtained by DPSO-TS algorithm, DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm is 538.5 kg, 467.8 kg, 488.9 kg, and 508.8 kg; the average mass of the remaining fuel obtained by DPSO-TS algorithm is better than that of the other algorithms. Except the ninth experiment of DPSO-TS algorithm, the mass of the remaining fuel obtained by DPSO-TS algorithm in the other experiments is superior to that obtained by the other algorithms in any experiment; the average computational time obtained by DPSO-TS algorithm, DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm is 1792.3 s, 1623.2 s, 1769.1 s, and 1797.3 s. The average computational time obtained by DPSO-TS algorithm differs from the shortest average computational time by 169.1 s, which has no effect on the service mission because there is a long preparation time before the mission is executed. The computational time of these four algorithms can be considered basically the same.

| Table 2: Parking orbit elements of satellites. |
| Satellite | $A$ (km) | $e$ | $I$ (°) | $\alpha$ (°) | $\Omega$ (°) | $f$ (°) |
|----------|---------|-----|--------|------------|----------|-------|
| S_1      | 9306    | 0.238550 | 28.321 | 224.138    | 307.927  | 135.862 |
| S_2      | 7059    | 0.000225 | 98.084 | 76.815     | 185.081  | 283.332 |
| S_3      | 9237    | 0.000165 | 52.006 | 171.386    | 350.005  | 188.701 |
| S_4      | 7828    | 0.001219 | 101.618| 87.562     | 293.441  | 282.681 |
| S_5      | 7707    | 0.054088 | 99.031 | 56.562     | 347.938  | 308.882 |
| S_6      | 7554    | 0.002882 | 90.045 | 245.844    | 262.555  | 29.901  |
| S_7      | 8428    | 0.201645 | 82.982 | 262.555    | 29.901   | 44.341  |
| S_8      | 9501    | 0.213200 | 28.226 | 34.607     | 152.421  | 325.392 |
| S_9      | 9401    | 0.016556 | 64.822 | 302.192    | 337.387  | 56.304  |
| S_{10}   | 9939    | 0.321639 | 56.917 | 231.629    | 218.498  | 128.371 |
| S_{11}   | 10687   | 0.355917 | 57.022 | 347.224    | 317.457  | 122.776 |
| S_{12}   | 6903    | 0.000549 | 96.390 | 110.033    | 325.324  | 203.070 |

| Table 3: Parking orbit elements of OSVs. |
| OSV     | $A$ (km) | $e$ | $I$ (°) | $\alpha$ (°) | $\Omega$ (°) | $f$ (°) |
|---------|---------|-----|--------|------------|----------|-------|
| O_1     | 7751    | 0.000120 | 32.223 | 56.232     | 76.223   | 255.520 |
| O_2     | 8400    | 0.000225 | 78.002 | 102.332    | 253.010  | 122.243 |
| O_3     | 8526    | 0.004120 | 76.924 | 154.257    | 73.020   | 243.000 |
| O_4     | 9201    | 0.370056 | 55.223 | 180.922    | 223.001  | 95.620  |
| O_5     | 7002    | 0.004021 | 87.235 | 102.556    | 330.765  | 88.598  |
| O_6     | 8002    | 0.529030 | 145.667| 156.007    | 200.321  | 270.321 |
| O_7     | 9923    | 0.052301 | 120.023| 254.124    | 122.232  | 55.220  |
| O_8     | 7125    | 0.003924 | 42.084 | 69.604     | 155.872  | 150.009 |
| O_9     | 8838    | 0.000249 | 98.178 | 23.270     | 200.020  | 60.000  |
| O_{10}  | 9123    | 0.013793 | 52.674 | 109.398    | 23.270   | 60.000  |
| O_{11}  | 7685    | 0.004693 | 37.477 | 52.707     | 158.017  | 60.000  |
| O_{12}  | 9215    | 0.489300 | 55.371 | 128.871    | 305.501  | 80.000  |
| O_{13}  | 7453    | 0.004218 | 23.618 | 77.562     | 293.441  | 50.101  |
| O_{14}  | 9675    | 0.054089 | 99.031 | 56.562     | 187.938  | 160.032 |
| O_{15}  | 6874    | 0.005555 | 97.420 | 104.012    | 316.284  | 204.781 |
| O_{16}  | 7534    | 0.003002 | 91.145 | 239.735    | 84.325   | 113.000 |
shown in Figures 3–7 in turn. Figures 8–12 show the
5, 3, 10, 1, and 8 experiments of CHIDPSO algorithm are
algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm
value curves obtained by DPSO-TS algorithm, DPSO al-
5501 6009 498 200 2215 33 781 7 299 5405 212 260
5522 4163 65 168 2216 32 781 2 299 5233 23 142
5522 4163 65 168 2216 32 781 2 299 5233 23 142
5524 6202 12 168 2109 32 789 0 297 5233 22 144

Table 4: Results of allocation optimization based on DPSO-TS algorithm.

| No. | X   | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 | S_8 | S_9 | S_{10} | S_{11} | S_{12} | t_s (s) | f (X) |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|---------|------|
| 1   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1801.4 | 540.3  |
| 2   | I   | 2301 | 6415 | 67 | 167 | 2251 | 449 | 781 | 1 | 299 | 5233 | 649 | 3339 | 1788.9 | 537.4 |
| 3   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1801.4 | 541.7  |
| 4   | I   | 4609 | 6422 | 1776 | 167 | 2254 | 2304 | 222 | 0 | 298 | 5233 | 1233 | 3339 | 1789.5 | 540.0 |
| 5   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1791.7 | 539.3  |
| 6   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1789.5 | 540.0  |

Table 5: Results of allocation optimization based on DPSO algorithm.

| No. | X   | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 | S_8 | S_9 | S_{10} | S_{11} | S_{12} | t_s (s) | f (X) |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|---------|------|
| 1   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1622.0 | 474.2  |
| 2   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1607.5 | 463.7  |
| 3   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1600.0 | 459.8  |
| 4   | OSV | O_5 | O_8 | O_{10} | O_{16} | O_{13} | O_3 | O_{14} | O_9 | O_7 | O_{12} | O_{15} | 1595.3 | 459.3  |
| 5   | I   | 5523 | 6201 | 12 | 168 | 2109 | 32 | 789 | 0 | 297 | 5233 | 22 | 144 | 1803.6 | 538.2 |

Figures 3–7 show the best, average, and worst fitness value curves obtained by DPSO-TS algorithm, DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm in the experiments with top 5 optimization results in turn. Figures 3–7 are used to show the changes of results obtained by these four algorithms. No. 6, 3, 1, 5, and 7 experiments of DPSO-TS algorithm are shown in Figures 3–7 in turn. No. 4, 1, 3, 5, and 7 experiments of DPSO algorithm are shown in Figures 3–7 in turn. No. 2, 7, 5, 9, and 8 experiments of DPSO-PDM algorithm are shown in Figures 3–7 in turn. No. 5, 3, 10, 1, and 8 experiments of CHIDPSO algorithm are shown in Figures 3–7 in turn. Figures 8–12 show the comparison of best fitness value curves obtained by DPSO-TS algorithm, DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm in the experiments with top 5 optimization results in turn, from which we can see that the DPSO-TS algorithm’s fitness value of each iteration is superior to the other algorithms after the eleventh iteration. Figures 8–12 show that the TS algorithm has strong local search capacity.

The simulation results demonstrate that the DPSO-TS algorithm has a higher accuracy compared to the DPSO, the DPSO-PDM, and the DPSO-CSA algorithms and can effectively solve the OSV allocation optimization problem.
### Table 6: Results of allocation optimization based on DPSO-PDM algorithm.

| No. | x   | S₁ | S₂ | S₃ | S₄ | S₅ | S₆ | S₇ | S₈ | S₉ | S₁₀ | S₁₁ | S₁₂ | tᵢ (s) | f(X)        |
|-----|-----|----|----|----|----|----|----|----|----|----|-----|-----|-----|--------|-------------|
| 1   | OSV | O₄ | O₁₁| O₁₀| O₁₃| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1750.5 | 475.6      |
| 2   | OSV | O₇ | O₆ | O₁₀| O₆ | O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1789.0 | 502.4      |
| 3   | I   | 2520| 412| 79 | 168| 2255| 321| 780| 650| 498| 5437| 6663| 4004 | 1752.9 | 481.0      |
| 4   | OSV | O₄ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1752.4 | 480.2      |
| 5   | I   | 2522| 160| 494| 211| 1238| 4300| 6750| 334| 498| 5001| 6678| 123  | 1749.5 | 499.1      |
| 6   | OSV | O₄ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1735.6 | 490.3      |
| 7   | OSV | O₄ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1785.9 | 500.4      |
| 8   | I   | 3546| 2455| 167| 163| 6721| 188| 428| 0  | 378| 5013| 343 | 667  | 1812.4 | 492.3      |
| 9   | OSV | O₄ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1789.3 | 495.6      |
| 10  | OSV | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1773.4 | 472.4      |

### Table 7: Results of allocation optimization based on CHIDPSO algorithm.

| No. | x   | S₁ | S₂ | S₃ | S₄ | S₅ | S₆ | S₇ | S₈ | S₉ | S₁₀ | S₁₁ | S₁₂ | tᵢ (s) | f(X)        |
|-----|-----|----|----|----|----|----|----|----|----|----|-----|-----|-----|--------|-------------|
| 1   | OSV | O₄ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1827.3 | 511.6      |
| 2   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1792.9 | 506.7      |
| 3   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1779.3 | 520.3      |
| 4   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1833.2 | 489.3      |
| 5   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1804.5 | 520.9      |
| 6   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1790.3 | 499.6      |
| 7   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1802.0 | 510.6      |
| 8   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1739.8 | 511.5      |
| 9   | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1788.4 | 499.6      |
| 10  | OSV | O₆ | O₁₀| O₁₁| O₁₆| O₁₅| O₃ | O₁₂| O₉ | O₁₄| O₂  | O₈  | 1815.2 | 518.3      |

![Figure 3: Continued](a) ![Figure 3: Continued](b)
Figure 3: Best, average, and worst fitness value curves in each of the four algorithms’ experiment with the first optimization results. (a) DPSO-TS algorithm. (b) DPSO algorithm. (c) DPSO-PDM algorithm. (d) CHIDPSO algorithm.

Figure 4: Best, average, and worst fitness value curves in each of the four algorithms’ experiment with the second optimization results. (a) DPSO-TS algorithm. (b) DPSO algorithm. (c) DPSO-PDM algorithm. (d) CHIDPSO algorithm.

Figure 5: Continued.
Figure 5: Best, average, and worst fitness value curves in each of the four algorithms' experiment with the third optimization results. (a) DPSO-TS algorithm. (b) DPSO algorithm. (c) DPSO-PDM algorithm. (d) CHIDPSO algorithm.

Figure 6: Best, average, and worst fitness value curves in each of the four algorithms' experiment with the fourth optimization results. (a) DPSO-TS algorithm. (b) DPSO algorithm. (c) DPSO-PDM algorithm. (d) CHIDPSO algorithm.

Figure 7: Continued.
Figure 7: Best, average, and worst fitness value curves in each of the four algorithms’ experiment with the fifth optimization results.

Figure 8: Comparison of the best fitness value curves in each of the four algorithms’ experiment with the first optimization results.

Figure 9: Comparison of the best fitness value curves in each of the four algorithms’ experiment with the second optimization results.

Figure 10: Comparison of the best value curves in each of the four algorithms’ experiment with the third optimization results.
5. Conclusions and Future Research

Fuel-efficient OSV allocation is a basic work to accomplish the on-orbit service mission in the background of deploying a large number of satellites in space in the future. We establish a transfer trajectory optimization model to obtain the mass of remaining fuel, which lays a foundation of allocation model. Then, we propose DPSO-TS algorithm to solve the allocation model, and the simulation results demonstrate that the proposed algorithm is more effective than DPSO algorithm, DPSO-PDM algorithm, and CHIDPSO algorithm. The DPSO-TS algorithm can be applied to other allocation optimization problems.

Although the proposed method has high accuracy, it still has some limitations. Further accelerating the computational time and investigating more algorithms with strong local search capacity in future research would be better.

Data Availability

The simulation data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, “On-orbit servicing missions: challenges and solutions for spacecraft operations,” in Proceedings of the AIAA Space Operations 2010 Conference, Space Operations: exploration, scientific utilization, and technology development, pp. 213–233, Huntsville, AL, USA, April 2010.
[2] Y. Li, J. B. Yao, C. J. Xin, and X. C. Su, “The approaching process analysis of on-orbit service vehicle to circular co-planarity orbit target,” Aerospace Control, vol. 33, no. 4, pp. 56–61, 2015.
[3] J. A. Andrew, B. S. David, and J. H. Terry, “Early mission design of transfers to halo orbits via particle swarm optimization,” The Journal of the Astronautical Sciences, vol. 63, no. 2, pp. 81–102, 2016.
[4] M. C. F. Bazzocchi and M. R. Emami, “Stochastic optimization of asteroid three-dimensional trajectory transfer,” Acta Astronautica, vol. 152, pp. 705–718, 2018.
[5] M. Ceriotti and J. P. Sanchez, “Control of asteroid retrieval trajectories to libration point orbits,” Acta Astronautica, vol. 126, pp. 342–353, 2016.
[6] M. J. Grant and R. D. Braun, “Rapid indirect trajectory optimization for conceptual design of hypersonic missions,” Journal of Spacecraft and Rockets, vol. 52, no. 1, pp. 177–182, 2015.
[7] C. L. Ranieri and C. A. Ocampo, “Optimization of roundtrip, time-constrained, finite burn trajectories via an indirect method,” Journal of Guidance, Control, and Dynamics, vol. 28, no. 2, pp. 306–314, 2005.
[8] B. Andrea and C. Christian, “A hybrid, self-adjusting search algorithm for optimal space trajectory design,” Advances in Space Research, vol. 50, no. 4, pp. 471–488, 2012.
[9] M. Rasotto, R. Armellin, and P. Di Lizia, “Multi-step optimization strategy for fuel-optimal orbital transfer of low-thrust spacecraft,” Engineering Optimization, vol. 48, no. 3, pp. 519–542, 2016.
[10] U. Boryczka and L. Strąk, “Heterogeneous DPSO algorithm for DTSP,” in Computational Collective Intelligence, vol. 9330, pp. 119–128, Springer, Berlin, Germany, 2019.
[11] X. L. Ding, X. G. Yuan, J. N. Zhang, Y. Zhang, and G., “Adaptive polarization mode dispersion compensation with DPSO algorithm,” Applied Mechanics and Materials, vol. 687–691, pp. 4015–4018, 2016.
[12] S. Sahoo, L. P. Mishra, and M. N. Mohanty, “Optimization of Z-shape microstrip antenna with I- slot using discrete particle swarm optimization (DPSO) algorithm,” Procedia Computer Science, vol. 92, pp. 91–98, 2016.
[13] X. J. Li, X. L. Wu, S. Xu, S. Qing, and P. C. Chang, “A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation,” Applied Soft Computing Journal, vol. 81, pp. 1–13, 2019.
[14] J. Xin, C. Yan, and X. Han, “Improved DPSO algorithm with dynamically changing inertia weight,” Advances in Swarm and Computational Intelligence, vol. 9140, pp. 115–123, 2015.
[15] A. A. Nagra, F. Han, and Q. H. Ling, “An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search,” Engineering Optimization, vol. 51, no. 7, pp. 1115–1132, 2019.
[16] Y. Wang, D. Y. Zhou, T. Yao, and H. Bai, “SA-DPSO-based weapon-target assignment for cooperative air defense with anti-aircraft gun-missile weapons,” *Materials Science and Engineering*, vol. 569, pp. 14–32, 2019.

[17] T. Vairam, S. Sarathambekee, and K. Umamaheshwari, “Multiprocessor task scheduling problem using hybrid discrete particle swarm optimization,” *Sadhana*, vol. 43, no. 12, pp. 206–219, 2018.

[18] B. Vasudevan and A. K. Sinha, “Reliability improvement of reconfigurable distribution system using GA and PSO,” *Electrical Engineering*, vol. 100, no. 2, pp. 1263–1275, 2018.

[19] H. Shayeghi, M. Mahdavi, and A. Bagheri, “An improved DPSO with mutation based on similarity algorithm for optimization of transmission lines loading,” *Energy Conversion and Management*, vol. 51, no. 12, pp. 2715–2723, 2010.

[20] C. Qi, “Application of improved discrete particle swarm optimization in logistics distribution routing problem,” *Procedia Engineering*, vol. 15, pp. 3673–3677, 2011.

[21] Y. Yan and X. Pingping, “A novel design of sharp MDFT filter banks with low complexity based on DPSO-MFO algorithm,” *Recent Developments in Intelligent Systems and Interactive Applications*, vol. 541, pp. 424–430, 2017.

[22] M. Bereta, “Regularization of boosted decision stumps using tabu search,” *Applied Soft Computing*, vol. 79, pp. 424–438, 2019.

[23] L. M. Hvattum, “On the value of aspiration criteria in tabu search,” *International Journal of Applied Metaheuristic Computing*, vol. 7, no. 4, pp. 39–49, 2016.

[24] G. Oualid, N. Placide, T. Raca, H. Saïd, and G. Fred, “Probabilistic tabu search for the cross-docking assignment problem,” *European Journal of Operational Research*, vol. 277, no. 3, pp. 875–885, 2019.

[25] R. Thomas, S. Andreas, S. Christian, P. Martin, and S. Georg, “Iterative trajectory learning for highly accurate optical satellite tracking systems,” *Acta Astronautica*, vol. 164, pp. 121–129, 2019.

[26] A. Erdal, B. Dominique, and S. Kai, “NMPC using Pontryagin’s minimum principle—application to a two-phase semi-batch hydroformylation reactor under uncertainty,” *Computers & Chemical Engineering*, vol. 108, pp. 47–56, 2018.

[27] O. Nadir, A. Lounis, and A. Rustem, “From offline to adaptive online energy management strategy of hybrid vehicle using Pontryagin’s minimum principle,” *International Jurnal of Automotive Technology*, vol. 19, no. 3, pp. 571–584, 2018.

[28] H. Kita, “Genetic algorithms for noisy fitness functions—applications, requirements and algorithms,” *Proceedings of the ISCIIE International Symposium on Stochastic Systems Theory and its Applications*, vol. 2001, pp. 137–142, 2001.

[29] X. Jiang and S. Li, “Mars atmospheric entry trajectory optimization via particle swarm optimization and Gauss pseudospectral method,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 12, pp. 2320–2329, 2016.