

## Research Article

# Asteroid Rendezvous Mission Design Using Multiobjective Particle Swarm Optimization

Ya-zhong Luo<sup>1</sup> and Li-ni Zhou<sup>2</sup>

<sup>1</sup> College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China

<sup>2</sup> Center for National Security and Strategy Studies, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Ya-zhong Luo; [luoyz@nudt.edu.cn](mailto:luoyz@nudt.edu.cn)

Received 18 October 2013; Revised 26 January 2014; Accepted 13 February 2014; Published 31 March 2014

Academic Editor: Kui Fu Chen

Copyright © 2014 Y.-z. Luo and L.-n. Zhou. 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.

A new preliminary trajectory design method for asteroid rendezvous mission using multiobjective optimization techniques is proposed. This method can overcome the disadvantages of the widely employed Pork-Chop method. The multiobjective integrated launch window and multi-impulse transfer trajectory design model is formulated, which employs minimum-fuel cost and minimum-time transfer as two objective functions. The multiobjective particle swarm optimization (MOPSO) is employed to locate the Pareto solution. The optimization results of two different asteroid mission designs show that the proposed approach can effectively and efficiently demonstrate the relations among the mission characteristic parameters such as launch time, transfer time, propellant cost, and number of maneuvers, which will provide very useful reference for practical asteroid mission design. Compared with the PCP method, the proposed approach is demonstrated to be able to provide much more easily used results, obtain better propellant-optimal solutions, and have much better efficiency. The MOPSO shows a very competitive performance with respect to the NSGA-II and the SPEA-II; besides a proposed boundary constraint optimization strategy is testified to be able to improve its performance.

## 1. Introduction

The optimization of interplanetary trajectories to an asteroid continues to arouse a great deal of interest [1–4]. Although the low-thrust propulsion is employed in asteroid rendezvous missions, an impulsive trajectory is always assumed for preliminary mission design and optimization, in which the access feasibility is evaluated, the launch window is determined, the gravity-assist maneuvers scheme is planned, and so forth. In a preliminary asteroid rendezvous trajectory design, the Pork-Chop method is widely used. This method employs the two-impulse algorithms including the optimal two-impulse noncoplanar transfer [1, 2], the classical Hohmann transfer [3], and the two-impulse Lambert algorithm [4] to design the transfer trajectory. The total  $\Delta v$  corresponding to different departure time and arrival time is then obtained and the contours of minimum total  $\Delta v$  are plotted to assist the designer to find the best launch window and transfer trajectory.

This type of method is very intuitionistic and easily executed. However, only the two-impulse trajectory is investigated in this method, and as demonstrated by Lawden's theory [5, 6], the two-impulse trajectory is not the propellant-optimal solution under most conditions. Besides, this method is in essence of an exhaustive searching method, its computation cost increases exponentially as the search space increases, and much human intervention is required as no orderliness exists in most of the contours.

In this paper, a new preliminary trajectory design method for asteroid rendezvous mission using multiobjective optimization techniques is proposed. This method can overcome the disadvantages of the Pork-Chop method. The multiobjective integrated launch window and multi-impulse transfer trajectory design model is formulated, which employs minimum-fuel cost and minimum-time transfer as two objective functions. In this model, the Earth departure date, hyperbolic velocity, and the interplanetary transfer impulses

are all chosen optimization design variables. The multiobjective particle swarm algorithm is employed to locate the Pareto solution, by which the relationships characteristics among the overall mission parameters can be effectively revealed.

The particle swarm optimization (PSO) algorithm firstly introduced by Kennedy and Eberhart [7] in 1995 has been successfully applied in many fields of research. However, it is only recently applied to aerospace trajectories optimization [8–11]. The single-objective PSO algorithm has been testified as one successful spacecraft trajectory optimizer in designing impulsive interplanetary trajectories [8], low-thrust trajectories for asteroid exploration [9], Lyapunov and Halo (periodic) orbits [10], and multiple-burn rendezvous trajectories [11].

The implementation of the PSO algorithm adopts a population of particles, whose behavior is affected by either the best local (i.e., within a certain neighborhood) or the best global individual. The relative simplicity of PSO and the fact that it is a population-based technique have made it a natural candidate to be extended for multiobjective optimization. In a survey paper in 2006 on multiobjective particle swarm optimization (MOPSO) [12], it was reported that there were currently over twenty-five different proposals of MOPSO reported in the specialized literature. The studies on MOPSO remain a very active area of research, and the MOPSO has been successfully applied to many practical multiobjective optimization problems, recently applied to robotics [13], industrial management [14], and chemical engineering [15]. It also has been applied in the domain of aerospace including airfoil shape optimization, complex physics/shape optimization, and multidisciplinary design optimization [16]. However, it was seldom applied in solving multiobjective spacecraft trajectory optimization problems. The recent studies show that the single-objective PSO algorithm is one effective, reliable, and accurate spacecraft trajectory optimizer [8–11]. In this study, we will show that the MOPSO could be one successful optimizer for multiobjective spacecraft trajectory optimization problems.

In summary, the main contribution of this paper is two-fold. (1) A novel asteroid rendezvous mission design method using the multiobjective techniques is proposed. Compared with the current widely employed Pork-Chop method, the proposed approach is demonstrated to be able to provide much more easily used results, obtain better propellant-optimal solutions, and have much better efficiency. (2) As far as we know, it is the first time to apply the MOPSO to spacecraft trajectory optimization. The MOPSO proves to be quite effective in finding the Pareto-optimal solutions to asteroid rendezvous multiobjective optimization problems. The MOPSO shows a very competitive performance with respect to two highly competitive multiobjective evolutionary algorithms: the nondominated sorting genetic algorithm-II (NSGA-II) [17] and the strength Pareto evolutionary algorithm-II (SPEA-II) [18].

## 2. Optimization Problem

*2.1. Asteroid Rendezvous Design Problem.* The interplanetary transfer trajectory is always divided into three different

segments, that is, planet departure segment, heliocentric transfer segment, and capture segment by using the concept of influencing sphere. As the influencing sphere of the planet is much smaller than that of the sun, the flight path and flight time of the departure and capture segments are much small compared with that of the heliocentric transfer. Therefore, for a preliminary trajectory design of an asteroid exploration mission, the design emphasis is firstly focused on the heliocentric transfer, while the planet departure is assumed as instantaneous process with an impulsive maneuver and the capture segment is omitted.

This paper studies the asteroid rendezvous problem departure from the Earth.  $t_0$  represents the spacecraft launch date which is to be designed, and  $V_\infty$ ,  $u$ , and  $v$  define the departure hyperbolic velocity according to the formulas

$$\theta = 2\pi u,$$

$$\varphi = \arccos(2v - 1) - \frac{\pi}{2}, \quad (1)$$

$$\frac{\mathbf{v}_\infty}{V_\infty} = \cos(\varphi) \cos(\theta) \mathbf{i} + \cos(\varphi) \sin(\theta) \mathbf{j} + \sin(\varphi) \mathbf{k}.$$

Let  $\mathbf{r}_E(t_0)$ ,  $\mathbf{v}_E(t_0)$  be the heliocentric position and velocity of the Earth at  $t_0$ ; then the initial position and velocity vector of the spacecraft,  $\mathbf{r}(t_0)$  and  $\mathbf{v}(t_0)$ , entering the heliocentric transfer trajectory are defined as

$$\mathbf{r}(t_0) = \mathbf{r}_E(t_0), \quad \mathbf{v}(t_0) = \mathbf{v}_E(t_0) + \mathbf{v}_\infty. \quad (2)$$

The heliocentric transfer trajectory is modeled by a two-body dynamic model with the following governing equations:

$$\begin{aligned} \dot{x} &= v_x, \\ \dot{y} &= v_y, \\ \dot{z} &= v_z, \\ \dot{v}_x &= -\frac{\mu x}{(x^2 + y^2 + z^2)^{3/2}} + a_x, \\ \dot{v}_y &= -\frac{\mu y}{(x^2 + y^2 + z^2)^{3/2}} + a_y, \\ \dot{v}_z &= -\frac{\mu z}{(x^2 + y^2 + z^2)^{3/2}} + a_z, \end{aligned} \quad (3)$$

where  $x$ ,  $y$ , and  $z$  are position components of spacecraft,  $v_x$ ,  $v_y$ , and  $v_z$  are velocity components of spacecraft,  $a_x$ ,  $a_y$ , and  $a_z$  are acceleration component of spacecraft, and  $\mu$  is the gravity parameter of the sun.

The thrust acceleration  $\Gamma(t) = (a_x, a_y, a_z)^T$  can be approximated as  $n$  impulses

$$\Gamma(t) = \sum_{i=1}^n \Delta \mathbf{v}_i \delta(t - t_i), \quad (4)$$

where  $t_i$  is the time where an impulse is applied,  $\Delta \mathbf{v}_i = (\Delta v_{xi}, \Delta v_{yi}, \Delta v_{zi})^T$  ( $i = 1, 2, \dots, n$ ), and they are all the variables to be designed. The final state conditions are defined

$$\mathbf{r}(t_f) = \mathbf{r}_A(t_f), \quad \mathbf{v}(t_f) = \mathbf{v}_A(t_f), \quad (5)$$

where  $t_f$  is the rendezvous time of spacecraft with the target asteroid and  $\mathbf{r}_A(t_f)$  and  $\mathbf{v}_A(t_f)$  are the position and velocity vector of the target asteroid at the rendezvous time.

**2.2. Feasible-Solution Iteration Model.** In order to avoid dealing with the equality constraints described as (5), the Lambert algorithm is employed to establish the infeasible iteration optimization model. The chosen independent variables, that is, the optimization variables, are impulse times and the first  $n - 2$  impulses

$$t_i \quad (i = 1, 2, \dots, n), \quad \Delta \mathbf{v}_i \quad (i = 1, 2, \dots, n - 2). \quad (6)$$

The last two impulses are determined by solving the Lambert problem constrained by (5). In this feasible iteration model, each evaluation of the objective function produces a feasible solution that satisfies implicitly the rendezvous conditions. Detail on this multi-impulse rendezvous optimization model using the Lambert algorithm can be found in [19, 20].

**2.3. Multiobjective Optimization Model.** The total velocity characteristic is chosen as the first objective function

$$\min f_1(\mathbf{x}) = \Delta v = V_\infty + \sum_{i=1}^n |\Delta \mathbf{v}_i|. \quad (7)$$

The heliocentric transfer time is chosen as the second objective function

$$\min f_2(\mathbf{x}) = t_f - t_0. \quad (8)$$

The constraint on the time of impulse is considered. The general constraint on  $t_i$  ( $i = 1, 2, \dots, n$ ) is

$$t_0 \leq t_1 < t_2 \cdots < t_n \leq t_f. \quad (9)$$

As the transfer time is one of the objective functions,  $t_f$  is chosen as an optimization variable. The total optimization variables include four parts:  $t_0$ ,  $V_\infty$ ,  $u$ ,  $v$ ,  $t_i$  ( $i = 1, 2, \dots, n$ ),  $\Delta \mathbf{v}_i$  ( $i = 1, 2, \dots, n - 2$ ), and  $t_f$ . In order to improve optimization performance, the variable-scaling method is imposed on  $t_i$ . Let  $\alpha_i = (t_i - t_0)/(t_f - t_0)$ ,  $\alpha_i \leq 1$ . Therefore, the final optimization variables  $\mathbf{x}$  are

$$\mathbf{x} = (t_0, V_\infty, u, v, \alpha_1, \dots, \alpha_n, \Delta \mathbf{v}_1, \dots, \Delta \mathbf{v}_{n-2}, t_f)^T. \quad (10)$$

### 3. Optimization Algorithms

**3.1. Introduction of MOEA.** A general multiobjective optimization problem is to find the design variables that optimize a vector objective function over the feasible design space. The objective functions are the quantities that the designer wishes to minimize, maximize, or attain a certain value. The problem formulation in standard form for a minimization is given here, which is similar for the other cases:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \quad (11)$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, p, \quad (12)$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q,$$

where

$$\mathbf{x} = (x_1, x_2, \dots, x_v)^T \in \mathbf{X} \subset \mathbf{R}^v. \quad (13)$$

For the multiobjective asteroid rendezvous design problem, the two objective functions are described by (7) and (8), the constraints are described by (9), and the optimization variables are described as (10).

The classical optimization method for a multiobjective optimization problem is the weighting method. In recent years, the multiobjective evolutionary algorithms (MOEA) have been greatly investigated in the domain of multiobjective optimization. There are many variants of MOEA reported in the literature; a recent survey on MOEA and their application in aeronautical and aerospace engineering has been made in [16].

In the study, except for the MOPSO, we also test two other mostly popular MOEA. The first is the NSGA-II algorithm which is proposed by Deb et al. [17]. This algorithm uses the idea of transforming the  $m$  objectives to a single fitness measure by the creation of a number of fronts, sorted according to nondomination. During the fitness assignment, the first front is created as the set of solutions that is not dominated by any solutions in the population. These solutions are given the highest fitness and temporarily removed from the population, then a second nondominated front consisting of the solutions that are now nondominated is built and assigned the second-highest fitness, and so forth. This is repeated until each of the solutions has been assigned a fitness. After each front has been created, its members are assigned crowding distances (normalized distance to closest neighbors in the front in the objective space) later to be used for niching. The NSGA-II has been successfully applied in spacecraft trajectory optimization, for example, in designing a three-objective impulse rendezvous problem [20, 21] and a two-objective robust rendezvous problem with considering uncertainty [22].

The second is the SPEA-II proposed by Zitzler et al. [18]. It uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set, removing the dominated solutions. For each individual in this external set, a strength value is computed. Pareto dominance is adopted to ensure that the solutions are properly distributed along the Pareto front. It also uses a nearest neighbor density estimation technique and a fine-grained fitness assignment strategy which guide the search more efficiently.

**3.2. Brief Description of the MOPSO.** The MOPSO applied in this study is the algorithm proposed by Pulido and Coello [23], which was competitive against the most popular MOEA such as the NSGA-II, the PAES, and other MOPSO on typical benchmark problems, under common performance metrics [23]. The source code of the MOPSO is available from the EMOO repository located at <http://delta.cs.cinvestav.mx/~ccoello/EMOO>.

The MOPSO is based on the use of Pareto ranking and a subdivision of decision variable space into several subswarms

```

Begin
  For each swarm
    (1) Initialize its particles
    (2) Initialize the set of global leaders:  $g_{leader}$ 
  End For
  DO
    For each swarm
      Do
        For each particle
          (3) Select a leader
          (4) Perform the flight
          (5) Update the value
          If it is a leader then add to the  $g_{leader}$ 
        End For
        While (number of iterations  $\leq sgmax$ )
          (6) Store leaders in  $g_{leader}$  in  $n_{swarms}$  groups
        End For
        (7) Assign each leader group to a swarm
      While (number of iterations  $\leq GMax$ )
    End

```

ALGORITHM 1

which is done using clustering techniques. The complete execution process of this algorithm can be divided into three stages: initialization, flight, and generation of results [23].

At the first stage, every swarm is initialized. Each swarm creates and initializes its own particles and generates the leaders set among the particle swarm set by using Pareto ranking. In the second stage, it firstly performs the execution of the flight of every swarm; next, it applies a clustering algorithm to group the guide particles. This is performed until reaching a total of  $GMax$  iterations. The execution of the flight of each swarm can be seen as an entire PSO process (with the difference that it will only optimize a specific region of the search space). First, each particle will select a leader to which it will follow. At the same time, each particle will try to outperform its leader and to update its position. If the updated particle is not dominated by any member of the leaders set, then it will become a new leader. The execution of the swarm will start again until a total of  $sgmax$  iterations are reached. Once all the swarms have finished their flights, a clustering algorithm takes the control by grouping the closest particle guides into  $n_{swarms}$  swarms. Each resulting group will be assigned to a different swarm. The third and final stage will present all the nondominated solutions found.

Details of this algorithm can be found in [23] and its pseudocode code is shown in Algorithm 1.

The MOPSO algorithm requires the following parameters: (1)  $GMax$ : the total number of generations that the algorithm will be executed; (2)  $sgmax$ : the number of internal generations that the particles of each swarm will run before sharing their leaders; (3)  $n_{particles}$ : the total number of particles; and (4)  $n_{swarms}$ : the number of particle groups.

**3.3. Constraints Optimization Method.** The multiobjective asteroid mission design problem is a highly constrained problem whose constraints are described in (9). The simulation experiments show that these constraints strongly influence the convergence. In the MOPSO, the constraints are always handled in checking Pareto dominance [23, 24]. When we compare two individuals, we first check their feasibility. If they are both feasible, then the comparison is done using Pareto dominance. If one is feasible and the other is infeasible, the feasible individual wins. If both are infeasible, then the individual with the lower amount of total constraint violation wins.

The total constraint is calculated by making use of a non-differentiable penalty function. For the general constrained problem in (10)–(13), the penalized, total constraint function  $C(\mathbf{x}, M)$  is

$$C(\mathbf{x}, M) = M \left[ \sum_{i=1}^p \max(0, g_i(\mathbf{x})) + \sum_{j=1}^q |h_j(\mathbf{x})| \right]. \quad (14)$$

In the present work  $M = 1000000$  is used as a penalty coefficient.

For most optimization problems, each design variable has its own upper and low values. Thus, a strategy to maintain the particles within the search space in case they go beyond their boundaries is necessary for the MOPSO algorithm.

In [23, 24], the following strategy was employed. When a decision variable goes beyond its boundaries, the decision variable takes the value of its corresponding boundary (either the lower or the upper boundary). The strategy will be likely effective when the Pareto solutions are located in the boundary of variables.

However, our simulation experiments show that this strategy is not very effective in solving our problems. Therefore, another simple strategy is proposed. When a decision variable goes beyond its boundaries, the decision variable takes a random value from its feasible design space. The probability-based disposal to boundary constraint could enrich the diversity of swarm flight.

The MOPSO with these two different boundary constraint optimization strategies is, respectively, called as MOPSO-I and MOPSO-II. Their performance is compared in Section 4.3.

**3.4. Algorithm Assessment Metrics.** In order to allow a quantitative assessment of the performance of the MSOPO, we adopted the following two metrics.

The first one is the epsilon indicator [25]. Given a reference Pareto front (ideally the true Pareto front, if available), the epsilon indicator measures the minimum amount  $\epsilon$  necessary to translate all the points of the found Pareto front to weakly dominate the reference set. The epsilon indicator has two types, that is, the additive type and the multiplicative type, and the multiplicative type is used here.

The second one is the hypervolume indicator [26]. The hypervolume of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively. Generally, hypervolume is favored because

TABLE 1: Orbit elements of asteroid.

Index	Asteroid 1	Asteroid 2
Name	1999YR14	2340
$a$ (AU)	1.65365126892224	0.84421076388332195
$e$	0.40069261757759106	0.44975834146342486
$i$ (deg)	3.7221930161441943	5.8547882390182853
$\Omega$ (deg)	3.1338963493744654	211.50460158030430
$\omega$ (deg)	9.4143875285008676	39.994195753797953
$M$ (deg)	114.73402134869427	240.44827444641544
Epoch (MJD2000)	3255.0000	3255.000000

it captures in a single scalar both the closeness of the solutions to the optimal set and, to some extent, the spread of the solutions across objective space.

The statistical test chosen for result evaluation is the Mann-Whitney test [27]. This is a nonparametric rank-based test that can be used to compare two independent sets of sampled data. It outputs  $P$  values that estimate the probability of a failure to reject the null hypothesis of the study question. Here the  $(1 - P)$  values can be interpreted as the probability that the performance of one algorithm is different (superior or junior) with statistical significance to that of the other.

## 4. Simulation Results

**4.1. Problem Configuration.** In order to testify the effectiveness of the proposed method, two different asteroid mission designs are illustrated. Table 1 lists the orbit elements of the two asteroids. The upper and lower space of the optimization variables are provided in Table 2. Three different cases with the number of impulses of 2, 3, and 4 are, respectively, tested for each asteroid mission.

**4.2. Pareto Fronts Analysis.** From our experiments, this multiobjective asteroid rendezvous design problem is very difficult to be solved, and the true Pareto fronts of this multiobjective problem are not known. In order to obtain the Pareto fronts as possible close to the true ones, the MOPSO is executed with a much larger number of function evaluations. The parameters of the MOPSO are  $n_{\text{particles}} = 400$ ,  $GMax = 400$ ,  $sgmax = 5$ , and  $n_{\text{swarms}} = 20$ .

Considering the stochastic characteristic of the MOPSO, 10 independent runs for each test case are completed. All the Pareto solutions of the 10 independent runs are compared, and the repeated and non-Pareto solutions are deleted, and the revised Pareto solutions are selected as the final solutions. In the following examples, the figured Pareto fronts are all obtained in the same method. The MOPSO with different boundary constraint optimization methods, that is, the MOPSO-I and MOPSO-II, is both tested. The performance comparisons between the MOPSO-I and MOPSO-II will be analyzed in the next section.

Figure 1 compares the Pareto solution front for two-impulse, three-impulse, and four-impulse of the Asteroid 1999YR14 mission. Figure 2 compares this for the Asteroid 2340 mission. The tradeoffs between the total  $\Delta v$  and

the transfer time are clearly demonstrated by Figures 1 and 2, which will be useful for a mission designer. Several inherent peculiarities regarding the optimal multiobjective asteroid rendezvous trajectories have been observed from the optimization results. For the second asteroid, the corresponding total  $\Delta v$  reduces evidently as the transfer time increases when the transfer time is in the range of [100, 140] day, but small change in other ranges; therefore the Pareto front concentrates on a narrow range of 40 days. For the first asteroid, the corresponding total  $\Delta v$  reduces evidently as the transfer time increases in a large range of [100, 900] day; therefore the Pareto solutions distribute in a much larger space, but the Pareto front is discontinuous in small range as demonstrated in Figure 1.

In order to explain this phenomenon, the propellant-optimal solutions are obtained by using the approach employed in [28]. In this optimization, the transfer time is fixed and  $\Delta v$  is calculated every 5 days in the transfer time of range [100, 1100] days. The relations between the transfer time and  $\Delta v$  of the propellant-optimal solutions are illustrated in Figure 3. Some transfer time ranges with much-higher propellant cost are located; for example, the  $\Delta v$  is about 9.5 km/s for the transfer time of 300 days, while being only 6.2 km/s for the transfer time of 280 days. The former solution is larger than the latter solution by about 50% in  $\Delta v$ ; besides its transfer time is also larger than the latter. This issue can explain why the Pareto fronts of two-impulse solutions discontinuous near the transfer time of 300 days.

The influence of different number of impulses on the total  $\Delta v$  has been also demonstrated by Figures 1 and 2. It is obvious that a three-impulse trajectory and a four-impulse trajectory cost less propellant in compared with a two-impulse trajectory. It is necessary to increase the number of impulse to find a much better trajectory in the preliminary asteroid rendezvous mission design.

### 4.3. Performance Analysis of the MOPSO

**4.3.1. Boundary Constraint Optimization Comparisons.** As seen from Figures 1 and 2, the boundary constraint optimization method of the MOPSO has a great influence on the Pareto fronts. Graphically, the MOPSO-II has much better performance in diversity, and the spread of its solutions found is much larger than that of the MOPSO-I. This can easily explain that probability-based disposal to boundary constraint can enrich the diversity of swarm flight.

It is not easily to determine which one is much closer to the true Pareto fronts from Figure 1, and the two-impulse comparison case is shown in Figure 4(a), while it is easy to see from Figure 2 that the Pareto fronts obtained by MOPSO-I are much closer to the true Pareto fronts, and the three-impulse comparison case is redrawn in Figure 4(b) to demonstrate this clearly.

Therefore, the quantitative metrics are calculated and the statistical results are provided in Table 3. As seen from Table 3, the MOPSO-II has better performance with respect to the hypervolume indicator (larger value means better) for the two cases. While, for the epsilon indicator (smaller value means

TABLE 2: Design space of variables.

Variables	Space	Units
$t_0$	[4000, 10000]	MJD2000
$V_\infty$	[0, 5]	km/s
$u$	[0, 1]	n/a
$v$	[0, 1]	n/a
$\alpha_i$ ( $i = 1, 2, \dots, n$ )	$n = 2$ : [0, 0.6], [0.3, 1] $n = 3$ : [0, 0.4], [0.1, 0.8], [0.5, 1] $n = 4$ : [0, 0.4], [0.1, 0.8], [0.5, 0.9], [0.7, 1]	n/a
$\Delta v_i^x, \Delta v_i^y, \Delta v_i^z$ ( $i = 1, 2, \dots, n - 2$ )	[-4, 4]	km/s
$t_f - t_0$	[100, 1500]	days

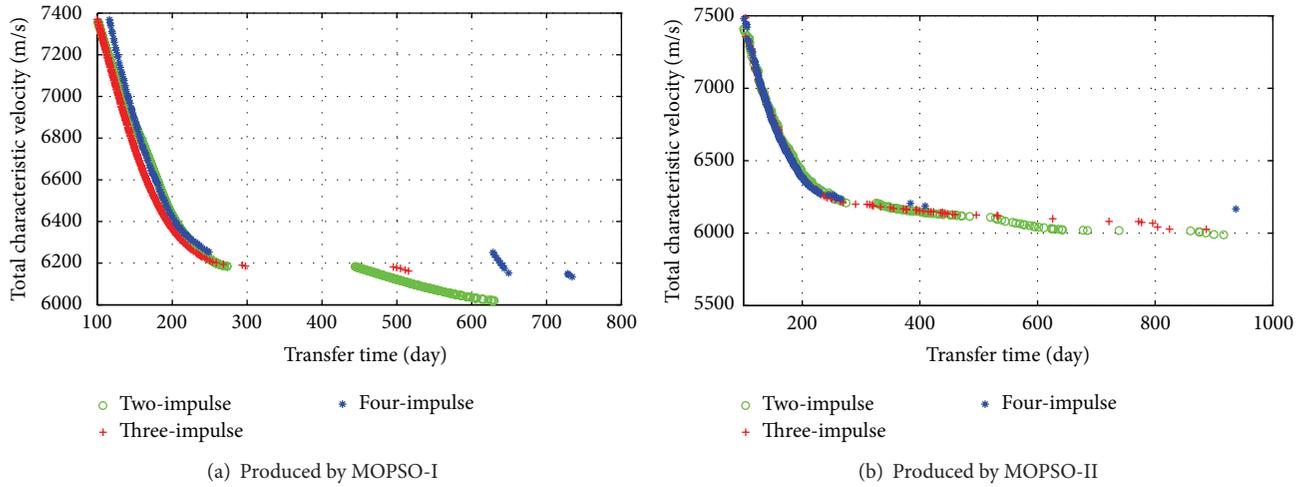


FIGURE 1: Pareto fronts of Asteroid 1.

better), the MOPSO-II is much better in case of Asteroid 1, two-impulse problem and the MOPSO-I are much better in case of Asteroid 2, three-impulse.

Through the comparisons provided in Table 2 and Figure 4, our proposed boundary constraint demonstrates much better performance in total. Besides, these comparisons show that the same algorithm will have different performance in solving the same type of problem with different configurations.

**4.3.2. Algorithm Parameters Analysis.** Our simulation experiments show that  $n_{\text{particles}}$  and  $GMax$  are the two main parameters affecting the performance of the MOPSO in solving this complex multiobjective problem. In order to quantitatively evaluate their influence,  $n_{\text{particles}}$  is, respectively, set as 400, 200, and 100,  $GMax$  is also, respectively, set as 400, 200, and 100, and a total of nine groups of parameters (in Table 4) are further tested. Other parameters are the same as used in Section 4.2.

Ten independent runs for the MOPSO-II with each group of parameters in solving the three-impulse, Asteroid 1 problem, are executed.

Figure 5(a) compares the Pareto fronts of Cases 1, 5, and 9, Figure 5(b) compares that of Cases 2, 5, and 8,

and Figure 5(c) compares that of Cases 7, 8, and 9. These comparisons are to show the influence of the total number of function evaluations. It is clearly seen from Figure 5 that the larger number of function evaluations evidently increases the performance of the MOPSO. Thus, a larger number of swarm size and iteration are necessary for obtaining the optimal Pareto fronts for this practical multiobjective optimization problem.

Figure 6(a) compares the Pareto fronts of Cases 2 and 4, Figure 6(b) compares that of Cases 6 and 8, and Figure 6(c) compares that of Cases 3, 5, and 7. These comparisons are to show the influence of  $n_{\text{particles}}$  and  $GMax$  with the same number of function evaluations. Evidently, we cannot determine which case is better by the graphical results provided in Figure 6. Therefore, the quantitative metrics are calculated and the statistical results are provided in Tables 5, 6, 7, and 8.

The comparisons between Case 2 and Case 4 show that larger size of swarm can obtain better average epsilon indicator with a  $P$  value of 67.76% but worse average hypervolume indicator with a  $P$  value of 14.05%. The comparisons between Case 6 and Case 8 show that larger size of swarm both increase average performance in epsilon and hypervolume, with a  $P$  value of 47.27% and 7.57%. By comparing Cases 3, 5, and 7, Case 5 demonstrates the best performance in total.

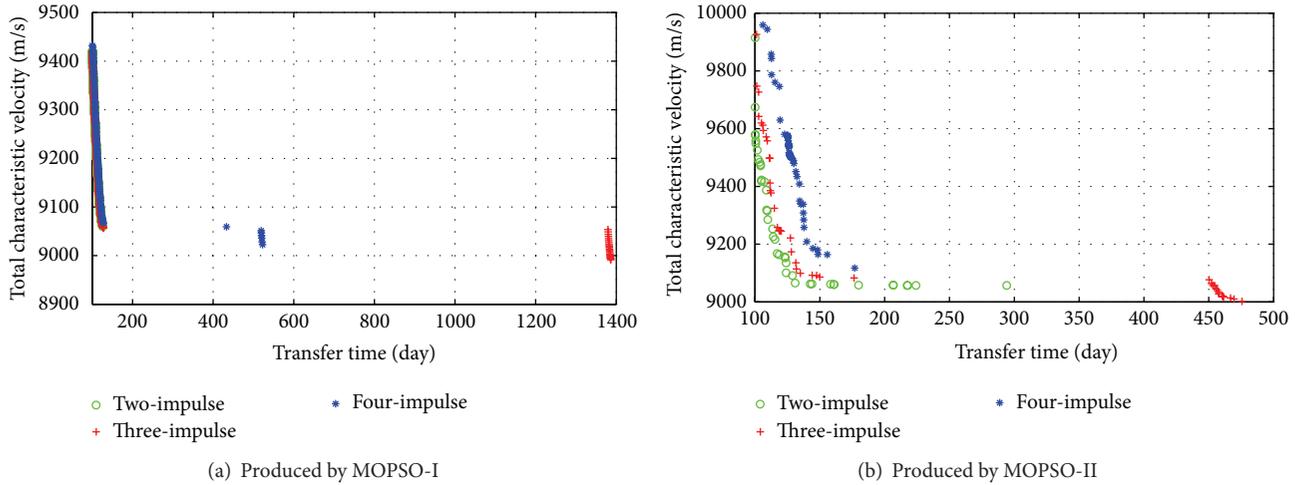


FIGURE 2: Pareto fronts of Asteroid 2.

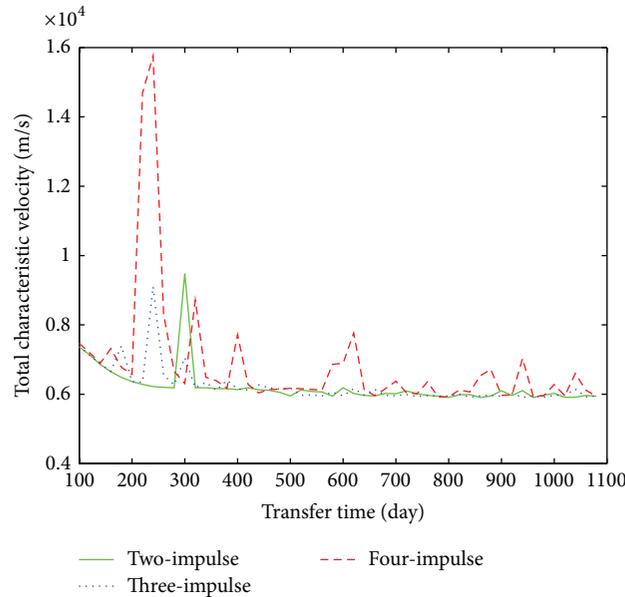


FIGURE 3: Relations between transfer time and total characteristic velocity (propellant-optimal solutions, Asteroid 1).

TABLE 3: MOPSO performance with different boundary constraint optimization.

Test problem	Algorithm	Epsilon			Hybervolume		
		Mean	std	<i>P</i> value	Mean	std	<i>P</i> value
Asteroid 1, two-impulse	MOPSO-I	0.011987	0.008378	0.1620	0.972618	0.006829	0.0257
	MOPSO-II	0.006833	0.004066		0.977958	0.001301	
Asteroid 2, three-impulse	MOPSO-I	0.025630	0.0278051	0.1620	0.87509	0.308357	0.6232
	MOPSO-II	0.0399274	0.026254		0.968537	0.016548	

TABLE 4: MOPSO parameters configurations.

Index	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
$n_{particles}$	400	400	400	200	200	200	100	100	100
<i>GMax</i>	400	200	100	400	200	100	400	200	100

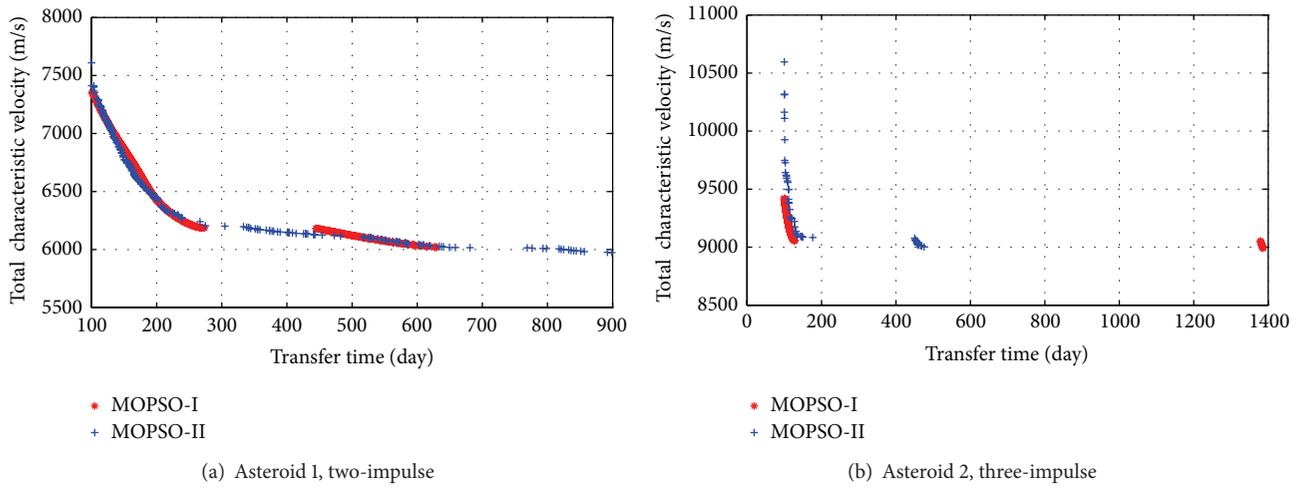


FIGURE 4: Pareto fronts produced by MOPSO with different boundary constraint optimization.

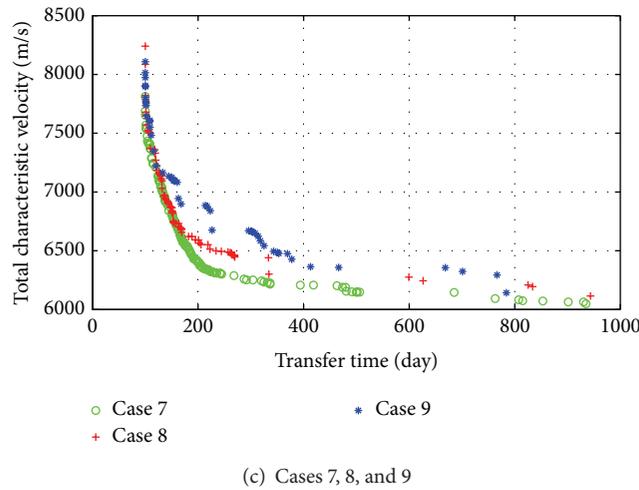
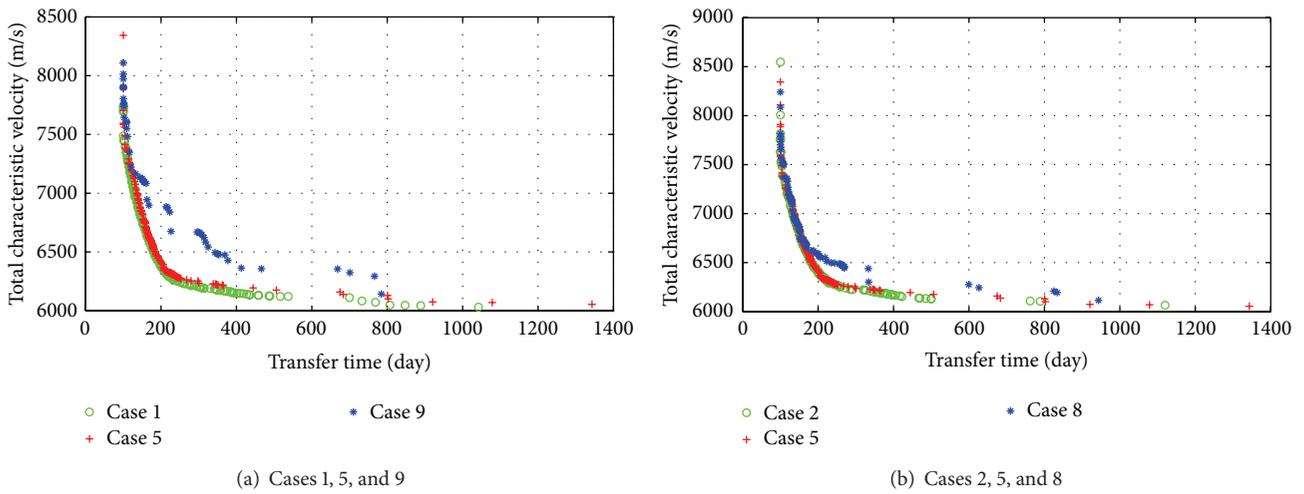


FIGURE 5: Pareto fronts produced by MOPSO-II with different number of function evaluations.

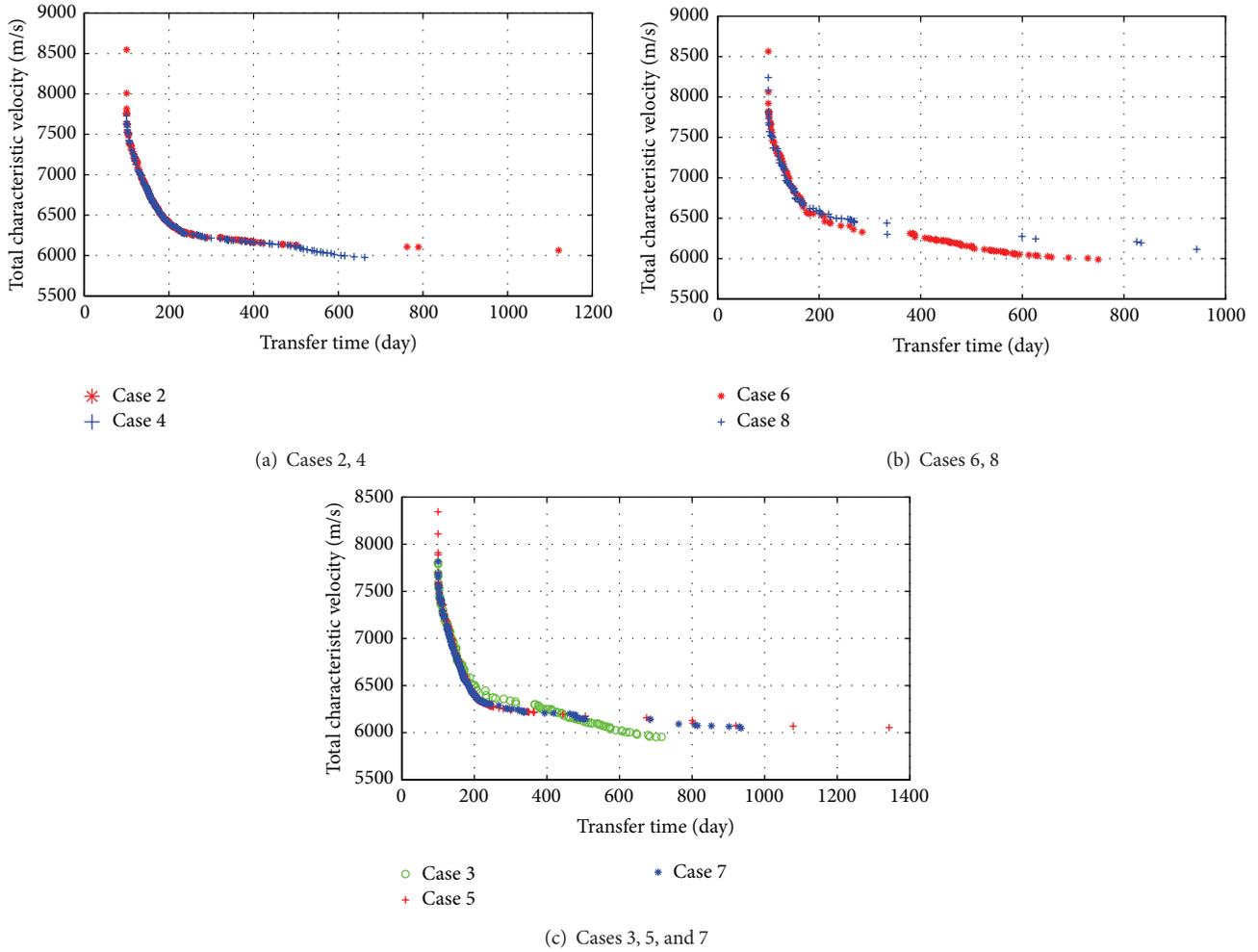


FIGURE 6: Pareto fronts produced by MOPSO-II with same number of function evaluations.

TABLE 5: MOPSO-II performance with different parameters (Cases 2 and 4).

Test problem	Algorithm	Epsilon			Hybervolume		
		Mean	std	<i>P</i> value	Mean	std	<i>P</i> value
Asteroid 1, three-impulse	Case 2	0.010692	0.006797	0.6776	0.984063	0.002738	0.1405
	Case 4	0.012918	0.012848		0.985643	0.002736	

TABLE 6: MOPSO-II performance with different parameters (Cases 6 and 8).

Test problem	Algorithm	Epsilon			Hybervolume		
		Mean	std	<i>P</i> value	Mean	std	<i>P</i> value
Asteroid 1, three-impulse	Case 6	0.006045	0.002645	0.4727	0.99544	0.000816	0.0757
	Case 8	0.007806	0.004042		0.994456	0.001074	

TABLE 7: MOPSO-II performance with different parameters (Cases 3, 5, and 7).

Test problem	Algorithm	Epsilon		Hybervolume	
		Mean	std	Mean	std
Asteroid 1, three-impulse	Case 3	0.00583413	0.0036399	0.994062	0.00135846
	Case 5	0.00438474	0.00129004	0.994755	0.00100777
	Case 7	0.00564449	0.00274183	0.994917	0.00119178

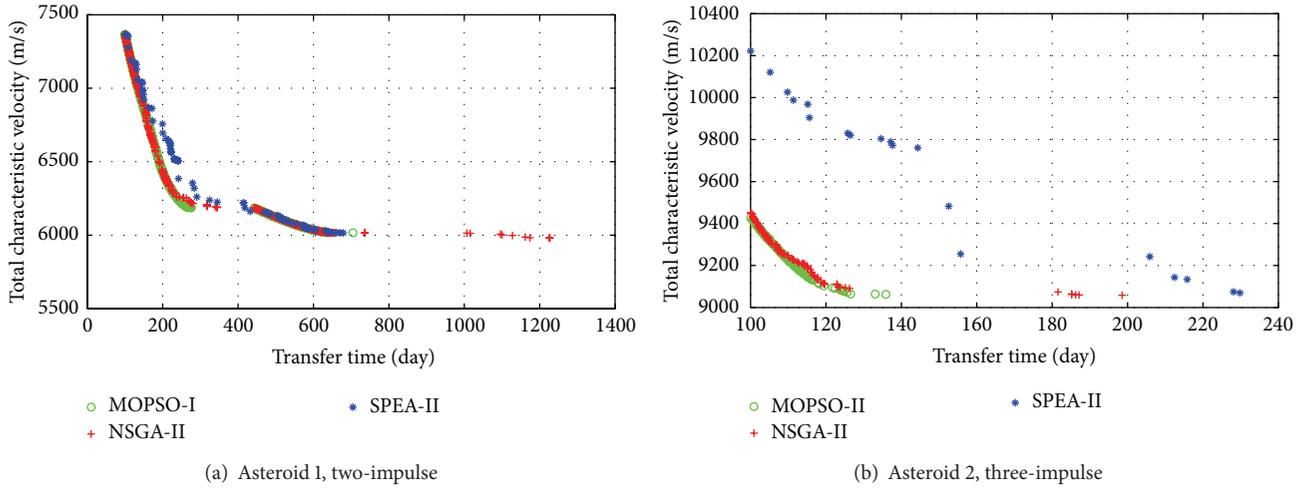


FIGURE 7: Pareto fronts produced by MOPSO, NSGA-II, and SPEA-II.

TABLE 8:  $P$  value of MOPSO (Cases 3, 5, and 7).

	Case 3	Case 5	Case 7
Case 3		<b>0.4727</b>	<b>0.9097</b>
Case 5	0.3447		<b>0.4727</b>
Case 7	0.2123	0.7337	

Bold indicates epsilon; the other is hypervolume.

However, the improvements are not evident as the  $P$  values are all a large value.

Although the result obtained from this experiment seems to be inconclusive on which parameter is the best, we can argue that the MOPSO's performance is not sensitive to the main PSO parameters, under the condition that the total number of function evaluation retains a large value. We have obtained competitive results in most cases without paying special attentions on PSO parameters.

**4.4. Comparison of the MOPSO with Other MOEA.** To demonstrate the performance of the employed MOPSO, two other popular algorithms are tested and compared: the NSGA-II and the SPEA-II.

In the following examples, the total number of function evaluations was set to 20000 for all the algorithms compared. The NSGA-II and the SPEA-II were run using a population size of 200, a maximum number of generations of 100. The MOPSO-I and MOPSO-II used 200 particles, a maximum number of generations of 20, a maximum number of generations per swarm of 5, and a total of 5 swarms. The source code of the NSGA-II, and SPEA-II provided in the EMOO repository is also employed in this study.

The two test problems are the same to those employed in Section 4.3 to compare boundary constraint optimization. Ten independent runs for each test case are executed. The final Pareto fronts are compared in Figure 7 and the statistical results with respect to epsilon and hypervolume indicators are provided in Tables 9, 10, and 11.

Evidently seen from Figure 7(a), the MOPSO-I produces better Pareto fronts than the NSGA-II and SPEA-II for the first test problem. The statistical results of the two indicators also support this. The  $P$  value with respect to epsilon is calculated as 0.58% and 1.13%, respectively, and 0.73% and 0.17% for hypervolume. This shows that the MOPSO-I is absolutely super to the NSGA-II and SPEA-II.

Also evidently seen from Figure 7(b), the MOPSO-I produces much closer Pareto fronts than the NSGA-II and SPEA-II for the second test problem. However, its Pareto fronts are a little narrower, which may result in that its epsilon and hypervolume indicator are junior to the NSGA-II and SPEA-II. Furthermore, the fact that the Pareto fronts are highly discontinuous would make these metrics irrelevant. In the view of practical spacecraft mission design, obtaining much closer Pareto fronts from which the designer can choose one single satisfying solution for engineering design would be much desirable performance for a multiobjective optimizer. Thus, we can say that the MOPSO-I is a little better than the NSGA-II and SPEA-II; even the statistical results on quantitative metrics do not support this.

In conclusion, the MOPSO showed a better performance with respect to the NSGA-II and the SPEA-II for the two test problems. For other test problems, the MOPSO is not always better, but its performance is very competitive.

**4.5. Comparison with PCP Method.** The widely used Pork-Chop (PCP) method is also tested here for comparison. The  $\Delta v$  is calculated every 1 day in both Earth departure time and asteroid arrival time, and the two-impulse Lambert algorithm is employed. The contours of time of flight corresponding to the optimal transfers for Asteroid 1 are illustrated in Figure 8. For convenience, only the solutions with a  $\Delta v$  less than 18 km/s and Earth departure and asteroid arrive times in [5000, 7000] (MJD2000) are presented.

We analyze the comparisons between the proposed multiobjective optimization approach and the PCP method in the following three aspects.

TABLE 9: Performance comparisons of MOPSO, NSGA-II, and SPEA-II.

Test problem	Algorithm	Epsilon		Hybervolume	
		Mean	std	Mean	std
Asteroid 1, two-impulse	MOPSO-I	0.035041	0.038848	0.876849	0.026781
	NSGA-II	0.113438	0.094049	0.821205	0.080834
	SPEA-II	0.100881	0.032094	0.811347	0.035008
Asteroid 2, three-impulse	MOPSO-II	0.159179	0.081796	0.875023	0.078886
	NSGA-II	0.0594614	0.058546	0.838198	0.301149
	SPEA-II	0.12172	0.055917	0.902724	0.072393

TABLE 10:  $P$  value of MOPSO-I, NSGA-II, and SPEA-II.

	MOPSO-I	NSGA-II	SPEA-II
MOPSO-I		<b>0.0058</b>	<b>0.0113</b>
NSGA-II	0.0073		<b>0.5205</b>
SPEA-II	0.0017	0.1620	

Bold indicates epsilon; the other is hybervolume.

TABLE 11:  $P$  value of MOPSO-II, NSGA-II, and SPEA-II.

	MOPSO-II	NSGA-II	SPEA-II
MOPSO-II		<b>0.0113</b>	<b>0.3847</b>
NSGA-II	0.3447		<b>0.0211</b>
SPEA-II	0.4274	0.5205	

Bold indicates epsilon; the other is hybervolume.

**4.5.1. Uses of the Results.** In the PCP method, the contours of the  $\Delta v$  are plotted to assist the designer to find the best launch window and transfer trajectory. However, the contours have many local peaks and it is not very convenient to determine which solution is the best one. Besides, the size of the region containing the global minimum is a very important factor in finding real trajectories in any given year. This is not easily observed from the Pork-Chop figures.

In contrast to the PCP method, the proposed multi-objective optimization approach can provide friendly the designer of this information. The relations between total characteristic velocity, transfer time, and earth departure time of those obtained Pareto-optimal solutions for two test cases are provided in Figure 8. From Figure 8, it can be clearly observed that the earth departure time for all the optimal rendezvous trajectories concentrates on one arrow domain for these two cases. The optimized Earth launch window and its size can be easily determined through the multiobjective optimization design, which will provide very useful information for engineering design.

**4.5.2. Global Optimality.** The minimum-propellant solution searched by the PCP method for the first asteroid mission is calculated with a  $\Delta v$  of 6015.3 m/s and it is located in Figure 9 (\* denotes). As seen in Figure 1, the MOPSO has located a set of Pareto solutions with a  $\Delta v$  about this value. This issue can also demonstrate the global convergence ability of the MOPSO. Besides, the PCP method obtains only the

two-impulse trajectory. As is well known, the two-impulse trajectory is not the propellant-optimal solution under most conditions, and increasing the number of impulses will reduce the propellant cost. Our proposed approach can design the multi-impulse trajectory; thus it locates better solution than the PCP method.

**4.5.3. Efficiency.** The PCP is in essence of an exhaustive searching method. In our test, the search space for departure time and arrive time is [4000, 10000] day. With the search step of 1 day, the PCP method calculates a total number of 6000 \* 6000 trajectories, while only 400 \* 400 \* 5 trajectories are calculated in the proposed method for the most highly cost case. The proposed approach improves the calculation efficiency by about 40 times. Besides, the PCP method needs much human intention to determine the final solutions, while the proposed approach is an automated search method.

## 5. Conclusions

The paper formulates the asteroid rendezvous preliminary trajectory design as a multiobjective optimization problem and employs the multiobjective particle swarm optimization (MOPSO) algorithm to locate the Pareto-optimal solution set. Compared with the widely employed Pork-Chop method, the proposed approach is demonstrated to be able to provide much more easily used results, obtain better propellant-optimal solutions, and have much better efficiency. The results show that the proposed approach can effectively and efficiently demonstrate the relations among the mission characteristic parameters such as launch time, transfer time, propellant cost, and number of maneuvers, which will provide useful reference for practical asteroid mission design. The MOPSO proves to be quite effective in finding the Pareto-optimal solutions and its performance can be improved by a proposed boundary constraint optimization strategy. The MOPSO is found to be very competitive with respect to two highly competitive multiobjective evolutionary algorithms: the NSGA-II and the SPEA-II.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

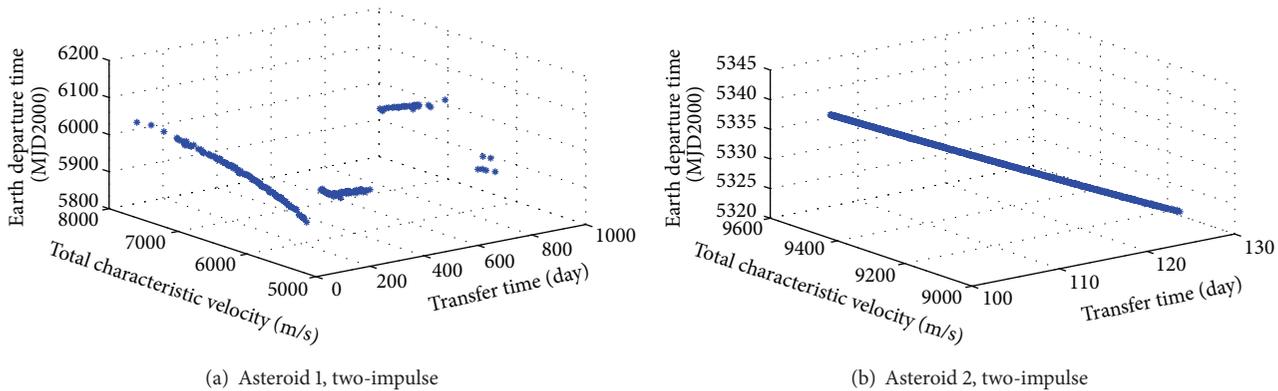


FIGURE 8: Relations between total characteristic velocity, transfer time, and earth departure time.

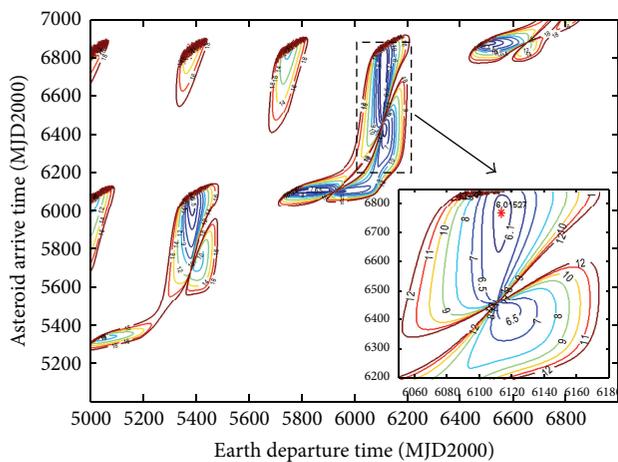


FIGURE 9: Contours of time of flight corresponding to the optimal transfers (Asteroid 1).

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 11222215), the 973 Project (no. 2013CB733100), and the Hunan Provincial Natural Science Foundation of China (no. 13JJ1001).

## References

- [1] N. D. Hulkower, C. O. Lau, and D. F. Bender, "Optimum two-impulse transfer for preliminary interplanetary trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 4, pp. 458–461, 1984.
- [2] C. O. Lau and N. D. Hulkower, "Accessibility of near-earth asteroids," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 3, pp. 225–232, 1987.
- [3] E. Perozzi, A. Rossi, and G. B. Valsecchi, "Basic targeting strategies for rendezvous and flyby missions to the near-earth asteroids," *Planetary and Space Science*, vol. 49, no. 1, pp. 3–22, 2001.
- [4] D. Qiao, H. T. Cui, and P. Y. Cui, "Evaluating accessibility of near-earth asteroids via earth gravity assists," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 502–505, 2006.
- [5] D. F. Lawden, *Optimal Trajectories for Space Navigation*, Butterworths, London, UK, 1963.
- [6] J. E. Pussing, "Primer vector theory and applications," in *Spacecraft Trajectory Optimization*, B. A. Conway, Ed., pp. 16–36, Cambridge University Press, New York, NY, USA, 2010.
- [7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Piscataway, NJ, USA, December 1995.
- [8] C. R. Bessette and D. B. Spencer, "Identifying optimal interplanetary trajectories through a genetic approach," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA Paper 2006-6306, pp. 809–826, Keystone, Colo, USA, August 2006.
- [9] K. Zhu, F. Jiang, J. Li, and H. Baoyin, "Trajectory optimization of multi-asteroids exploration with low thrust," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 52, no. 175, pp. 47–54, 2009.
- [10] M. Pontani and B. A. Conway, "Particle swarm optimization applied to space trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1429–1441, 2010.
- [11] M. Pontani, P. Ghosh, and B. A. Conway, "Particle swarm optimization of multiple-burn rendezvous trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1192–1207, 2012.
- [12] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: a survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.
- [13] K. B. Lee and J. H. Kim, "Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 755–766, 2013.
- [14] X. Y. Shao, W. Q. Liu, Q. Liu, and C. Y. Zhang, "Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9–12, pp. 2885–2901, 2013.
- [15] M. Shokrian and K. A. High, "Application of a multi objective multi-leader particle swarm optimization algorithm on NLP

- and MINLP problems,” *Computers & Chemical Engineering*, vol. 60, pp. 57–75, 2014.
- [16] A. Arias-Montañó, C. A. C. Coello, and E. Mezura-Montes, “Multiobjective evolutionary algorithms in aeronautical and aerospace engineering,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 662–694, 2012.
- [17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [18] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: improving the strength Pareto evolutionary algorithm,” in *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds., pp. 95–100, Springer, Berlin, Germany, 2002.
- [19] S. P. Hughes, L. M. Mailhe, and J. J. Guzman, “A comparison of trajectory optimization methods for the impulsive minimum fuel rendezvous problem,” in *Guidance and Control 2003*, I. J. Gravseth and R. D. Culp, Eds., vol. 113 of *Advances in the Astronautical Sciences*, pp. 85–104, 2003.
- [20] Y. Z. Luo, G. J. Tang, and Y. J. Lei, “Optimal multi-objective nonlinear impulsive rendezvous,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 994–1002, 2007.
- [21] Y. Z. Luo, G. J. Tang, and Y. J. Lei, “Optimal multi-objective linearized impulsive rendezvous,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, pp. 383–389, 2007.
- [22] Y. Z. Luo, Z. Yang, and H. N. Li, “Robust optimization of nonlinear impulsive rendezvous with uncertainty,” *Science China: Physics, Mechanics & Astronomy*, vol. 57, no. 3, pp. 1–10, 2014.
- [23] G. T. Pulido and C. A. C. Coello, “Using clustering techniques to improve the performance of a particle swarm optimizer,” in *Genetic and Evolutionary Computation—GECCO 2004*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 225–237, Springer, Berlin, Germany, 2004.
- [24] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [25] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [26] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms—a comparative case study,” in *Parallel Problem Solving from Nature—PPSN V*, A. E. Eiben, Ed., vol. 1498 of *Lecture Notes in Computer Science*, pp. 292–301, Springer, Amsterdam, The Netherlands, 1998.
- [27] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, Wiley Series in Probability and Statistics, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 1999.
- [28] Y. Z. Luo, J. Zhang, H. Y. Li, and G. J. Tang, “Interactive optimization approach for optimal impulsive rendezvous using primer vector and evolutionary algorithms,” *Acta Astronautica*, vol. 67, no. 3-4, pp. 396–405, 2010.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

